MARF: Multiscale Adaptive-Switch Random Forest for Leg Detection With 2-D Laser Scanners

Tianxi Wang®, Feng Xue®, Yu Zhou®, Member, IEEE, and Anlong Ming®, Member, IEEE

Abstract—For the 2-D laser-based tasks, e.g., people detection and people tracking, leg detection is usually the first step. Thus, it carries great weight in determining the performance of people detection and people tracking. However, many leg detectors ignore the inevitable noise and the multiscale characteristics of the laser scan, which makes them sensitive to the unreliable features of point cloud and further degrades the performance of the leg detector. In this article, we propose a multiscale adaptive-switch random forest (MARF) to overcome these two challenges. First, the adaptive-switch decision tree is designed to use noise-sensitive features to conduct weighted classification and noise-invariant features to conduct binary classification, which makes our detector perform more robust to noise. Second, considering the multiscale property that the sparsity of the 2-D point cloud is proportional to the length of laser beams, we design a multiscale random forest structure to detect legs at different distances. Moreover, the proposed approach allows us to discover a sparser human leg from point clouds than others. Consequently, our method shows an improved performance compared to other state-of-the-art leg detectors on the challenging Moving Legs dataset and retains the entire pipeline at a speed of 60+ FPS on low-computational laptops. Moreover, we further apply the proposed MARF to the people detection and tracking system, achieving a considerable gain in all metrics.

Index Terms—2-D laser scanner, adaptive switch, leg detection, multiscale, random forest (RF).

I. INTRODUCTION

As the emergence of the various domain-specific requirements for the robot, such as security [1], human–computer interaction (HCI) [2], [3], and navigation [4], [5], it is generally acknowledged that the people detection and tracking are necessary for robot [6], [7]. Recently, an increasing number of tasks [8]–[10] utilize laser scanner to perceive people and the environment. However, as shown in Fig. 1, since the 2-D laser scanner only measures the depth of a 2-D plane above the ground, it is impossible to directly capture 3-D human from 2-D point clouds. To overcome this challenge, several previous methods mount the 2-D laser scanner at the human leg’s height, first detect each leg, and second match two legs to construct a human. Therefore, leg is considered as the input of these 2-D laser-based people detection and tracking systems. For this reason, the leg detector plays a dominant role in the performance of these systems. Next, we introduce the existing leg detector.

Many leg detectors [11]–[13] employ a two-step leg detection approach as follows.

1) First, a 2-D laser scan is considered as a point cloud in 2-D space and is segmented into several point clusters by the distance jump between adjacent laser points [14].
2) Second, several geometric features are employed to represent these atom clusters, so that a classifier is trained to classify each atom cluster into leg or nonleg.

Following this pipeline, Chung et al. [11] utilized the support vector data description to classify atom clusters.
Li et al. [13] designed several spatial relationship features and utilized AdaBoost [15] to classify clusters. In recent years, random forest (RF) is widely employed to classify clusters due to its high efficiency and accuracy. It constructs many decision trees and adopts a bagging method to train each tree and aggregates the predictions of all trees. For clarity, RF is expressed as standard RF (SRF) below. Leigh et al. [12] and Linder et al. [16] proposed a SRF-based detector and applied it to build a robust people detection and tracking framework. However, the noise and the multiscale property of laser scans are rarely discussed, which degrades the performance of these SRF-based leg detectors. We present them in detail below, respectively.

**Global-Local Confidence Conflict:** We assume that the set of all clusters is expressed as $X \in \mathbb{R}^{F \times N}$, consisting of the leg clusters $X^+ \in \mathbb{R}^{F \times N^+}$ and the nonleg clusters $X^- \in \mathbb{R}^{F \times N^-}$, where $F$ is the dimensions of features and $N$ is the number of clusters, $N^+ + N^- = N$. For the cluster set $X$, we define the confidence of the $j$th feature as the dissimilarity between the $j$th feature distribution of the leg set $X^+$ and that of the nonleg set $X^-$, which is formulated as follows:

$$C_j = 1 - \frac{\min(\delta_j^+, \delta_j^-)}{\max(\delta_j^+, \delta_j^-)} \left\{ \begin{array}{ll} \delta_j^+ = \frac{1}{N^+} \sum_i (X^+_{ij} - \bar{x}_j^+)^2 \\
\delta_j^- = \frac{1}{N^-} \sum_i (X^-_{ij} - \bar{x}_j^-)^2 \end{array} \right. \tag{1}$$

where $i$ is the index of cluster and $\bar{x}_j^+$ is the average of the $j$th feature of leg clusters. Generally, a high confidence $C_j$ indicates a high difference between the $j$th feature of leg clusters and that of nonleg clusters, as shown in Fig. 1(a). During training a decision tree, the root node is trained to classify a sample set $X^+ \subseteq X$, and other internal nodes are trained to classify partial samples $A \subseteq X^*$. Assuming that the global confidence $\Phi_j$ and local confidence $\phi_j$ are the confidence of the $j$th feature on set $X$ and set $A$, the nodes of tree always learn a feature has a maximum local confidence $\phi_j$, but ignore the low global confidence $\Phi_j$ caused by noise. If the selected feature has a low confidence $\Phi_j$, the local-global confidence conflict occurs

$$\phi_j - \Phi_j \geq \epsilon, \epsilon \text{ is a threshold.} \tag{2}$$

In this case, the learned feature and the split point easily fail to classify samples out of the subset and, thus, degrades the ability of the tree to recognize the leg cluster. As shown in Fig. 1, the $j$th feature has strong discrimination on the positive and negative samples marked by orange circle well, but an extremely weak one on the samples marked by a gray circle. This phenomenon is common due to the inevitable fluctuation of noisy laser data.

**Multiscale Characteristics:** Since the laser beam consists of many rays emitted from the center of the scanner, the farther the point cluster is, the sparser the cluster is, and vice versa. This scattering mechanism affects the value of some distance-sensitive features, such as the number of points, average inscribed angular, and distance to the scanner. However, existing methods usually classify multiscale clusters by using a single-scale classifier, which degrades the leg detection performance in all scales.

**Contributions:** Aiming to address these two issues and significantly boost the accuracy of the leg detector, a novel multiscale adaptive-switch RF (MARF) is proposed in this article. By evaluating the local and global confidence of the feature, an adaptive-switch decision tree is proposed. It employs an adaptive-switch strategy to decide whether a node of adapt is switched to a regular node or a dichotomous node, which alleviates the misclassification caused by the global-local confidence conflict. Moreover, by taking the multiscale characteristic of the leg features into consideration, a multiscale structure is designed to detect the legs at multiple distances. Sufficient experimental results demonstrate the efficiency of the presented MARF.

In summary, the key contributions of this article lie in the following.

1) The global-local confidence conflict of RF is studied, which motivates us to introduce an adaptive-switch decision tree. It would conduct a weighted split if a conflict occurred, making the model more robust when using noise-sensitive features.

2) A multiscale RF (MRF) composed of adaptive-switch decision trees is introduced, which considers the multiscale property of laser scan to improve leg detection accuracy in all scales.

3) The proposed method outperforms all other leg detectors and improves the performance in the application of people detection and tracking. Meanwhile, it retains the entire leg detection pipeline at a speed 60+ FPS on a low-computational laptop.

II. RELATED WORKS

A. Review of Leg Detection

As a fundamental task of people detection in a 2-D laser scanner, there have been several works on leg detection in past decades [13], [17]–[19]. These leg detectors can be classified into two types: 1) simple heuristic methods and 2) learning-based methods.

Simple heuristic methods commonly use handcrafted features and threshold conditions to distinguish leg and nonleg clusters. Moreover, the selection of thresholds requires experience. Xavier et al. [20] used leg-like parameters, e.g., diameter, to distinguish nonleg and leg. Similarly, Topp and Christensen [21] proposed to pick out some legs with blob-like shapes, and Cui et al. [22] pursued extracting moving leg blobs to detect and track stationary or moving people. These methods are simple but lack generalization in clutter scenarios.

Learning-based methods couple handcrafted features or deep features with machine-learning approaches to build effective leg detectors and show excellent performance. Arras et al. [18] utilized geometric features of clusters to train an AdaBoost [15] classifier to detect legs. Weinrich et al. [23] designed distance-invariant features to describe clusters and train classifiers for people and wheelchairs detection. Based on prevailing works, Leigh et al. [12] combined features proposed in [24] and [25] and employed SRF to classify legs and detect people. The above works aim at conducting the leg detection by different classifiers but ignore the
inherent issues of LiDAR noise, occluded leg, etc., in the laser-based leg detection. To eliminate the misdetection of partially occluded legs, Li et al. [13] proposed multitype features to train strong AdaBoost classifiers. However, the complete lack of discussion in speed and noise impact restricts its practical value. Cha and Chung [26] detected the human leg in 3-D feature space for a person-following mobile robot, which improves the recall of leg detection, but suffers from the precision drop due to lacking discriminative feature representation. Recently, Beyer et al. [27] proposed a novel deep learning method to detect people and wheelchairs and show remarkable performance. However, the deep learning-based methods can hardly strike a balance between efficiency and accuracy on the low-cost robot currently. Different from them, we focus on the rarely discussed LiDAR feature noise and propose a noise-immune RF. Compared to our baseline [12], MARF improves the performance in all metrics. In addition, the proposed model can be directly trained based on noisy samples of other tasks, reflecting a better generalization of our model.

B. Review of Standard Random Forest

SRF is an ensemble learning method, which is utilized as a cluster classifier to divide point clusters into legs and nonlegs in [12]. It aggregates the outputs of a set of standard decision trees [28] to improve the prediction accuracy [29], [30]. The structure of each decision tree is shown in Fig. 2(a), which is a standard decision tree composed of regular internal nodes and leaf nodes.

Each decision tree is trained by a randomly selected subset from entire clusters training set X. In detail, from the root of the tree, the regular internal node recursively splits the reached clusters by a binary-split function below and passes them to the left or right child node, formulating two subsets

$$g(x_n, \tau_j) = \left[x_{n,j} < \tau_j \right]$$ (3)

where $x_{n,j}$ is the value of the $j$th feature of the $n$th cluster $x_n$, $\tau_j$ is a split point corresponding to the $j$th feature, and $\left[ \cdot \right]$ is the indicator function. If $g(x_n, \tau_j) = 1$, cluster $x_n$ is passed to the left child node, otherwise right.

To determine the optimal-split point $\tau_j$ that separates leg and nonleg clusters as completely as possible, Gini impurity is employed to evaluate the impurity of each split subset

$$G = \sum_c \lambda^c (1 - \lambda^c)$$ (4)

where $\lambda^c$ is the percentage of $c$-labeled clusters among the subset. Traversing each feature and each split point, the regular internal node selects the $j$th feature and split point $\tau_j$ that minimizes the sum of impurity $G$ of the two split sets on the node as the optimal parameter.

In this way, SRF learns optimal-split point $\tau_j$ for internal nodes that seem to distinguish the clusters clearly. However, since some features are sensitive to the noise of laser data, the split point learned from the partial training set tends to deviate from the entire training set, which is not optimal. During prediction, the node utilizes improper split points to split clusters to the left or right, which is prone to assign clusters with wrong labels mistakenly.

C. Review of Probabilistic Random Forest

Considering that features may be affected by noise, different from SRF, probabilistic RF (PRF) [31] does not split the samples, but estimates a left-split probability and a right-split probability for each sample. Then, it passes each sample to the left child node with the left-split probability and the right with the right-split probability simultaneously. As shown in Fig. 2(b), the probabilistic node is designed to calculate the probabilities. Assuming that $x_n$ denotes the $n$th reached sample on the current probabilistic node, the probabilistic-split function of this node can be formulated as follows:

$$h(x_n, \tau_j) = \{ p_b \cdot p_l, p_b \cdot p_r \}$$ (5)

where $\tau_j$ is a split point corresponding to the $j$th feature. $p_b$ is the basic probability of $x_n$, which is the probability passed from the parent node and initialized with 1 at the root node. $p_l$ and $p_r$ are the left-split probability and right-split probability of $x_n$ at the current node, respectively. PRF assumes that each feature of sample $x_n$ contains a Gaussian noise. Then, they define that $p_l$ and $p_r$ represent the probabilities of $x_n,j < \tau_j$ and $x_n,j \geq \tau_j$, respectively, where $x_n,j$ is the $j$th feature of sample $x_n$, considered as a random variable. In more details, probability $p_l$ can be figured by the cumulative distribution function of Gaussian distribution $p_l = P(x_{n,j} \leq \tau_j)$, and $p_r = 1 - p_l$. Then, $p_b \cdot p_l$ and $p_b \cdot p_r$ are passed to the left child node and
Basic Geometric Features

| Feature             | Description                      |
|---------------------|----------------------------------|
| Width               |                                  |
| Linearity           |                                  |
| Number of points    |                                  |
| Mean angular change |                                  |
| Average distance to median |        |

Circularity and Basic Geometric Features

| Feature             | Description                      |
|---------------------|----------------------------------|
| Linearity           |                                  |
| Number of points    |                                  |
| Mean angular change |                                  |
| Average distance to median |        |

 Relative Distance Features

| Feature             | Description                      |
|---------------------|----------------------------------|
| Mean curvature      |                                  |
| Distance to scanner |                                  |
| Average inscribed angular |                |
| Left occlusion      |                                  |
| Boundary regularity |                                  |
| Standard inscribed angular |            |
| Distance to scanner per point |   |
| Right occlusion     |                                  |

the right child node together with sample $x_n$, respectively. Note that $\tau_j$ is also determined by minimizing the sum of impurity of two sample subsets. But different from SRF, PRF defines $\lambda^c$ in (4) as $\lambda^c = \sum p_b^c / \sum p_b$ to measure the impurity of each split set, where $p_b^c$ indicates the basic probability of $c$-labeled sample in the current node.

Notably, as cluster $x_n$ goes deeper, the passing probability of $x_n$ sharply decreases. Once the $p_b$ approaches 0, namely, lower than an eligible threshold, the training is terminated, and this node is transformed to a leaf node. As shown in the chart in Fig. 2(b), the problem above is prone to generate lots of shallow-layer leaf nodes, which degrades the discrimination of the entire model.

III. PROBLEM FORMULATION

Following the prior methods [12], [18], this article decouples leg detection into two steps: 1) points clustering and 2) leg classification. In the first step, given a 2-D laser scan as input, the laser scan is clustered into a set of point clusters according to the jump distance between adjacent laser points [12], and each point cluster is a candidate for leg classification. In the second step, each cluster is expressed by utilizing several features [24], [25], as shown in Table I. And the cluster is classified into two types, namely, leg or nonleg. Typically, there are two types of features: 1) basic geometric features and 2) relative distance features. Basic geometric features describe the geometric shape parameters of the cluster. Relative distance features describe the distance between adjacent points of the cluster and are more sensitive to the distance variance. In detail, let $x_n \in \mathbb{R}^{17}$ denotes the feature vector of the $n$th cluster, all feature vectors are stacked as the entire training set $X \in \mathbb{R}^{N \times 17}$. The corresponding labels are denoted as $Y \in \mathbb{R}^N$, i.e., $+1$ for leg clusters and 0 for nonleg clusters.

IV. METHOD

In this section, our method is described in two levels: 1) tree level and 2) forest level. We first propose the adaptive-switch decision tree to overcome the global-local confidence conflict (see Section IV-A). Second, the MARF is proposed to fully utilize the multiscale characteristics of the human leg in the 2-D space (see Section IV-B).
Different from the regular internal node, the dichotomous node calculates two weights of each cluster $x_n \subset A_i$ by the dichotomous-split function as follows:

$$d(x_n, \tau_j) = \{w_l, w_r\}$$  \hspace{1cm} (6)

where $j$ indicates the $j$th feature selected by the regular internal node before the node is converted to the dichotomous node. $\tau_j$ is the reselected split point corresponding to the $j$th feature. Noteworthy, due to the effect of noise on features, we make a hypothesis that the $j$th feature $x_{n,j}$ has a Gaussian noise with variance $\sigma^2_j$, i.e., $x_{n,j} \sim \mathcal{N}(x_{n,j}, \sigma^2_j)$, where $\sigma^2_j$ is the variance of the $j$th feature in set $X$. More experimental discussion about the Gaussian noise hypothesis can be found in the supplementary material. Based on this hypothesis, $w_l$ is defined as the cumulative distribution of $x_{n,j}$ at split point $\tau_j$. And $w_r$ is the complementary cumulative distribution of $x_{n,j}$ at point $\tau_j$, i.e., $w_l + w_r = 1$. They are regarded as the weights of cluster $x_n$ passed to the left and right child nodes, respectively, and they are formulated as follows:

$$w_l = \frac{1}{\sqrt{2\pi}\sigma_j} \int_{-\infty}^{\tau_j} \exp\left(-\frac{(z - x_{n,j})^2}{2\sigma_j^2}\right)dz$$

$$w_r = \frac{1}{\sqrt{2\pi}\sigma_j} \int_{\tau_j}^{+\infty} \exp\left(-\frac{(z - x_{n,j})^2}{2\sigma_j^2}\right)dz$$  \hspace{1cm} (7)

where $z$ denotes the variable of integration.

To learn the split point $\tau_j$ that achieves the minimum summed impurity of the two split subsets, consisting of weighted clusters, we calculate the Gini impurity as follows. Assuming that the leg cluster corresponds to the superscript “+” and the nonleg cluster corresponds to “−”, the Gini impurity of left or right sets is calculated as follows:

$$G_{ij} = 2 \times \frac{\sum w_{l,j}^{+} \times \sum w_{r,j}^{-}}{\sum w_{l,j}^{+} \times \sum w_{r,j}^{-}}$$  \hspace{1cm} (8)

where $w^+$ denotes the weights of the leg clusters, and $l$ and $r$ indicate the left set and the right set. Furthermore, all split points of the $j$th feature are tried to minimize the summed impurity of the left set and right set

$$\tau_j = \text{arg min}_{\tau_j} \left(G_l \times \sum w_l + G_r \times \sum w_r\right).$$  \hspace{1cm} (9)

Finally, the dichotomous node stores the selected $j$th feature and the optimal-split point $\tau_j$ and then passes all clusters to both left child nodes. Furthermore, the left weight $w_l$ is passed to the left child node, and $w_r$ corresponds to the right child node, as shown in Fig. 3.

In this way, each regular internal node and dichotomous node are trained to construct an adaptive-switch decision tree. In addition, the node stops training and turns into a leaf node if any of the following constraints are met.

1) The cluster number of set $A_i$ satisfies $N_A < 2$.

2) The depth of the node is higher than 20.

3) The summed Gini impurity of the node is less than $e^{-6}$. Each leaf node saves the dominant label of clusters reaching on this node.

**Prediction of a Single Tree:** Assuming that $x_i$ denotes the feature vector of a point cluster, and $f$ denotes a trained adaptive-switch decision tree, we use the model $f$ to predict the label of the cluster $x_i$. The leaf nodes this cluster $x_i$ reached are denoted as $[l_q]$, $l_q \in \{0, 1\}$, $q \in \{1, 2, \ldots, Q\}$, and the weights of $x_i$ partitioned by the dichotomous nodes on the path from the root to the $q$th leaf nodes are denoted as $[w_l^q]$, $w_l^q \in \{0, 1\}$, $e \in \{1, 2, \ldots, E\}$, where $Q$ is the number of leaf nodes and $E$ is the number of dichotomous nodes that the sample $x_i$ passes from the root node to the $q$th leaf node. The prediction of the tree $f$ is formulated as follows:

$$f(x_i) = \begin{cases} 1, & \sum_{q=1}^{Q} \prod_{e=1}^{E} (w_l^q \times l_q) > 0.5 \\ 0, & \text{otherwise} \end{cases}$$  \hspace{1cm} (10)

where $f(x_i)$ denotes the label of cluster $x_i$ predicted by $f$. Compared to SRF, the proposed tree no longer decisively classifies clusters, but jointly considers the weights of the labels on multiple leaf nodes, avoiding wrong splitting caused by the noise-sensitive feature.

**Differences From PRF:** Although our method is inspired by PRF [31], there are considerable differences between them. First, PRF conducts probabilistic classification on all nodes, but our method only conducts weighted split when a noise-sensitive feature is inevitably selected, and the global-local confidence conflict occurs. Second, during training a node, PRF computes the probability of each sample by multiplying all the probabilities on the path from the root to this node, causing a sharp drop in the basic probabilities and premature termination of training. In contrast, our method rebalances the weights for the clusters that reached the current node before the split. The two differences make our methods fully trained and improve the performance of the leg detector.

**B. Multiscale Adaptive-Switch Random Forest**

Recently, the multiscale strategy has been proven to be very useful for detection in point clouds [32] and image [33]. To discover the long-distance legs, we design the MARF. It utilizes the adaptive-switch decision tree and the multiscale property of the point cluster. In this section, we first introduce the discrete 2-D space to determine the scale of each cluster and then describe the structure and the training scheme of MARF.

**Spatial Discretization:** To determine the scale of each cluster, we first discretize the depth interval $[0m, +\infty)$ into $K$ subintervals, where $K \in \mathbb{N}^+$. Since SRF hardly distinguishes the clusters at a distance larger than 6 m, we assign the point clusters further than 6 m into one scale. Then, we partition range $[0m, 6m)$ to determine other scales. In this way, the 2-D space is discretized into several depth intervals. For a two-scale model, the space is discretized into $[0m, 6m)$ and $[6m, +\infty)$, $[0m, 3m)$, $[3m, 6m)$, and $[6m, +\infty)$ for three-scale model, and $[0m, 1.5m)$, $[1.5m, 3m)$, $[3m, 6m)$, and $[6m, +\infty)$ for four-scale model. All clusters in the $k$th scale are denoted as $[X_k | X_k \in \mathbb{R}^{N_k \times 17}, k \in \{1, 2, \ldots, K\}]$.

**Structure:** The MARF consists of several adaptive-switch decision trees. Corresponding to the number of scales $K$, there are $K$ sets of trees in MARF, namely, the adaptive-switch RF (ARF), as shown in Fig. 4. In detail, the $k$th ARF contains $T_k$ trees. To recognize the long-distance leg as much as possible,
the kth ARF is utilized to classify the point clusters inside the kth scale. For instance, when K = 3, the 1st ARF classifies clusters inside the range \([0 \, \text{m}, +\infty)\), the 2nd ARF classifies clusters inside the range \([3 \, \text{m}, +\infty)\), and the 3rd ARF corresponds to the range \([6 \, \text{m}, +\infty)\). The training scheme is described as follows.

**Multiscale Training Scheme:** Let \( F = \{F^{(k)}\}, k \in \{1, 2, \ldots, K\} \) denotes the MARF with K scales, where \( F^{(k)} \) is the kth ARF in MARF. To capture the long-distance leg that is hardly detected, we adopt a biased sampling strategy to construct the training set for each tree during the training. More specifically, we randomly select clusters from set \( X \) with replacement. For the randomly selected cluster in the set \( X_{k} \cup X_{k+1} \cup \cdots \cup X_{N} \), it is accepted as a training sample. To avoid overfitting caused by the lack of scale variety, for the randomly selected cluster in the set \( X_{1} \cup X_{2} \cup \cdots \cup X_{k-1} \), it is accepted with probability \( \alpha \). In this way, we construct an independent training set that contained \( N \) clusters for each decision tree in \( F^{(k)} \). Then, each decision tree is constructed by following the way described in Section IV-A.

Since ARF \( F^{(1)} \) learns the leg clusters of all scales, it can be used to predict the legs of all scales. In contrast, ARF \( F^{(2)} \) focuses on learning the farther legs, further alleviating the false-positive detection of long-distance legs. And so on, ARF \( F^{(K)} \) has a more robust capability to distinguish the clusters inside the Kth scale. Thus, this ARF is only employed to predict the farthest clusters.

**Leg Prediction:** To recognize the legs in all scales, we introduce an overlapping fusion strategy to fuse all ARF predictions. Fig. 4 illustrates the prediction of a three-scale MARF. For a point cluster \( x \) in the kth scale, ARFs \( \{F^{(k)}\}_{k \in \{1, \ldots, K\}} \) are employed to predict its probability of being a leg. Each ARF outputs the average value of the outputs of all decision trees within it. Finally, if cluster \( x_{n} \) is predicted by number \( k \) scales ARF, MARF predicts the probability that the cluster belongs to the leg, can be formulated as follows:

\[
F(x_{n}) = \frac{1}{k} \sum_{k=1}^{K} F^{(k)}(x_{n}).
\]

In our method, if \( F(x_{n}) \) is higher than \( \beta \), the cluster \( x \) is predicted as a leg cluster, where \( \beta \) is a threshold for label assignment.

V. Experiments

In this section, the performance of MARF is evaluated and analyzed in detail. The experimental settings are first demonstrated in Section V-A. Next in Section V-B, the results of other leg detectors are compared with our method. Then, the ablation study is given in Section V-C to illustrate the significance of each design. Furthermore, in Section V-D, we conduct analysis and comparative experiments to represent the effectiveness of details in our method. Finally, we apply our method to a people detection and tracking framework to study its advancement in Section V-E.

A. Experimental Settings

1) Datasets: The Moving Legs dataset is collected from the People Tracking dataset [12] to evaluate the performance of leg detection. The laser scan of the Moving Legs dataset is the same as the People Tracking dataset. But the label of the Moving Legs dataset is extracted from People Tracking. In detail, the dataset labels the location of each person in the laser scan. Following the criterion utilized in [12], we define the true leg cluster as each two closest clusters near each ground-truth people, with the condition that the distance between the leg cluster and the location of the people should be less than 0.5 m. In this way, the collected dataset contains 26,344 legs and 72,251 nonlegs. In addition, since part of scenarios are only labeled with the positions of people tracked, hence, only leg clusters are collected from those laser scans. Generally, clusters collected in this dataset are all in the real environment. Therefore, the evaluation results on this dataset can significantly reflect the performance of leg detection.

**People Tracking** is a ROS-enable dataset provided by [12], which is the only public and available dataset for 2-D laser-based people tracking as we know. This dataset is captured by a laser scanner with 0.35° angle resolution and 15 m maximum

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
detection distance. Moreover, this dataset is collected in the scenarios with single and multiple people tracked, and those scenarios are further employed to evaluate the contribution of our approach for people detection and tracking.

1) SCENARIO I: Multiple people and stationary robot.
2) SCENARIO II: Multiple people and moving robot.
3) SCENARIO III: Single people and moving robot.

Wherein long-distance people and crowds are the main challenges for people detection and tracking in SCENARIO I. In SCENARIO II, the robot motion induces more fluctuations, which further increases leg detection difficulty. In SCENARIO III, only the tracked person is labeled, and the tracked person walks naturally with interaction with the environment, including speed change and crowd crossing, and this scenario is also challenging for leg detection.

2) Evaluation Metrics: Following previous works [18], [23], TP, FP, precision rate, and recall rate are employed as metrics to evaluate the leg detection performance. TP is the number of leg clusters that are successfully detected. FP corresponds to nonleg clusters classified to legs. The precision and recall rates are formulated as follows:

\[
\text{Precision} = \frac{TP}{TP + FP} \quad \text{Recall} = \frac{TP}{TP + FN}
\]

where FN corresponds to the leg clusters that fail to be detected. Furthermore, to intuitively compare the performance of different leg detectors, the precision–recall curve (PR curve) is depicted in our experiment by taking 100 values from 0.1 to 0.9 with equal intervals as the label assignment threshold β.

To further evaluate the contribution of our MARF to people detection and tracking, we evaluate the performance on several indicators: the number of people successfully detected (Correct), the number of people missed (Miss), the number of wrong detected people (FPP), and widely used multiobject tracking accuracy (MOTA) [34]. The MOTA is formulated as follows:

\[
\text{MOTA} = 1 - \frac{\sum_i (ID_i + \text{Miss}_i + \text{FPP}_i)}{\sum_i GT_i}
\]

where ID_i is the number of people whose ID changed from frame i−1 to frame i, Miss_i, FPP_i are the Miss and the FPP in frame i, GT_i is the number of people labeled in frame i.

3) Parameter Settings: ϵ in (2) is a parameter to check global-local confidence conflict. To determine its value setting, we explore the effect of choices of parameter ϵ, as shown in Fig. 5. The experiments are based on the best model configuration MARF@3, ϵ is set to 0.1, and α is set to 0.6. The default number of scale K is set to 3.

Table II Comparison With Other Leg Detector

| Method | Model | TP   | FP   | Precision | Recall |
|--------|-------|------|------|-----------|--------|
| Belliotto* et al.[35]   | -     | 14722| 18470| 55.88%    | 44.27% |
| Aras* et al.[18]       | AdaBoost| 13474| 10080| 57.20%    | 51.15% |
| Leoni et al.[12]       | SRF   | 21415| 5272 | 80.25%    | 81.29% |
| Reis* et al.[31]       | PRF   | 22363| 3428 | 86.71%    | 84.89% |
| Ours                 | MARF@3| 22955| 2825 | 89.04%    | 87.14% |

Label assignment threshold β is set to 0.5. * marks the re-implementation version.

B. Comparison With Other Leg Detectors

Table II shows the leg detection performances of several leg detectors. The detector [35] is a heuristic method for leg recognition that directly outputs the cluster’s label and we extend it for comparison of leg detection. The detector [18] is an AdaBoost-based detector trained with the features they design. The detector [12] is a public SRF-based leg detector. And the detector [31] is a PRF-based leg detector. It is first proposed to classify noisy astronomical datasets, and we reimplement it to classify legs for comparison. Our leg detector is the proposed MARF@K model, and the K is set to 3.

Intuitively, our method achieves 8.79% precision improvement and 5.85% recall improvement over the SRF-based detector. Furthermore, the FP of ours is only 53.58% of that of the SRF-based method. There is no doubt that our method has a considerable boost in performance compared with the SRF-based method, as the proposed method has immunity to noise that is inevitable in 2-D laser. Moreover, although the PRF-based detector also has specific immunity to noise, ours achieves 2.33% precision improvement, 2.25% recall improvement, and 17.6% FP decrease over the PRF-based detector. The reason is that the proposed method avoids terminating training too early and fully utilizes the multiscale characteristics of the leg cluster. For TP, our method also achieves the best performance. Fig. 6 compares their performances more comprehensively. Intuitively, our method outperforms...
other methods at almost all thresholds. Although PRF achieves a weak advantage when recall ranges from 0.9 to 0.95, its precision has a more apparent decline when recall ranges from 0.6 to 1. In contrast, ours achieves a great advantage when recall ranges from 0.6 to 0.9 while maintaining a high and stable precision, which further proves the effectiveness of our method.

C. Ablation Study

To demonstrate the significance of the adaptive-switch decision tree and multiscale structure, respectively, we disable each proposed module to evaluate our contributions. Note that all variants are derived from MARF@3 and evaluated at the Moving Legs dataset.

1) Contribution of Adaptive-Switch Decision Tree: The MRF is the SRF with a multiscale structure, replacing the adaptive-switch decision tree with a standard decision tree. Table III shows the variants of the proposed leg detector. MARF@3 achieves 6.98% precision improvement, 1.07% recall improvement, and 43% FP decrease over MRF, which is due to the effectiveness of the proposed adaptive-switch decision tree. It is noteworthy that the significant drop in false-positive detection helps the follow-up work to eliminate nonleg clusters. The PR curves, shown in Fig. 6, also illustrate the necessity of an adaptive-switch decision tree. The reason is that the proposed method measures the global-local confidence conflict and employs dichotomous nodes to make a more reasonable split, promoting the discriminative capability of each classifier in the forest.

2) Contribution of Multiscale Structure: Discarding the multiscale structure, ARF is a single-scale RF composed of adaptive-switch decision trees. By comparing MARF@3 and ARF in Table III, it can be seen that MARF@3 achieves 3.32% precision improvement, 0.98% recall improvement, and 25.3% FP decrease over MRF. Moreover, the PR curve shown in Fig. 6 depicts a similar observation. The intrinsic reason is that the multiscale structure contributes to the robustness of the exploitation of multiscale leg features and overlapping fusion boosts leg detection accuracy in all scales.

D. Model Details Analysis

Previous comparisons illustrate the performance and advantage of our method. However, there are still some details of module design that are worthy of being further discussed. In this section, we first discuss how the setting of scale number $K$ influences the performance of our model. Subsequently, another distinct multiscale structure is employed to compare with the proposed structure. Finally, to illustrate the contribution of the proposed dichotomous node, based on the best multiscale structure, we replace the dichotomous node with the probabilistic node for comparison.

1) Analysis of the Scale Number: In this part, three multiscale structures with different scale numbers are designed for experimental comparison. We refer to the MARF with $K$ scales forests as MARF@$K$, and the scale details are presented as follows.

1) MARF@2: $[0 \ m, 6 \ m], [6 \ m, +\infty)$.
2) MARF@3: $[0 \ m, 3 \ m], [3 \ m, 6 \ m], [6 \ m, +\infty)$.
3) MARF@4: $[0 \ m, 1.5 \ m], [1.5 \ m, 3 \ m], [3 \ m, 6 \ m], [6 \ m, +\infty)$.

In our method, three ARFs in MARF@3 are employed to predict the cluster in three ranges: 1) $[0 \ m, +\infty)$; 2) $[3 \ m, +\infty)$; and 3) $[6 \ m, +\infty)$. The results are shown in Table IV. MARF@3 outperforms MARF@2 and MARF@4 and obtains the best performance. The results adequately illustrate that the depth discretization of MARF@3 is more suitable for leg detection in different distances.

2) Analysis of the Multiscale Structure: To illustrate the rationality of the proposed overlapping multiscale structure, we construct a nonoverlapping structure for comparison, namely, MARF@3*. In MARF@3*, there are no intersection scales between each ARFs, e.g., three ARF focus on the clusters ranges $(0 \ m-3 \ m], (3 \ m-6 \ m], (6 \ m-\infty]$, respectively. Each scale forest is trained with clusters in the corresponding scales and predicts the legs independently. The results are shown in Table IV. MARF@3* has a large decline in performance compared with MARF@3, which proves the advantage of overlapping structures in the leg detection task. The main reason is that the overlapping structure shares training clusters between scales. This enriches the scale variety of training sets and contributes to the generalization of the decision tree and even the entire model.

3) Analysis of the Dichotomous Node: We build a variant of the adaptive-switch decision tree to illustrate the effectiveness of the proposed dichotomous node. In this variant, called MARF@3-P, we employ the probabilistic node proposed in [31] to replace the dichotomous node in MARF@3. The comparison is shown in Table IV. MARF@3 achieves 3.08% precision improvement, 3.01% recall improvement, and 22.9% FP decrease over MARF@3-P. The main reason is that clusters passed by the probabilistic nodes are difficult to go deeper, which generates more shallow-layer leaf nodes and leads to insufficient training. The average depth of leaf nodes is shown in Table V, which proves our inference.

### Table III

| Method  | TP    | FP    | Precision | Recall  |
|---------|-------|-------|-----------|---------|
| MRF     | 22674 | 4956  | 82.06%    | 86.07%  |
| ARF     | 22697 | 3782  | 85.72%    | 86.16%  |
| MARF@3  | 22955 | 3825  | 89.04%    | 87.14%  |

Label assignment threshold $\beta$ is set to 0.5.

### Table IV

| Method  | TP    | FP    | Precision | Recall  |
|---------|-------|-------|-----------|---------|
| MARF@2  | 22932 | 2893  | 88.80%    | 87.08%  |
| MARF@3  | 22955 | 2825  | 89.04%    | 87.14%  |
| MARF@4  | 22943 | 2917  | 88.72%    | 87.09%  |
| MARF@3* | 22843 | 4520  | 83.48%    | 86.71%  |
| MARF@3-P | 22427 | 3663  | 85.96%    | 85.13%  |

Label assignment threshold $\beta$ is set to 0.5.
E. Further Application on People Detection and Tracking

Due to the dominant effect of the leg detector in people detection and tracking, we further evaluate the proposed detector’s contribution to this task. In more detail, we apply MARF@3 to the Joint Leg [12] people tracking framework. The Joint Leg employs SRF to detect legs from 2-D laser scans at first. Then, they match legs as leg pairs and apply a heuristic method to detect people. Finally, the method uses a Kalman Filter to track people in consecutive laser scans. For reasonable evaluation and comparison, MARF@3 is substituted for SRF in Joint Leg to detect legs.

1) Quantitative Results: As shown in Table VI, our MARF-based system achieves the best performance in Correct, Miss, and FPP for people detection. In SCENARIO I, since MARF detects more long-distance people and achieves a noticeable improvement in correct people detection, ours achieves 4.3% Correct improvement and 2.3% Miss decrease over Joint Leg [12]. In SCENARIO II, contrasting to SRF generating more FP, our approach achieves 17.9% Correct improvement, 7.5% Miss decrease, and 47.4% FPP decrease over the Joint Leg [12]. In SCENARIO III, our MARF-based system achieves 1.8% Correct improvement and 68.7% Miss decrease over Joint Leg [12]. These results further prove that the robust classification capability of MARF conduces to detect people.

| Method   | Average Depth | Average Leaf Node Number |
|----------|---------------|--------------------------|
| MARF@3   | 17.3          | 264.6                    |
| MARF@3-P | 13.8          | 196.7                    |

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
TABLE VI
RESULTS ON PEOPLE TRACKING DATASET

| Dataset     | Tracker     | Correct | Miss | FPP  | MOTA     |
|-------------|-------------|---------|------|------|----------|
| SCENARIO I  | leg_detector[25] | 478     | 1605 | 28   | 19.2%    |
|             | Joint Leg [12] | 711     | 1372 | 9    | 33.2%    |
|             | MARF-based    | 743     | 1341 | 9    | 35.05%   |
| SCENARIO II | leg_detector[25] | 164     | 481  | 294  | -22.5%   |
|             | Joint Leg [12] | 165     | 480  | 97   | 10.2%    |
|             | MARF-based    | 201     | 444  | 51   | 23.01%   |
| SCENARIO III| leg_detector [25] | 15621   | 3425 | -    | -        |
|             | Joint Leg [12] | 18561   | 485  | -    | -        |
|             | MARF-based    | 18894   | 152  | -    | -        |

* Due to only part of people are labeled on the ground truth in DATASET III except tracked people, FP and MOTA are ignored [12].

For people tracking, benefiting from leg detection in consecutive laser data frames, robust people detection contributes to a higher MOTA score. MARF achieves 1.85% MOTA improvement in SCENARIO I and 12.81% MOTA improvement in SCENARIO II. Therefore, the results can further prove the advance of MARF.

2) Qualitative Results: Fig. 7 shows the qualitative result tested on the People Tracking dataset. The first row shows the scenario that only nonleg clusters occur nearby the laser scanner. The SRF-based method [12] mistakenly recognizes nonleg clusters as the people, while our MARF-based system and PRF-based system correctly predict it. The second row shows a person standing nearby the wall, and an ambiguous cluster is similar to the leg. The SRF-based method can hardly distinguish both leg and nonleg clusters. The PRF-based method correctly recognizes the people, however, it also induces the false-positive detections. Similarly, our MARF-based system can effectively avoid fake leg clusters and accurately determine people in the fifth scenario. The third row depicts the scenario with crowded people. Ours detects the most people, but there are still some challenges in such occluding conditions. The fourth row shows a long-distance people which the SRF-based and the PRF-based approaches can hardly detect. While our MARF-based system successfully detects these long-distance people. The fifth row shows that two persons that are far away from the robot. Although the two persons are near each other, our method successfully detects them all.

3) Efficiency and Speed Comparison: The efficiency of the algorithm is also concerned, especially for the application on low-cost robot platforms. We record time cost on a ThinkPad E550 laptop with CPU i5-5500U, Ubuntu 16.04LTS OS, and 8-GB RAM. Due to the dichotomous node, the prediction of each cluster is outputted by more leaf nodes in MARF@3 and is expected to cost more time. However, during leg detection, MARF@3 can still process laser scans at almost 62 Hz, satisfying the real-time detection requirement. Moreover, we also argue that an optimization of the implementation of the approach can further improve the running efficiency. While applying to the people detection and tracking framework [12], the MARF-based system achieves the same efficiency as the SRF-based approach, with a speed 7.2 Hz. It proves that data association for people detection and tracking is the computational bottleneck in current work. That is also the next work we aim to study and optimize in the future.

To clarify our method more comprehensively, we statistic the time cost of the proposed RF and the existing RF models, which is demonstrated in Table VII. It can be seen that the speed of MARF is relatively slower among all models. The reason is that, in PRF, each point cluster passes through 526 nodes on average, while 4091 nodes for MARF. This phenomenon is consistent with Fig. 2, namely, MARF is usually deeper than PRF, so there are more nodes of MARF involved in the calculation. In addition, SRF is implemented with OpenCV API, thus we cannot fairly compare MARF to SRF. In the future, how to obtain better performance with the same number of nodes is also a question worth studying.

F. Failure Cases

Fig. 8 illustrates several failure cases of our method. First, our method still cannot completely discover the legs with too few laser points. This issue is also a common difficulty for other leg detectors. Second, since the proposed method follows a single-leg detection pipeline that first segments the entire 2-D point cloud and then classifies each point cluster, it is inevitable for our model to classify some wrong-segmented clusters, such as a cluster indicating leg pair. These leg-pair clusters are so different from the single leg, thus they usually obtain very low confidence, as shown in the second row of Fig. 8. Third, another unavoidable issue is the false-positive detection of the static nonleg point cluster with leg shape in...
the scene. As shown in the third row of Fig. 8, these clusters belong to static objects at different distances. In general, these failure cases are basically due to the lack of information in the 2-D laser point cloud. In the future, we are going to introduce richer information to address this issue, e.g., combining multiple laser scans or incorporating consecutive laser scans to extract dynamic leg clusters.

VI. CONCLUSION

This article focuses on a new problem, i.e., the conflict between global and local confidence of cluster features in 2-D laser-based leg detection, and proposes an approach to alleviate this problem. Moreover, a MARF is proposed to enhance the classification ability of multiscale leg clusters. We carefully evaluate our approach on the dataset and further apply it to people detection and tracking. Expanding experiments and comparative analysis validates the effectiveness of the proposed method.

REFERENCES

[1] C.-H. Lin and K.-T. Song, “Probability-based location aware design and on-demand robotic intrusion detection system,” IEEE Trans. Syst., Man, Cybern., Syst., vol. 44, no. 6, pp. 705–715, Jun. 2014.

[2] C. Ye, “Navigating a mobile robot by a traversability field histogram,” IEEE Trans. Syst., Man, Cybern., B, Cybern., vol. 37, no. 2, pp. 361–372, Apr. 2007.

[3] J. Yuan, S. Zhang, Q. Sun, G. Liu, and J. Cai, “Laser-based intersection-aware human following with a mobile robot in indoor environments,” IEEE Trans. Syst., Man, Cybern., Syst., vol. 51, no. 1, pp. 354–369, Jan. 2021.

[4] C. J. Pai, H.-R. Tyan, Y.-M. Liang, H.-Y. Liao, and S.-W. Chen, “Pedestrian detection and tracking at crossroads,” Pattern Recognit., vol. 37, no. 5, pp. 1025–1034, 2004.

[5] R. Triebel et al., SPENCER: A Socially Aware Service Robot for Passenger Guidance and Help in Busy Airports, Cham, Switzerland: Springer, 2016, pp. 607–622.

[6] H. Zhang, C. Reardon, and L. E. Parker, “Real-time multiple human perception with color-depth cameras on a mobile robot,” IEEE Trans. Cybern., vol. 43, no. 5, pp. 1429–1441, Oct. 2013.

[7] O. S. Gedik and A. A. Alatan, “3-D rigid body tracking using vision and depth sensors,” IEEE Trans. Cybern., vol. 43, no. 5, pp. 1395–1405, Oct. 2013.

[8] A. Kod, A. Howard, and M. J. Mataric, “A laser-based people tracker,” in Proc. IEEE Int. Conf. Robot. Autom., (ICRA), 2002, pp. 3024–3029.

[9] L. Beyer, A. Hermans, T. Linder, K. O. Arras, and B. Leibe, “Deep person detection in two-dimensional range data,” IEEE Robot. Autom. Lett., vol. 3, no. 3, pp. 2726–2733, Jul. 2018.

[10] H. Zhao and R. Shibasaki, “A vehicle-borne urban 3-D acquisition system using single-row laser range scanners,” IEEE Trans. Syst., Man, Cybern., B, Cybern., vol. 33, no. 4, pp. 658–666, Aug. 2003.

[11] W. Chung, H. Kim, Y. Yoo, C.-B. Moon, and J. Park, “The detection and following of human legs through inductive approaches for a mobile robot with a single laser range finder,” IEEE Trans. Ind. Electron., vol. 59, no. 8, pp. 3156–3166, Aug. 2012.

[12] A. Leigh, J. Pineau, N. Olmedo, and H. Zhang, “Person tracking and following with 2D laser scanners,” in Proc. IEEE Int. Conf. Robot. Autom. (ICRA), 2015, pp. 726–733.

[13] D. Li, L. Li, Y. Li, F. Yang, and X. Zuo, “A multi-type features method for leg detection in 2-D laser range data,” IEEE Sensors J., vol. 18, no. 4, pp. 1675–1684, Feb. 2018.

[14] G. Borges and M.-J. Aldon, “Line extraction in 2D range images for mobile robotics,” J. Intell. Robot. Syst., vol. 40, pp. 267–297, Jul. 2004.

[15] R. E. Schapire, “A brief introduction to boosting,” in Proc. 16th Int. Joint Conf. Artif. Intell., 1999, pp. 1401–1406.

[16] T. Linder, S. Breuers, B. Leibe, and K. O. Arras, “On multi-modal people tracking from mobile platforms in very crowded and dynamic environments,” in Proc. IEEE Int. Conf. Robot. Autom. (ICRA), 2016, pp. 5512–5519.

[17] L. Spinello and R. Siegwart, “Human detection using multimodal and multidimensional features,” in Proc. IEEE Int. Conf. Robot. Autom. (ICRA), 2008, pp. 3264–3269.

[18] K. O. Arras, O. M. Mozos, and W. Burgard, “Using boosted features for the detection of people in 2D range data,” in Proc. IEEE Int. Conf. Robot. Autom. (ICRA), 2007, pp. 3402–3407.

[19] H. T. Duong and Y. S. Suh, “Human gait tracking for normal people and walker users using a 2D LIDAR,” IEEE Sensors J., vol. 20, no. 11, pp. 6191–6199, Jun. 2020.

[20] J. Xavier, M. Pacheco, D. Castro, A. E. Ruano, and U. Nunes, “Fast line, arc/circle and leg detection from laser scan data in a player-driven,” in Proc. IEEE Int. Conf. Robot. Autom. (ICRA), 2005, pp. 3930–3935.

[21] E. A. Topp and H. I. Christensen, “Tracking for following and passing persons,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS), 2005, pp. 2321–2327.

[22] J. Cui, H. Zha, H. Zhao, and R. Shibasaki, “Tracking multiple people using laser and vision,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS), 2005, pp. 235–280.

[23] C. Weinrich, T. Wengefeld, C. Schröter, and H. M. Gross, “People detection and distinction of their walking aids in 2D laser range data based on generic distance-invariant features,” in Proc. RO-MAN IEEE Int. Symp. Robot & Human Interact. Commun., 2014, pp. 767–773.

[24] K. O. Arras et al., “Range-Based People Detection and Tracking for Socially Enabled Service Robots,” Heidelberg, Germany: Springer, 2012, pp. 235–280.

[25] D. Lu and W. D. Smart, “Towards more efficient navigation for robots and humans,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS), 2013, pp. 1707–1713.

[26] D. Cha and W. Chung, “Human-leg detection in 3D feature space for a person-following mobile robot using 2D lidars,” Int. J. Precis. Eng. Manuf., vol. 21, no. 14, pp. 1299–1307, 2020.

[27] L. Beyer, A. Hermans, and B. Leibe, “DROW: Real-time deep learning-based wheelchair detection in 2-D range data,” IEEE Robot. Autom. Lett., vol. 2, no. 2, pp. 585–592, Apr. 2017.

[28] R. J. Lewis, “An introduction to classification and regression tree (CART) analysis,” in Proc. Annu. Meeting Soc. Acad. Emerg. Med., 2000.

[29] A. Sen, M. M. Islam, K. Murase, and X. Yao, “Binarization with boosting and oversampling for multiclass classification,” IEEE Trans. Cybern., vol. 46, no. 5, pp. 1078–1091, May 2016.

[30] A. González, D. Vázquez, A. M. López, and J. Amores, “On-board object detection: Multicue, multimodal, and multiview random forest of local experts,” IEEE Trans. Cybern., vol. 47, no. 11, pp. 3980–3990, Nov. 2017.

[31] I. Reis, D. Baron, and S. Shahaf, “Probabilistic random forest: A machine learning algorithm for noisy data sets,” Astron. J., vol. 157, no. 1. p. 16, 2019.

[32] C. R. Qi, Y. Li, S. Hao, and L. J. Guibas, “PointNet++: Deep hierarchical feature learning on point sets in a metric space,” in Proc. Conf. Neural Inf. Process. Syst. (NIPS), 2017, pp. 5009–5108.

[33] T. T. Ma, Q. Wang, P. Li, and W. Zuo, “Multi-scale structural kernel representation for object detection,” Pattern Recognit., vol. 110, Feb. 2021, Art. no. 107593.

[34] K. Bernardin and R. Stiefelhagen, “Evaluating multiple object tracking performance: The CLEAR mot metrics,” Eurasip J. Image Video Process., vol. 2008, no. 1, 2008, Art. no. 246309.

[35] N. Bellotto and H. Hu, “Multisensor-based human detection and tracking for mobile service robots,” IEEE Trans. Syst., Man, Cybern., B, Cybern., vol. 39, no. 1, pp. 167–181, Feb. 2009.