Monocular Person Tracking and Identification with On-line Deep Feature Selection for Person Following Robots

Kenji Koide[a], Jun Miura[b], Emanuele Menegatti[c]

[a]*The Department of Information Technology and Human Factors, National Institute of Advanced Industrial Science and Technology, Japan*
[b]*The Department of Computer Science and Engineering, Toyohashi University of Technology, Japan*
[c]*The Department of Information Engineering, the University of Padova, Italy*

Abstract

This paper presents a new person tracking and identification framework based on solely a monocular camera. In this framework, we first track persons in the robot coordinate space using Unscented Kalman filter with the ground plane information and human height estimation. Then, we identify the target person to be followed with the combination of Convolutional Channel Features (CCF) and online boosting. It allows us to take advantage of deep neural network-based feature representation while adapting the person classifier to a specific target person depending on the circumstances. The entire system can be run on a recent embedded computation board with a GPU (NVIDIA Jetson TX2), and it can easily be reproduced and reused on a new mobile robot platform. Through evaluations, we validated that the proposed method outperforms existing person identification methods for mobile robots. We applied the proposed method to a real person following robot, and it has been shown that CCF-based person identification realizes robust person following in both indoor and outdoor environments.

Keywords: person tracking, person identification, mobile robot

1. Introduction

There is an increasing demand for service robots which can follow a person. Such robots have been expected to be common in the next decade for supporting people in daily tasks, to name a few: guiding, guarding, and elderly care. To follow a person, robots have to be able to robustly track the position of the target person. While following a person, it often happens that the robot cannot keep tracking the person since he/she moves out from the robot's sensor view, or is occluded by other persons. In such cases, to resume the tracking and following, the robot has to re-identify the person using a target person model learned before losing the track.

In the past, a number of works proposed person tracking and identification frameworks for person following robots. Most of those works require a range sensor, such as Laser Range Finder (LRF) [1, 2, 3], stereo camera [4, 5], and infrared RGB-D camera [6]. However, LRFs and stereo cameras are typically unaffordable, and infrared RGB-D cameras cannot be used in outdoor environments. We believe that the lack of a person tracking and identification framework with an affordable monocular camera prevents service robots to be exploited for daily tasks, and it triggered us to propose a complete monocular vision-based framework for person following robots.

In this paper, we propose a person tracking and identification framework for mobile robots which relies on solely an affordable and common monocular camera (see Fig. 1). In this framework, we first detect persons with *OpenPose*, a deep neural network-based skeleton detector [7]. Then, inspired by [8, 9], we estimate the positions and heights of persons in the robot space rather than in the image space based on the ground plane information. It allows us to reliably track persons and know their positions in the real space which is the most fundamental information for person following. For instance, with the estimated person position in the robot space, we can easily make the robot keep the distance to the person constant to follow him/her. Then, a person identification method based on the combination of Convolutional Channel Features [10] and online boost-

Email addresses: k.koide@aist.go.jp (Kenji Koide), jun.miura@tut.jp (Jun Miura), emg@dei.unipd.it (Emanuele Menegatti)

A supplementary video is available at: https://youtu.be/SsIrXxnOgaQ

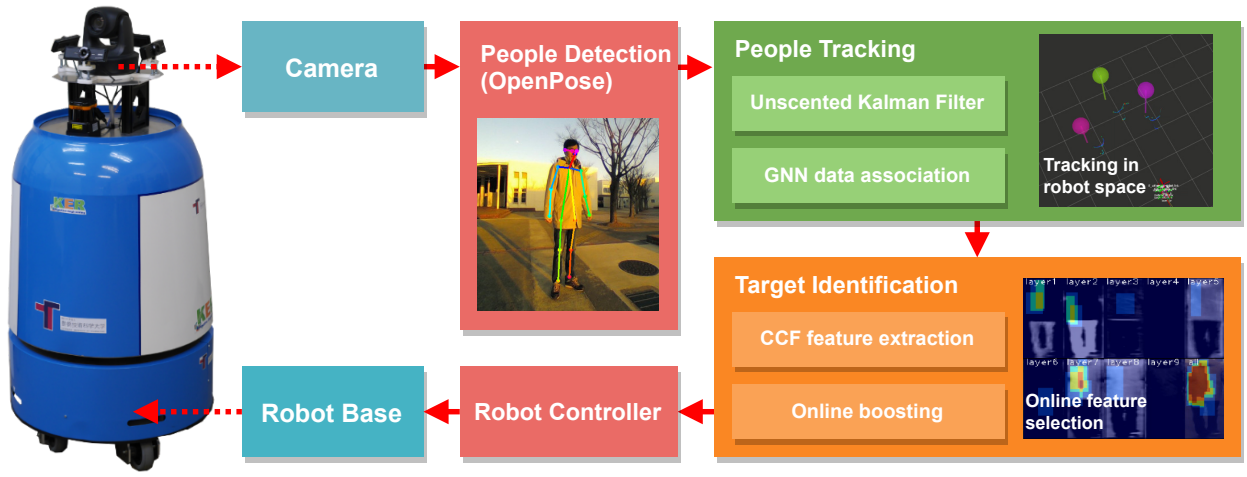


Figure 1: The proposed person tracking and identification framework with a monocular camera.

ing [11] runs on the top of the tracking module. It attentively learns the appearance of a specific target person based on the deep neural network-based discriminative features. If the robot loses the track of the target person, it re-identifies the target person among surrounding persons with the online learned appearance model. The entire system is designed such that it can run on an affordable embedded computer with a GPU (NVIDIA Jetson TX2) in real-time. The use of this common computing board allows us to easily reproduce and reuse the system on a new mobile robot platform.

The contributions of this paper are three-fold. First, we propose a robust vision-based people tracking method which employs the ground plane information and human height estimation to estimate the people positions in the robot space. It allows us to reliably track persons and control the robot with respect to a person easily. Second, we present a comprehensive evaluation of the CCF-based online person identification method, which was originally proposed in [12]. Through evaluations, it has been confirmed that the proposed system achieves a state-of-the-art performance. Third, the source code of the proposed framework is available from a public repository [1]. The system can easily be reproduced and reused on a new mobile robot platform.

The rest of the paper is organized as follows. Sec. 2 describes related work. Sec. 3 and Sec. 4 describe the proposed vision-based people tracking method and the CCF-based online person identification method, respectively. Sec. 5 presents evaluations of the person tracking method and the person identification method. Sec. 6 shows a person following experiment conducted to demonstrate that the proposed method can be applied to real person following robots. Sec. 7 concludes the paper.

2. Related work

2.1. People tracking for mobile robots

A number of people tracking methods for mobile robots have been proposed in the past. Most of these works exploit range sensors such as, laser range finders (LRFs) [1, 13], infrared RGB-D cameras [6, 14], and stereo cameras [5, 4, 15], to make the person detection easy. Such range sensors provide persons' positions accurately as long as they are visible from the sensors and promise reliable person tracking capabilities. However, such range sensors are typically unaffordable (LRFs and stereo cameras) or not available in outdoor environments (RGB-D cameras).

Several works tackled monocular vision-based people tracking for mobile robots. Zhang et al. proposed a method which tracks people in the image space, and controls the robot using a visual servo technique [16]. However, from the view point of robot control, it is desirable to estimate a person's position in the robot space rather than the image space. The challenge here is that, with a monocular camera, it is impossible to estimate the distance to the person without any prior knowledge. Choi and Savarese [8] proposed a method to track and estimate object positions, ground plane features, and camera intrinsic parameters simultaneously. The ground plane features help to estimate the camera

[1] https://github.com/koide3/monocular_person_following

parameters and person trajectories in the real space robustly. Ardiyanto and Miura [9] proposed a method which tracks people in the real space with Unscented Kalman filter based on the human height information. Such people tracking in the real space could be more robust than tracking in the image space, because we can take advantage of assumptions on the people motion in the space where they are actually moving on [8]. However, they lack the capability of person identification. In daily situations, there are sometimes several persons around the robot, and the robot may loose the track of the target person due to occlusion, or it tracks a wrong person when the target is close to another one. In such cases, the robot has to re-identify the person based on a target person model learned before the occlusion to resume tracking and following the target person.

To our knowledge, only a few works proposed monocular camera-based person following robots with person identification capabilities. Bakar and Saad proposed a specific person detection method for mobile robots [17]. However, since this method requires to put several markers on the target person, it cannot be applied to non-cooperative scenarios. Yao et al., proposed a face detection-based person tracking system for a miniature robotic blimp [18]. They estimate a person's position based on the detected face pose. Thus, the face of the person has to be always visible to the robot in this system.]

2.2. Person re-identification

Person re-identification has been widely investigated for camera networks for surveillance and monitoring, and several features, such as gait [19], height [20], and skeletal information [21], have been proposed. In cases of mobile robots, the most standard feature for person re-identification is the appearance, such as color and texture of clothes, since it can easily be obtained from a mobile robot. It has been proven that the combination of appearance features and an online learning method works very well for the person following task [5, 14, 22]. Online learning methods allow us to adapt the person model to a specific target person. For instance, when there are persons wearing similar shirts and dissimilar trousers, online learning methods can focus on the discriminative part, trousers in this case, to re-identify the target person robustly. However, the most of existing methods for mobile robots use naive hand-crafted appearance features, such as Haar-like features [14], Local Binary Patterns (LBP) [5], edge features [22] on color and depth images. They are not dedicated features for person re-identification, and they may not be discriminative when persons are wearing similar clothes.

Recently, deep neural networks have been successfully applied to various vision applications. Person re-identification is one of such applications, and Convolutional Neural Network (CNN) based methods outperform traditional systems [23, 24]. However, a few works [15] applied such CNN-based methods to mobile robots due to the limitation of computation resource on mobile robots. On a mobile robot, it is not always feasible to use a high performance GPU, and thus, it is hard to directly apply such CNN-based methods to person following robots. Moreover, in person following tasks, it is important to adapt the person model to the target person online. Without an online learning approach, it is sometimes hard to distinguish persons wearing similar clothes even with a deep neural network. Although there are methods to update neural networks online [25], those methods are very costly, and it is not feasible to run it on a mobile robot.

Yang et al. proposed Convolutional Channel Features (CCF) [10]. In this technique, they take the first a few convolution layers from a trained deep CNN, and use the set of convolution layers as a feature extractor. By training light-weight models, such as SVM and boosting, with the deep feature representation, they adapt the framework to several tasks without expensive tuning of the network. Following their work, in this paper, we introduce CCF to person identification for mobile robots to take advantage of deep representation while keeping the processing cost low.

3. People tracking

As the starting point of the people tracking process, we use *OpenPose*, a deep convolutional neural network-based human detector [7]. It provides the position of each joint of persons in the image space. We utilize an implementation of *OpenPose* which is sped up with mobilenet architecture [26] with depth-wise separable convolution filters [2]. Then, inspired by [8] and [9], we track persons in the robot coordinate space based on the detected joint positions. Tracking persons in the real space could be more robust than tracking in the image space, since we can take advantage of motion assumptions on where the persons are actually moving on [8]. In addition to that, person positions in the robot space are very useful for service robots to interact with them. For instance, with the estimated position in the robot

[2] `https://github.com/ildoonet/tf-pose-estimation`

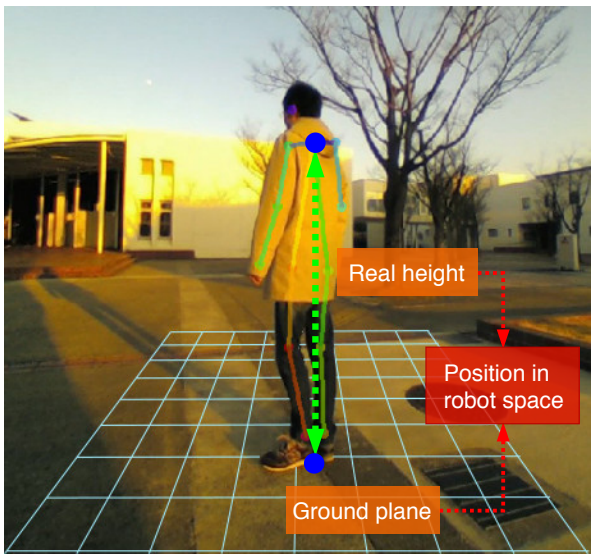


Figure 2: The proposed tracking method takes advantage of the ground plane information.

space, we can easily control the robot so that it keeps the distance to the person constant while avoiding other persons.

Fig. 2 illustrates the proposed tracking method. We assume that the camera pose with respect to the ground plane is calibrated beforehand. By projecting a detected ankle position onto the ground plane, we can estimate the person position in the robot space. However, while a person is walking, the ankle position varies due to the walking motion, and it would affect the position estimation. We, thus, simultaneously estimate the height of the person in addition to the position based on neck and ankle detections using Unscented Kalman Filter (UKF) [27] to make the estimation robust. Once the real height of the person is estimated, by comparing it with the height in the image space, we can estimate the distance to the person. It would contribute to the estimation accuracy when the ankle position varies largely (i.e., when the person is walking). Furthermore, when the real height is available, we can update the UKF with only a neck detection when the ankle is not visible to the camera.

3.1. State estimation

We define a state space to be estimated as $\boldsymbol{x}_t = [\boldsymbol{p}_t, \boldsymbol{v}_t, h_t]^T$ consisting of the position, velocity, and height of a person. To compensate for the robot motion, we estimate the state in the *world* frame and transform it in the robot frame using odometry. With UKF, we estimate the state from observations of neck and ankle positions in the image space $\hat{\boldsymbol{z}}_t = [\boldsymbol{p}_t^{neck}, \boldsymbol{p}_t^{ankle}]^T$.

Assuming the constant velocity model, the system function f to update the state is defined by:

$$f(\boldsymbol{x}_t) = \boldsymbol{x}_{t+1} = [\boldsymbol{p}_t + \Delta t \cdot \boldsymbol{v}_t, \boldsymbol{v}_t, h_t]^T, \quad (1)$$

where Δt is the duration between $t+1$ and t. The observation function h is defined by:

$$h(\boldsymbol{x}_t) = \boldsymbol{z}_t = [\mathrm{Proj}(\boldsymbol{p}_t + [0, 0, h_t]^T), \mathrm{Proj}(\boldsymbol{p}_t)]^T, \quad (2)$$

where the function Proj is the pinhole camera projection function. When only a neck position is observed, we use the observation function without the ankle observation term to update the state:

$$h'(\boldsymbol{x}_t) = \boldsymbol{z}'_t = [\mathrm{Proj}(\boldsymbol{p}_t + [0, 0, h_t]^T)]^T. \quad (3)$$

3.2. Data association

To associate track instances and joint detections at a frame, we first calculate the expected observation distribution (neck and ankle positions distribution) of each track using Unscented Transform [27]:

$$\mu_t^z, \sigma_t^z = UT(\mu_t^{\boldsymbol{x}_t}, \sigma_t^{\boldsymbol{x}_t}, h). \quad (4)$$

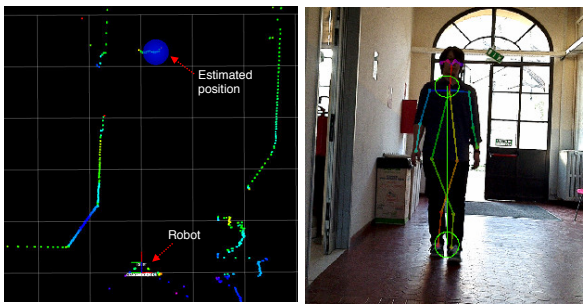
μ_t^z and σ_t^z are the expected observation distribution, $\mu_t^{\boldsymbol{x}_t}$ and $\sigma_t^{\boldsymbol{x}_t}$ are the distribution of the state $\boldsymbol{x}_t$, h is the observation function, and the function UT is the Unscented Transform function.

Then, we define the distance between a track and an observation as:

$$Dist(track_i, obs_j) = \begin{cases} \infty, & \text{if } D_M(\mu_t^z, \sigma_t^z, \hat{\boldsymbol{z}}_t) > th_{gate} \\ -\mathcal{N}(\mu_t^z, \sigma_t^z, \hat{\boldsymbol{z}}_t), & \text{otherwise} \end{cases} \quad (5)$$

, where D_M is the Mahalanobis distance function, and th_{gate} is the threshold for gating. Based on this distance function, we associate tracks and detections using the global nearest neighbor association [28]. Note that in the data association algorithm, a constant is added to the calculated distances to make them positive.

Fig. 3 shows a tracking result. The blue sphere in Fig. 3 (a) shows the estimated person position in the robot space, and the green ellipses in Fig. 3 (b) indicate the neck and ankle positions distribution calculated from the person state in the robot space.



(a) In the robot space. (b) In the image space.

Figure 3: A tracking result. The ellipses on the right image show the expected neck and ankle positions distribution calculated from the person position in the robot space. Note that the laser is used for only validation.

4. Person identification

4.1. Convolutional channel features

The tracking module outputs the neck and the ankle positions of the tracked persons, and the following identification module uses them to calculate the ROIs for extracting appearance features for identification.

In this work, we use a convolutional filter-based person identification method proposed in [12]. To take advantage of deep CNN-based feature representation, this method employs Convolutional Channel Features (CCF) [10] instead of traditional appearance features which have been used for mobile robots, such as color histograms [3], haar-like [1], and edge features [22]. CCF consists of a few convolutional layers taken from a trained deep CNN. It takes an input image and yields a set of response maps (i.e. feature maps) which are optimized for a specific task, such as person detection and classification.

We train Ahmed's network for person re-identification [23] as the base of CCF, and use the first two convolution filters of the network to extract appearance features for online person identification (see Fig.4). Ahmed's network takes a pair of person images and then applies convolution filters to extract feature maps for each input image. The extracted feature maps are compared together by taking the difference between each pixel in a feature map and the neighbor pixels of the corresponding pixel in the other map. Then, it applies convolution filters again to the difference map, and through a linear layer, the network judges whether the input images are the same person or not. The numbers of filters in the first and the second convolution layers are 20 and 25, and thus, they yield 25 feature maps. Since it may be costly for mobile systems to directly use this network, we also trained a tiny version of the network, where the numbers of convolution filters in both the first and the second layers are 10. We trained both the networks with a dataset consisting of CUHK01 [29] and CUHK03 [30]. The total number of identities in the dataset is about 2300, and the number of images is about 17000. We used nine tenths of the dataset for training and the rest for testing and confirmed that both the networks show over 98% of identification accuracy on the test set. In the rest of this paper, the CCFs taken from the original and the tiny version networks are denoted as CCF25 and CCF10, respectively.

Fig. 5 shows example feature maps extracted by CCF10. We can see that each filter shows strong responses for different color properties. For instance, filter 2 shows higher values on darker and blue regions, while filter 8 strongly responds orange regions. We can obtain diverse feature representation using CCF without hand-crafting, and such features would contribute to identification performances.

4.2. Online boosting-based person classifier

With the offline trained CCF, we extract feature maps from person images, and then train a target person classifier online. Following Luber's work [14], we employ online boosting [11] to construct the classifier. Online boosting constructs an ensemble of weak classifiers and uses it as a strong classifier. In this work, each weak classifier takes the sum of pixel values in a random rectangle region on a feature map and classifies images into the target and other persons using a naive Bayes classifier. Since online boosting selects the weak classifiers with the best classification accuracy, discriminative regions are automatically chosen for identification. In this work, we use online boosting with 10 weak classifier selectors, and each selector contains 15 weak classifiers. Thus, the total number of weak classifiers is 150, and 10 out of them are selected to construct an ensemble. Fig. 6 shows an example of the features selected by online boosting. We can see that online boosting automatically selects the discriminative regions, the upper body regions in this case, to construct a classifier ensemble.

5. Evaluation

5.1. Tracking accuracy evaluation

To evaluate the accuracy of the proposed tracking method, we recorded three image sequences with the robot shown in Fig. 7. A Jetson TX2 development board with an embedded monocular camera module is mounted on the robot. The height of the camera from

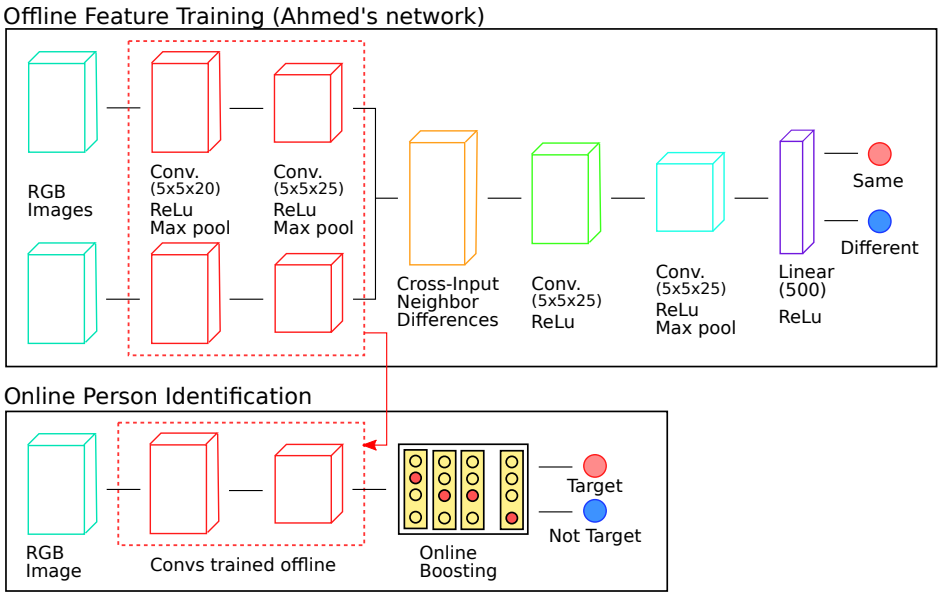


Figure 4: Convolutional Channel Features-based person identification framework. We take the first two layers of a network for person re-identification and use them to extract features for online person identification.

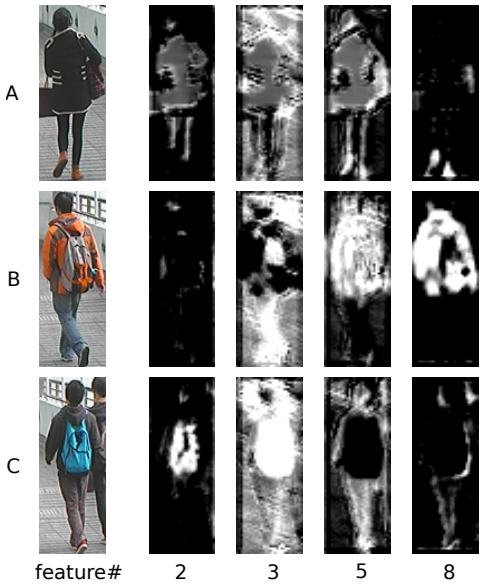
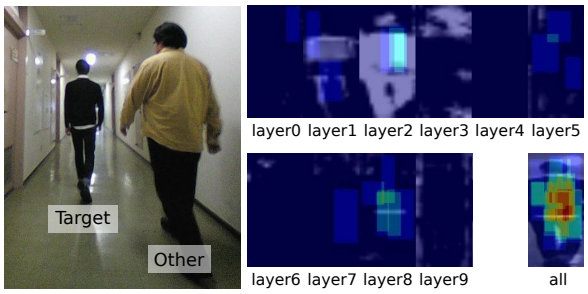


Figure 5: Feature maps extracted by CCF10. Each filter shows strong responses for different color properties.



(a) A snapshot of the environment.

(b) Features selected by online boosting. The selected regions are superimposed on the feature maps.

Figure 6: An example of features selected by online boosting. The discriminative regions, the upper body regions in this case, are automatically selected.

the ground is 0.6 [m], and the horizontal field of view is about 70 [deg]. The camera pose with respect to the ground plane is calibrated by observing a chessboard pattern put on the ground. For validation, a laser range finder is also mounted on the robot, and we estimate the person position with the proposed method and a laser-based people tracking method [3]. We consider the laser-based result as the ground truth in this evaluation.

We define the error between trajectories as follows:

$$\boldsymbol{p}'_i = \arg\min_{\boldsymbol{p}^l_j \in \mathcal{L}} \left\| \boldsymbol{p}^v_i.t - \boldsymbol{p}^l_j.t - \Delta t \right\|, \tag{6}$$

$$E = \sum_{\boldsymbol{p}^v_i \in \mathcal{V}} \left\| \text{transform}(\boldsymbol{p}^v_i, \Delta x, \Delta y, \Delta\theta) - \boldsymbol{p}'_i \right\|, \tag{7}$$

where, $\mathcal{V} = [\boldsymbol{p}^v_0, \boldsymbol{p}^v_1, \cdots, \boldsymbol{p}^v_N]$, $\mathcal{L} = [\boldsymbol{p}^l_0, \boldsymbol{p}^l_1, \cdots, \boldsymbol{p}^l_M]$ are the trajectories measured by the vision and the laser-based methods, $\boldsymbol{p}_i = [x, y, t]$ is a point with timestamp,

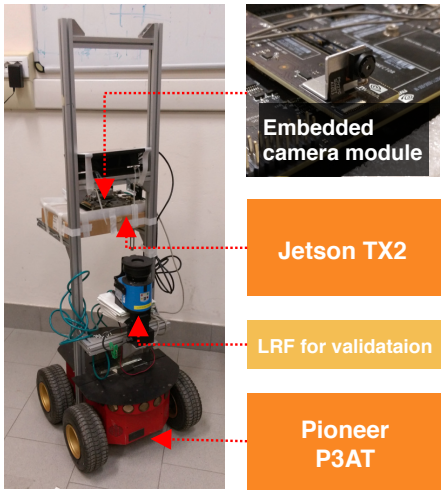
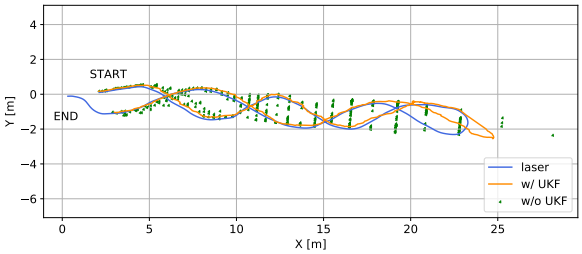
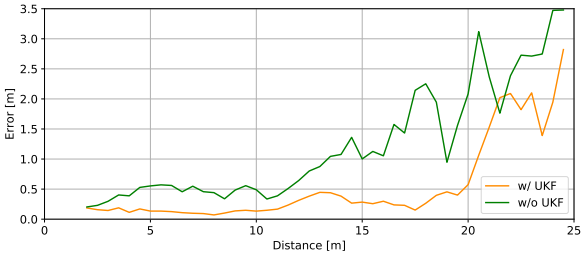


Figure 7: A mobile robot equipped with a Nvidia Jetson TX2 development board. An embedded camera module is bundled with the development board. An LRF is also mounted on the robot for the validation of the tracking system.



(a) Estimated trajectories



(b) Localization error vs distance

Figure 8: The tracking accuracy evaluation result.

and $\Delta x, \Delta y, \Delta\theta, \Delta t$ are the offset between the vision and the laser sensors. $\boldsymbol{p}'_i \in \mathcal{L}$ is the point with the timestamp closest to the timestamp of a point in $\mathcal{V}$. To align the vision-based trajectory with the laser-based one, we fist estimate the offset such that eq. 7 is minimized. Then, we calculate the error between the trajectories with the estimated offset.

Fig. 8 (a) shows an example of the estimated trajectories. To show the effect of the UKF-based tracking, the result without the UKF (projecting the ankle position onto the ground plane directly) is also shown in the figure. We can see that, without the UKF, the estimated position fluctuates due to the varying ankle positions during walking. We can also notice that the result without the UKF suffers from the "quantization effect" as the distance to the person gets larger. Although the image itself is quite large (1920×1080 [pix]), when the person is 20 [m] away from the camera, the person region in the image is only 100 [pix] in height. Furthermore, the image is down-scaled (1/3 scaling) in the detection process of *OpenPose*. With this image size, the method cannot preciously detect ankle positions, and thus it does not work well for distant persons.

With the UKF, the estimated trajectory is greatly improved thanks to the height information and the motion assumption. The UKF gets rid of the quantization effect and makes the trajectory smooth. Although the system tends to overestimate the distance to the person when the distance is larger than 20 [m], the trajectory within this distance matches with the ground truth well. During the trials, we confirmed that the minimum detection range of the system is about 2 [m] in this setting.

Fig. 8 (b) shows the plot of the localization error versus the distance between the camera and the person calculated from all the recorded sequences. We can see that, with the UKF, the position estimation accuracy is significantly improved. The error is smaller than 0.5 [m] in the range between $2 \sim 20$ [m], although it gets worse when the distance is larger than 20 [m]. Since the distance between a robot and a target person does not get larger than 10 [m] in usual service robot scenarios, we consider that these detection range and estimation accuracy are suitable for service robot systems.

5.2. *Person identification evaluation*

To evaluate the proposed person identification framework, we created a dataset consisting of a set of RGB image sequences taken from a mobile robot (shown in Fig. 1). For validation, we also recorded LRF data in addition to the images in this dataset. Fig. 9 shows snapshots of the dataset. We controlled the robot manually and made it follow a target person in indoor and outdoor environments. We collected six sequences, and two of them are recorded in indoor, and the rest are recorded in outdoor environments. In each sequence, a target person to be followed stands in front of the robot for the first seconds so that the robot can learn the appearance of the person, and then he/she starts walking. During the recording, the target person is often occluded by other persons so he/she becomes invisible from the robot, and the robot loses track of him/her.

(a) Sequence 1 (53 [sec]) (b) Sequence 2 (60 [sec]) (c) Sequence 3 (133 [sec])

(d) Sequence 4 (175 [sec]) (e) Sequence 5 (122 [sec]) (f) Sequence 6 (58 [sec])

Figure 9: Snapshots of the dataset for person identification evaluation in person following tasks. The dataset consists of RGB images and LRF data recorded from a mobile robot. The robot was manually controlled and following a person in indoor and outdoor environments.

Figure 10: The scene where the proposed system failed to detect the target person. The distance to the person is about 10 [m].

We evaluate the proposed monocular person identification framework with CCF10 on this dataset. To compare the proposed system with a laser-based system, we also run our previous system [12]. This system first detects people using the combination of a range data clustering technique and an SVM-based torso classifier [1], then it identifies the target person based on CCF10 and CCF25 extracted from the human regions calculated from the detected people positions. The main difference between this system and the proposed one is the detection part. The laser-based people detector used in this system allows to reliably detect distant people with a high recall rate.

For comparison, we also run a variation of [14] on the dataset. This method extracts Haar-like features and *Lab* color histograms from color and depth images acquired from an RGB-D camera, and constructs a person classifier using online boosting. Since we use a monocular camera only in this evaluation, we construct the classifier without depth images.

Table 1 shows a summary of identification results. To assess the identification performance, we categorize identification results in four states. CT (Correctly Tracked) means that the target was visible from the robot and correctly identified. CL (Correctly Lost) means that the target was invisible from the robot due to occlusion, and the system correctly judged that he/she is not in the view. WT (Wrongly Tracked) means the robot identified a wrong person as the target while the target was invisible, and WL (Wrongly Lost) means the robot judged that the target is not visible, although he/she was actually visible from the robot.

CCF-based methods outperform the traditional appearance feature-based method thanks to the robust deep feature representation. Even in sequences where clothes of the target and the others are similar, they correctly identified the target while the traditional one identified wrong persons as the target.

On the laser-based system, CCF10 and CCF25 show comparable results. However, in a few sequences, CCF25 failed to keep identifying the target person. For instance, it identified a wrong person as the target in sequence 3 and failed to re-identify the target after occlusion in sequence 4. We consider that this is due to the limitation of the feature selection of online boosting. Online boosting selects the best classifiers among a limited number of weak classifiers. When the feature space is vast, the set of weak classifiers cannot cover enough feature space, and thus, online boosting would fail to select discriminative features. The performance of CCF25 could be improved by increasing the number of weak classifiers. However, it increases the process-

Table 1: Person identification evaluation result. Bold indicates best results.

		Duration [sec]			
Sensors		LRF + Camera			Camera
Features		Haar Lab [14] *	CCF10 [12]	CCF25 [12]	CCF10 (Proposed)
Seq. 1	CT	38.78 (73.23%)	**40.84 (77.11%)**	37.96 (71.69%)	38.85 (73.36%)
	CL	6.62 (12.49%)	6.78 (12.80%)	**7.37 (13.92%)**	6.21 (11.72%)
	WT	3.91 (7.38%)	3.75 (7.08%)	**3.16 (5.96%)**	4.32 (8.17%)
	WL	3.65 (6.90%)	**1.59 (3.01%)**	4.47 (8.44%)	3.58 (6.76%)
Seq. 2	CT	43.76 (73.78%)	43.86 (73.95%)	**43.87 (73.97%)**	35.83 (60.40%)
	CL	11.28 (19.02%)	10.76 (18.14%)	**10.90 (18.37%)**	9.58 (16.15%)
	WT	**2.52 (4.24%)**	3.04 (5.12%)	2.90 (4.89%)	4.22 (7.11%)
	WL	1.76 (2.96%)	1.65 (2.79%)	**1.64 (2.77%)**	9.68 (16.33%)
Seq. 3	CT	48.08 (36.11%)	**106.31 (79.84%)**	88.60 (66.55%)	100.85 (75.75%)
	CL	7.67 (5.76%)	**20.18 (15.16%)**	19.67 (14.77%)	19.60 (14.72%)
	WT	46.45 (34.89%)	**3.94 (2.96%)**	6.47 (4.86%)	4.52 (3.40%)
	WL	30.94 (23.24%)	**2.71 (2.04%)**	18.40 (13.82%)	8.17 (6.13%)
Seq. 4	CT	37.89 (21.56%)	141.19 (80.33%)	85.60 (48.70%)	**143.45 (81.56%)**
	CL	24.83 (14.13%)	23.18 (13.19%)	21.57 (12.27%)	**21.95 (12.48%)**
	WT	12.08 (6.88%)	**5.83 (3.32%)**	6.30 (3.58%)	7.06 (4.02%)
	WL	100.95 (57.44%)	5.56 (3.16%)	62.29 (35.44%)	**3.42 (1.95%)**
Seq. 5	CT	98.33 (80.38%)	98.75 (80.73%)	**98.89 (80.84%)**	98.59 (80.59%)
	CL	16.66 (13.62%)	**18.39 (15.03%)**	18.36 (15.00%)	16.19 (13.24%)
	WT	5.12 (4.19%)	**3.32 (2.71%)**	3.38 (2.76%)	5.52 (4.51%)
	WL	2.22 (1.81%)	1.88 (1.53%)	**1.70 (1.39%)**	2.04 (1.67%)
Seq. 6	CT	33.10 (59.67%)	41.90 (75.55%)	**43.67 (78.74%)**	41.66 (75.11%)
	CL	2.68 (4.84%)	**9.01 (16.24%)**	**9.01 (16.24%)**	7.27 (13.11%)
	WT	16.80 (30.28%)	**0.06 (0.11%)**	**0.06 (0.11%)**	1.80 (3.24%)
	WL	2.88 (5.20%)	4.49 (8.10%)	**2.73 (4.91%)**	4.73 (8.53%)
Total	CT	299.94 (50.08%)	**472.86 (78.94%)**	398.60 (66.55%)	459.23 (76.65%)
	CL	69.75 (11.64%)	**88.29 (14.74%)**	86.87 (14.50%)	80.80 (13.49%)
	WT	86.89 (14.51%)	**19.94 (3.33%)**	22.26 (3.72%)	27.44 (4.58%)
	WL	142.40 (23.77%)	**17.89 (2.99%)**	91.24 (15.23%)	31.62 (5.28%)

CT(Correctly Tracked), CL(Correctly Lost), WT(Wrongly Tracked), WL(Wrongly Lost)
* [14] without depth images.

Table 2: Processing time for each person image

	method	time [msec]
feature extraction	Haar & Lab	1.2
	CCF10	4.2
	CCF25	6.0
classifier update	all	0.1

ing cost, and it may lead to over-fitting. Although the feature space of CCF10 is smaller than CCF25, the "average effectiveness" of CCF10 features could be better than CCF25 since it was optimized to identify persons with fewer filters. As a result, CCF10 shows a better result than CCF25 in this case.

The result of the monocular vision-based system with CCF10 is comparable but a bit worse than the result of the laser-based system because the vision-based system failed to detect the target person when he was distant from the robot (see Fig. 10). The distance to the person was about 10 [m] in this scene. Since we used a wide angle camera in this experiment, the maximum detection range was smaller compared to the evaluation in Sec. 5.1. This result suggests that the laser-based system has the advantage of detecting and tracking persons in a long distance. However, once the robot got close to the target person, it correctly detected and re-identified him, and the tracking was resumed properly. As shown

in Sec. 5.1, the vision-based method can track persons up to 10 or 20 [m] depending on the camera characteristics, and we consider that, the distance between the target person and the robot would not get so long during a following task. Furthermore, a target person search approach like [31] could be helpful to search for the target when the robot loses the track of him/her and compensate for the drawback of the vision-based person detection.

Note that, we also tested the original Ahmed's network on this dataset, however, the results were very poor. In each sequence, we compared every person image with the target person images of the first ten seconds using the network, and classified the image into the target and others by majority-voting. However, it worked well on only easy situations (Sequence 1 and 2), and in the rest of sequences, it classified all similar persons as the target (Sequence 3, 4, and 6) or classified the target as other persons (Sequence 5). The result suggests that, even with the deep feature representation, we cannot obtain a good identification result without the online learning approach. In addition to that, it takes about 1 [sec] for each frame and is far from real-time performance.

Table 2 shows the average processing time of the feature extraction and the person classifier update on a computer with Core i7-6700K (without GPU). While the traditional feature extraction method takes 1.2 [msec] for each person image, CCF10 and CCF25 take 4.2 [msec], and 6.0 [msec], respectively. Although the CCFs are more costly than the traditional one, they are still able to run in real-time. Since the processing time of updating the person classifier depends on only the number of weak classifiers, every method takes the same time for updating (0.1 [msec] per person image).

5.3. Person identification evaluation on a public dataset

We evaluated the proposed monocular vision-based framework on a public dataset for person following robots [15]. This dataset consists of 11 sequences acquired with a stereo camera mounted on a mobile robot. At the beginning of each sequence, a person is standing in front of the robot, and the system to be evaluated learns the appearance of the person and keeps tracking him. The dataset contains hard situations for person identification (e.g., clothes and illumination changes), and the system has to deal with such situations. Since our proposed method is designed for monocular cameras, we use only the left images of the stereo image sequences to test the proposed method.

In this dataset, person identification methods are evaluated in terms of the target localization accuracy. If the distance between the center positions of the estimated and the ground truth person regions is smaller than a threshold, we judge that the system succeeded to identify the person at that frame.

We compare the proposed method with other methods reported in [15]. *OAB* [32] and *ASE* [33] are object tracking algorithms for monocular cameras, while *SOAB* [5], *DS-KCF* [34] are tracking algorithms for stereo cameras. There is also a convolutional neural network-based tracking algorithm for stereo images and its variations [15]. *CNN_v1* directly receives RGB-D images while *CNN_v2* has two streams for each of RGB and depth images and fuse them later. *CNN_v3* is a network for regular RGB images. All the networks output the similarity of an input image region to the target person.

Fig. 11 shows the evaluation result. Following the evaluation procedure in [15], we set the location error threshold to 50 pixels. The proposed method successfully keeps tracking the target persons in all the sequences, and thanks to the good accuracy of the *OpenPose* skeleton detector, the proposed method outperforms the others in this evaluation. Although the proposed method fails to detect the target person in two sequences ("*lab_and_seminar*" and "*sidewalk*" sequences) when he gets too close to the camera (see Fig. 12), once he moves away from the camera, the system correctly detects and re-identifies the target, and the tracking gets recovered. As a result, the proposed method keeps tracking the target person in the entire sequences. Fig. 13 (a) shows the plot of the localization precision versus the localization error threshold. Thanks to the good localization accuracy, the proposed method shows much higher precision under lower error thresholds. However, since it fails to track the target when he is too close to the camera, the precision under large thresholds is worse than the other state-of-the-art method (CNN_v1). Fig. 13 (b) shows the evaluation result where the two sequences shown in Fig. 12 are excluded. Under this setting, with the proposed method, the precision under smaller thresholds outperforms the others, and the result under larger thresholds is also comparable with the state-of-the-art method. It is worth mentioning that the proposed method uses only monocular images, while *SOAB, SD-KCF, CNN_v1, and CNN_v2* use stereo images.

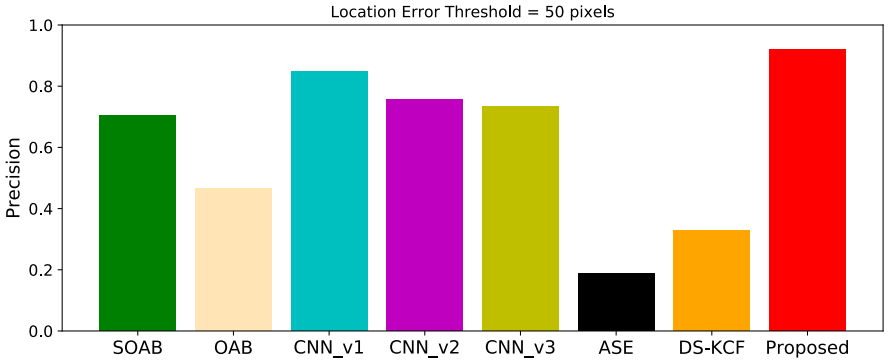
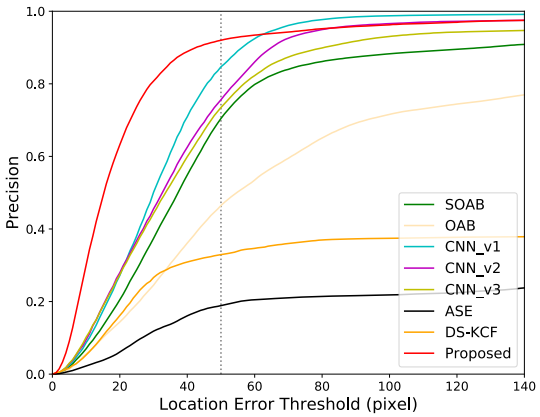


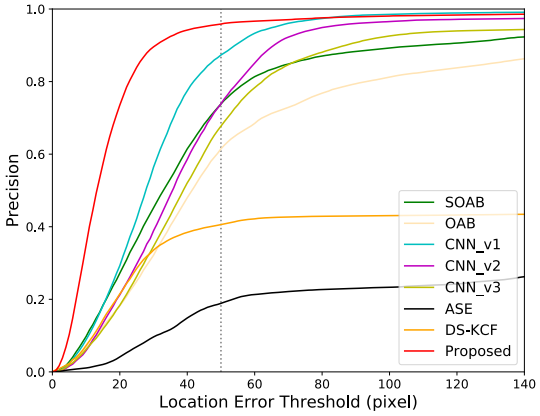
Figure 11: Target person localization evaluation result on the dataset [15].

(a) "*lab_and_seminar*" (b) "*sidewalk*"

Figure 12: Failed scenes. The target person was too close to the camera, and the system failed to detect him.



(a) With all the sequences.



(b) Without the sequences shown in Fig. 12.

Figure 13: Target localization precision vs location error threshold.

6. Person following experiment

To demonstrate that the proposed method can be applied to real robots, we conducted a person following experiment. We implemented a simple robot controller for person following; the robot moves toward the target person, and when the robot loses track of the target, it stops and waits until the person re-appears. This controller receives the target person position from the proposed person tracking and identification framework and generates velocity commands to drive the robot. We used the mobile robot shown in Fig. 7 with a Jetson TX2. All the modules including the person detection, tracking, identification, and robot controller run on the Jetson TX2, thus, we did not use any other computers in this experiment.

Fig. 14 shows snapshots of the experiment. At the beginning of the experiment, the robot learned the appearance of the target person and started following him (Fig. 14 (a)). During the experiment, the target person was occluded by the other person several times, and the robot lost the track of the target (Fig. 14 (b)(c)). However, once he re-appeared in the camera view, the robot correctly re-identified him with the online learned appearance model, and kept following him (Fig. 14 (d)). Although there was a significant illumination change when the target moved out from the room (Fig. 14 (e)(f)), the appearance model was updated adeptly, and as a result, the robot successfully followed the target person. Fig. 15 shows the features selected by online boosting during the experiment. We can see that

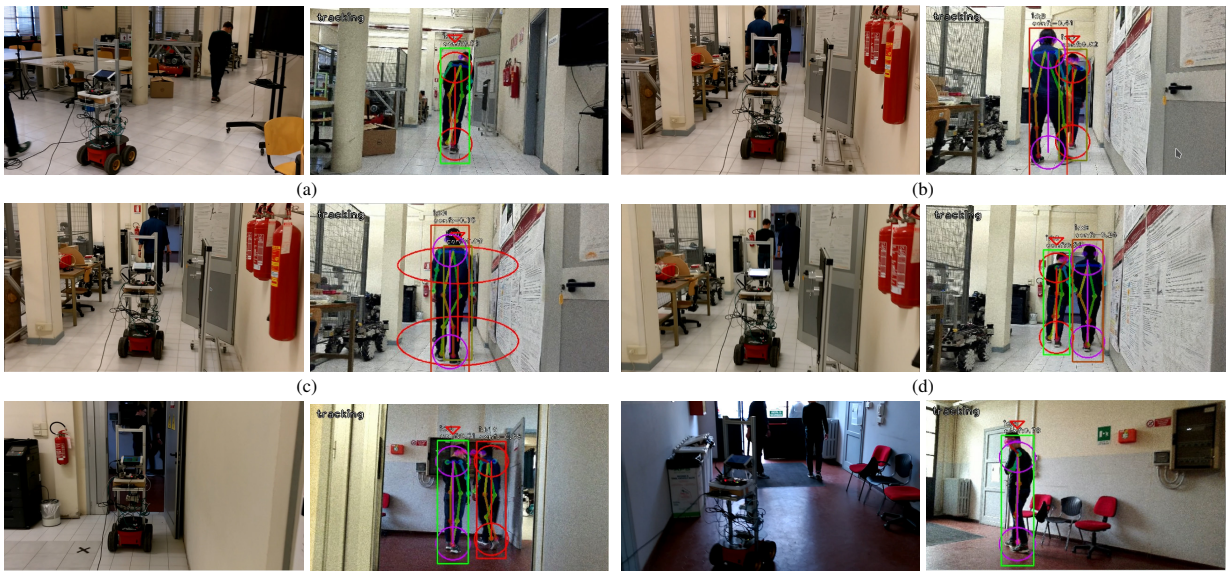


Figure 14: The person following experiment. The left images are the snapshots of the experiment, and the right images are the tracking identification results. The red triangles in the right images indicate the person identified as the target.

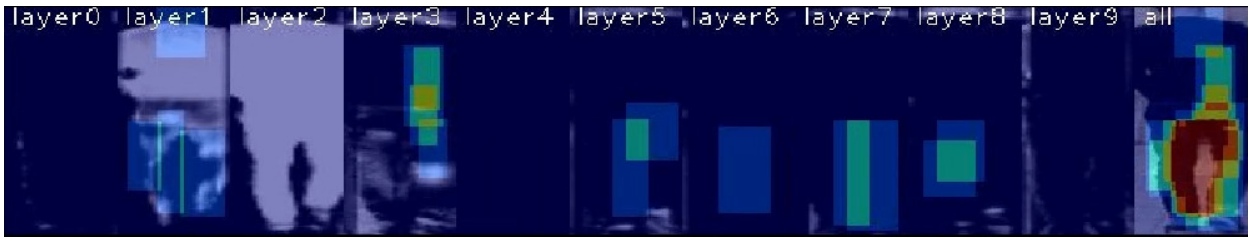


Figure 15: Features selected by online boosting during the person following experiment.

Figure 16: The outdoor person following experiment. The robot successfully kept tracking and following the target person even the target was occluded by the other one several times.

the classifier focused on the trousers region to robustly identify the target in this case.

We also run the proposed system in an outdoor environment. Fig. 16 shows snapshots of the trial. The robot successfully kept tracking and following the target person even the target was occluded by the other one several times. Although the ground is rough compared to the indoor environment, it did not affect the position estimation so much, and the robot kept a certain distance while following the person during this trial.

7. Conclusion

This paper has proposed a monocular camera-based person tracking and identification framework with online selection of deep features for person following

robots. The proposed framework first detects persons using *OpenPose* and then track them in the robot space using Unscented Kalman Filter with the ground plane prior information and human height estimation. A person identification based on the combination of Convolutional Channel Features and online boosting runs on the top of the tracking module to keep tracking the target person robustly. Through evaluations, it has been shown that the proposed framework outperforms other state-of-the-art methods. The experiment demonstrated that the proposed method can be applied to a real mobile robot.

As a future work, we are planning to improve the detection rate of close persons by replacing the camera with one with a wide view angle, and incorporate the proposed framework with an active person search strategy to re-identify a distant target person.

References

[1] K. O. Arras, O. M. Mozos, W. Burgard, Using boosted features for the detection of people in 2D range data, in: IEEE International Conference on Robotics and Automation, IEEE, 2007, pp. 3402–3407 (2007). doi:10.1109/ROBOT.2007.363998.

[2] T. Linder, S. Breuers, B. Leibe, K. O. Arras, On multi-modal people tracking from mobile platforms in very crowded and dynamic environments, in: 2016 IEEE International Conference on Robotics and Automation, IEEE, 2016 (may 2016). doi:10.1109/icra.2016.7487766.

[3] K. Koide, J. Miura, Identification of a specific person using color, height, and gait features for a person following robot, Robotics and Autonomous Systems 84 (2016) 76–87 (2016). doi:10.1016/j.robot.2016.07.004.

[4] J. Satake, M. Chiba, J. Miura, A SIFT-based person identification using a distance-dependent appearance model for a person following robot, in: IEEE International Conference on Robotics and Biomimetics, IEEE, 2012, pp. 962–967 (2012). doi:10.1109/robio.2012.6491093.

[5] B. X. Chen, R. Sahdev, J. K. Tsotsos, Person following robot using selected online ada-boosting with stereo camera, in: Conference on Computer and Robot Vision, 2017, pp. 48–55 (2017).

[6] M. Munaro, E. Menegatti, Fast RGB-d people tracking for service robots, Autonomous Robots 37 (3) (2014) 227–242 (2014). doi:10.1007/s10514-014-9385-0.

[7] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, Y. Sheikh, OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields, in: arXiv preprint arXiv:1812.08008, 2018 (2018).

[8] W. Choi, S. Savarese, Multiple target tracking in world coordinate with single, minimally calibrated camera, in: European Conference on Computer Vision, Springer Berlin Heidelberg, 2010, pp. 553–567 (2010). doi:10.1109/ICCV.2015.18.

[9] I. Ardiyanto, J. Miura, Partial least squares-based human upper body orientation estimation with combined detection and tracking, Image and Vision Computing 32 (11) (2014) 904–915 (nov 2014). doi:10.1016/j.imavis.2014.08.002.

[10] B. Yang, J. Yan, Z. Lei, S. Z. Li, Convolutional channel features, in: IEEE International Conference on Computer Vision, IEEE, IEEE, 2015 (dec 2015). doi:10.1109/ICCV.2015.18.

[11] H. Grabner, H. Bischof, On-line boosting and vision, in: IEEE Conference on Computer Vision and Pattern Recognition, Vol. 1, IEEE, 2006, pp. 260–267 (2006). doi:10.1109/cvpr.2006.215.

[12] K. Koide, J. Miura, Convolutional channel features-based person identification for person following robots, in: Intelligent Autonomous Systems 15, Springer International Publishing, 2018, pp. 186–198 (dec 2018). doi:10.1007/978-3-030-01370-7_15.

[13] A. Leigh, J. Pineau, N. Olmedo, H. Zhang, Person tracking and following with 2d laser scanners, in: IEEE International Conference on Robotics and Automation, 2015, pp. 726–733 (2015). doi:10.1109/ICRA.2015.7139259.

[14] M. Luber, L. Spinello, K. O. Arras, People tracking in RGB-d data with on-line boosted target models, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2011, pp. 3844–3849 (2011). doi:10.1109/iros.2011.6095075.

[15] B. X. Chen, R. Sahdev, J. K. Tsotsos, Integrating stereo vision with a CNN tracker for a person-following robot, in: Lecture Notes in Computer Science, Springer International Publishing, 2017, pp. 300–313 (2017).

[16] M. Zhang, X. Liu, D. Xu, Z. Cao, J. Yu, Vision-based target-following guider for mobile robot, IEEE Transactions on Industrial Electronics (2019) 1–1 (2019). doi:10.1109/TIE.2019.2893829.

[17] M. N. A. Bakar, A. R. M. Saad, A monocular vision-based specific person detection system for mobile robot applications, Procedia Engineering 41 (2012) 22–31 (2012). doi:10.1016/j.proeng.2012.07.138.

[18] N. Yao, E. Anaya, Q. Tao, S. Cho, H. Zheng, F. Zhang, Monocular vision-based human following on miniature robotic blimp, in: 2017 IEEE International Conference on Robotics and Automation, IEEE, 2017 (may 2017). doi:10.1109/icra.2017.7989369.

[19] Y. Makihara, H. Mannami, Y. Yagi, Gait analysis of gender and age using a large-scale multi-view gait database, in: Asian Conference on Computer Vision, Springer, 2011, pp. 440–451 (2011). doi:10.1007/978-3-642-19309-5_34.

[20] G. Berdugo, O. Soceanu, Y. Moshe, D. Rudoy, I. Dvir, Object reidentification in real world scenarios across multiple non-overlapping cameras, in: European Signal Processing Conference, 2010, pp. 1806–1810 (2010).

[21] M. Munaro, S. Ghidoni, D. T. Dizmen, E. Menegatti, A feature-based approach to people re-identification using skeleton keypoints, in: IEEE International Conference on Robotics and Automation, IEEE, 2014, pp. 5644–5651 (2014). doi:10.1109/icra.2014.6907689.

[22] V. Alvarez-Santos, X. M. Pardo, R. Iglesias, A. Canedo-Rodriguez, C. V. Regueiro, Feature analysis for human recognition and discrimination: Application to a person-following behaviour in a mobile robot, Robotics and autonomous systems 60 (8) (2012) 1021–1036 (2012).

[23] E. Ahmed, M. Jones, T. K. Marks, An improved deep learning architecture for person re-identification, in: IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2015, pp. 3908–3916 (2015). doi:10.1109/cvpr.2015.7299016.

[24] A. Schumann, R. Stiefelhagen, Person re-identification by deep learning attribute-complementary information, in: IEEE Conference on Computer Vision and Pattern Recognition Workshops, IEEE, 2017 (2017). doi:10.1109/CVPRW.2017.186.

[25] D. Sahoo, Q. Pham, J. Lu, S. C. H. Hoi, Online deep learning: Learning deep neural networks on the fly, in: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, Vol. abs/1711.03705, International Joint Conferences on Artificial Intelligence Organization, 2017 (jul 2017). arXiv:1711.03705, doi:10.24963/ijcai.2018/369.

[26] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam, Mobilenets: Efficient con-

volutional neural networks for mobile vision applications, arXiv preprint arXiv:1704.04861 (2017).

[27] E. Wan, R. V. D. Merwe, The unscented Kalman filter for nonlinear estimation, in: Adaptive Systems for Signal Processing, Communications, and Control Symposium, IEEE, 2000 (2000). doi:10.1109/asspcc.2000.882463.

[28] Z. Radosavljevic, A study of a target tracking method using global nearest neighbor algorithm, Vojnotehnicki glasnik (2) (2006) 160–167 (2006). doi:10.5937/vojtehg0602160r.

[29] W. Li, R. Zhao, X. Wang, Human reidentification with transferred metric learning, in: Asian Conference on Computer Vision, 2012 (2012).

[30] W. Li, R. Zhao, T. Xiao, X. Wang, Deepreid: Deep filter pairing neural network for person re-identification, in: IEEE Conference on Computer Vision and Pattern Recognition, 2014 (2014).

[31] C. Granata, P. Bidaud, A framework for the design of person following behaviors for social mobile robots, in: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2012 (oct 2012). doi:10.1109/iros.2012.6385976.

[32] H. Grabner, M. Grabner, H. Bischof, Real-time tracking via online boosting, in: Procedings of the British Machine Vision Conference 2006, British Machine Vision Association, 2006 (2006). doi:10.5244/c.20.6.

[33] M. Danelljan, G. Häger, F. S. Khan, M. Felsberg, Accurate scale estimation for robust visual tracking, in: Proceedings of the British Machine Vision Conference 2014, British Machine Vision Association, 2014 (2014). doi:10.5244/c.28.65.

[34] M. Camplani, S. Hannuna, M. Mirmehdi, D. Damen, A. Paiement, L. Tao, T. Burghardt, Real-time RGB-d tracking with depth scaling kernelised correlation filters and occlusion handling, in: Procedings of the British Machine Vision Conference 2015, British Machine Vision Association, 2015 (2015). doi:10.5244/C.29.145.