Person Identification and Person's Awareness Estimation for an Attendant Robot
(付き添いロボットのための人物識別とアウェアネス推定)

March, 2019

Doctor of Philosophy (Engineering)

Kenji Koide
小出健司

Toyohashi University of Technology

別紙４－１（課程博士（英文））

Date of Submission（month day, year）: March 15th , 2019

Department of Computer Science and Information Engineering	Student ID Number	D113413
Applicant's name	Kenji Koide	

Supervisors	Jun Miura Michio Okada

Abstract（Doctor）

Title of Thesis	Person Identification and Person's Awareness Estimation for an Attendant Robot

Approx. 800 words

Developed countries including Japan are becoming aged societies, and they suffer from a chronic shortage of caregivers. In this decade, robotic caregiving has been attracting people's attention as a solution for the problem and they are expected to be deployed in the next decade.

In this thesis, we investigate a robotic attendant system for caregiving. In this system, a robot follows a person and keeps him/her away from dangerous situations. In case the person is going to be involved in an accident (e.g., bumping into an obstacle and falling from a step), the robot prevents the accident by interacting with him/her (e.g., informing the person of the obstacle). However, if the robot interacts with the person every time it finds an accident risk, it could be annoying for the person. To be socially accepted, the robotic attendant system has to avoid disturbing the person as long as he/she is in a safe situation so that it becomes comfortable for the person. To minimize the risk of accidents while maximizing the comfortableness of attendance, we focus on a person's awareness. By estimating the person's awareness of obstacles, the robot can assess the collision risk and interact with the person only when an accident is likely to happen.

First, we present robust person tracking and identification methods. Attendant robots have to be able to follow a specific target person reliably. In case there are several persons, the robot may lose track of the target person due to occlusion. In such cases, it is required to identify the target person among surrounding persons (i.e., re-identification) to resume the tracking and continue to follow the target. We propose online learning-based person identification methods based on deep convolutional neural network-based appearance features and illumination invariant height and gait features. They can identify the target person robustly in severe illumination environments. We also propose a wearable device-based identification method. In this method, we let the target person hold a smartphone and identify him/her by matching the foot strike timings detected by the smartphone and the ones of surrounding persons detected by a laser range finder mounted on the robot. By combining the online learning-based and wearable device-based identification methods, we realize a robust and reliable person following system.

Second, we propose a system to measure and analyze real persons' attending behavior. In this system, an observer carrying a 3D LIDAR follows persons to be measured while keeping them in the sensor view. It allows us to measure the behavior of the persons without area and time limitations. The system first constructs a 3D environmental map beforehand and then estimates the sensor pose and tracks surrounding people online. As a field test, we measured the behavior of professional caregivers attending an elderly with dementia in a hospital. A preliminary analysis of the attendant behavior reveals how they decide the positioning with respect to the elderly while paying attention to the surrounding environment to prevent accidents.

Third, we propose methods to estimate a person's awareness of the surrounding environment. In the case of person-following robots, they cannot observe features which directly reflect the person's awareness (e.g., head orientation and gaze). We, thus, propose methods to estimate a person's awareness solely from the person's trajectory. As a proof-of-concept, we propose a model to estimate a person's awareness of an obstacle in a corridor. Then, we extend the model so that it can handle arbitrary obstacles and environmental structures with a deep convolutional network.

Finally, we present a proposal for the system design of an attendant robot. The robot reliably follows a specific target person with the proposed person identification methods, and its behavior is designed based on the assessment of accident risks with awareness estimation for safe and comfortable attendance. It would be a step towards a socially acceptable robotic caregiving system.

Contents

Chapter 1

Introduction

1.1 Attendant Robots for Caregiving

Developed countries including Japan are becoming aged societies, and they suffer from a chronic shortage of caregivers [1]. One of the important problems in such countries is the increasing caregiving resource, and robotic caregiving has been attracting people's attention as a solution to the problem [2]. Several kinds of robotic caregiving systems have been deployed in hospitals in the last decade.

One way for the robotic caregiving is physical support of elderly persons and human caregivers. RT. WORKS developed a robotic walker RT. 2 which assists an elderly person's walking and prevent them from falling and stumbling with semi-automatic brake control [3]. When the elderly gets on a slope or stumbles into an obstacle, it applies a brake automatically to avoid the elderly from falling. Cyberdyne has developed a robotic suit to support human caregivers' tasks, such as moving an elderly from a bed to a wheelchair and giving rehabilitation to the elderly [4]. It enhances the caregiver's power and reduces the effort for those tasks. Those systems provide physical support for caregivers and elderly persons, and they are practical and promised ways. However, even though such systems reduce the effort of caregivers, they require caregivers' monitoring or operation. That is, they essentially rely on human caregivers. Without largely increasing caregivers, they cannot deal with the growing number of elderly persons. For this reason, we believe that autonomous robotic systems, which do not require caregivers' monitoring and operation, are necessary to deal with the problem.

Paro and Palro, interaction robots respectively developed by AIST and Fuji software, have been deployed in several hospitals to communicate with elderly persons

(A) Paro [5].

(B) Palro [6].

FIGURE 1.1: Interaction robots used for elderly care.

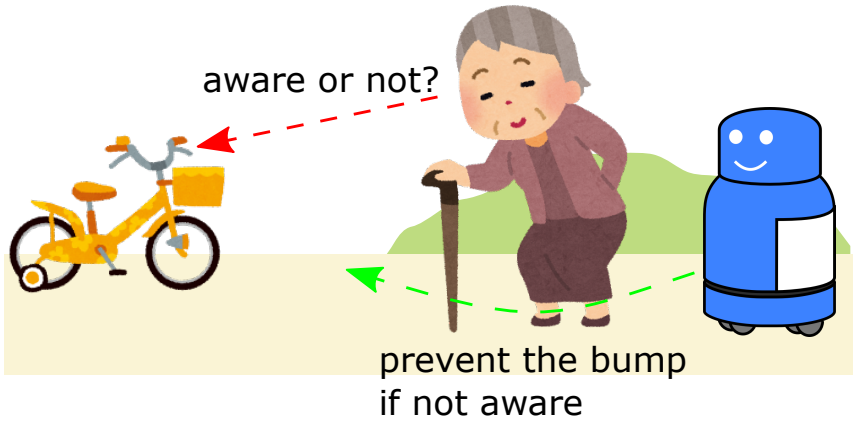


FIGURE 1.2: Robotic attendant based on awareness estimation.

with dementia [5, 6] (see Fig. 1.1). Through communication with them, elderly persons suppress loneliness and could slow down the progression of dementia. Since they do not need caregiver's monitoring, they can be scalable to the growing aged population. However, their applications are limited to the basic communication services in indoors.

One of the most common demands from elderly persons under caregiving is walking outdoors. However, for safety reasons, they are not allowed to freely go out by themselves sometimes. In order to avoid the risk of accidents, such as falling from a step and hitting by a car, they must be attended by a caregiver when they go out. Even in indoor scenes, caregivers' monitoring is required depending on the level of dementia. However, the limitation of the numbers of caregivers makes it hard to allow elderly persons to have enough time for walking. We believe that walking freely outside as well as inside is essential to increase the quality of life, and attendant robots would be one possible way to deal with this problem.

1.2 Research Goal

Our goal is to develop an attendant robot, which keeps elderly persons away from dangerous situations while avoiding disturbing them. To achieve this, we focus on person's awareness. We consider that, by estimating a person's awareness, we can assess the risk of an accident (see Fig. 1.2). If a person is not aware of an obstacle, there is a high risk that the person bumps into it. On the other hand, if she is aware of the obstacle, she would avoid the obstacle by herself, and the risk is low. In the former case, the robot should take an action, for instance, informing her of the obstacle, to prevent the accident while it should not do it to avoid disturbing the person in the latter case. We believe that such an attendant robot, which interacts with persons only when it is necessary, is suitable for not only elderly caregiving, but also other applications, such as observing children and watching people in home and public environments.

In this thesis, we propose a robotic attendant framework which consists of 1) a person following capability based on robust person identification, 2) methods for people behavior measurement and analysis, and 3) an awareness estimation model for mobile robots.

To follow a person, robots have to reliably identify the person to be followed. We propose several methods to identify a specific person using sensors mounted

on the robot and a marker held by the person. We also propose a system for long-term and wide-area people behavior measurement. With this system, we measured professional caregivers' behavior during attendance to elderly persons. The measured behavior data is useful to design socially acceptable robot behavior. Last but not least, we construct a model to estimate a person's awareness of the surrounding environment from his/her motion. It allows robots to assess the risk of accidents during attendance.

1.3 Related Work

1.3.1 Awareness Estimation

Driver's awareness is an important feature in driver assistance. If a driver is not aware of a pedestrian, there is a risk of severe accidents, and in that case, we can prevent the accident by informing the driver of the pedestrian. Phan et al. [7] estimated a car driver's awareness of a pedestrian from driving signals, such as acceleration pedal position, brake force, steering wheel angle, and vehicle speed. They trained an HMM (Hidden Marcov Model) from the driving signals to estimate a driver's awareness of pedestrians. Bar et al. [8] trained a decision tree which assesses a driver's awareness of traffic objects, such as pedestrians, other cars, and traffic cones. They observed the driver's gaze and combined it with traffic objects information detected by cameras and laser scanners to construct the decision tree. Chutorian et al. [9] used a driver's head pose instead of gaze information since detecting gaze is sometimes impractical in real driving situations. They proposed a method to estimate a driver's head pose from an image, and monitored the driver's awareness by using the estimated head pose.

Awareness estimation has also been researched in human-system interaction. Stiefelhagen et al. [10] estimated the point, where a person is looking, from his/her head pose to enhance the usability of computer interaction systems. Doshi et al. [11] also estimated persons' attention using head pose estimation to find distractive objects in a meeting room. These works used persons' awareness to monitor persons' state and tried to realize comfortable human-system interaction.

Although awareness estimation plays an important role in these researches, the existing works are based on features which directly reflect a person's awareness, such as gaze, head pose, and user's operation. However, in the case of person following robots, it is difficult to observe these features from a robot, because it cannot see the person's face from the behind of target persons, nor it receives no explicit operation from the persons. We need to estimate a person's awareness from features which can be observed from a person following robot, such as person's position and velocity.

1.3.2 Social Interaction Robots

Human-robot social interaction is becoming an important research field in the robotics community. There is an increasing demand for robotic services, such as receptionist and waiter, and these tasks require sophisticated interaction abilities of robots. Service robots have to interact with people in "social" manners to realize natural and comfortable services and be accepted by people [12]. Some works proposed social interaction-based robot systems. Dewantara et al. [13] proposed a navigation robot framework based on an extended social force model, and they train the

model parameters from the person's position, velocity, and head orientation using reinforcement learning.

These works proposed robotic systems under consideration of social interaction with people. While these methods contribute to natural communication with people, they consider only the robot-human relationship. In attending tasks, the surrounding environment changes as a person and a robot move, and we need to take the person-environment interaction into account.

1.3.3 Person Following Robots

Person following tasks require several fundamental functions, such as person tracking and identification[14, 15], environment recognition [16], and path planning [17]. A lot of works have been done for such functions to realize reliable person following robots. As a result, person following robots have reached at a practical level, and some of them have already been in sales [18]. Some works proposed social interaction-based person following robots, which do not just keep the distance to the person constant but adjust the distance, for comfortable person following services [19, 20]. In these works, depending on the positions of the target person and surrounding people, the distance between the robot and the target person is decided based on social force model. Oishi et al. [21] introduced a state machine to adapt a robot's attending position depending on the target person's behavior. These works show that, by deciding attending position under consideration of social interaction, these methods achieve a certain degree of comfortableness in attending services. However, as described in Sec. 1.1, attendant robots have to not only follow the target persons but also keep them away from dangerous situations. To our knowledge, none of the existing works considered such a role, and we need to realize robots which provide not only comfortable but also safety attendant services.

1.4 Contributions

The main contribution of this work is the introduction of a novel robotic attendant system based on real human behavior analysis and awareness estimation. The framework is built on the top of reliable person identification [22, 23, 24]. The development of the identification methods is also a contribution of the thesis. The core of this framework is the awareness estimation model. Different from existing methods, our method depends only on a person's trajectory, which can easily be obtained from a mobile robot, so that it can be applied to real attendant services. In addition to that, our model can represent complex person-to-person and person-to-environment relationships, which are hard to model with traditional social models [25, 26], thanks to the use of a deep convolutional neural network-based approach. Furthermore, we propose a wide-area and long-term people behavior measurement system using a 3D LIDAR. Based on an analysis of professional caregivers' behavior measured by this system, we also present a design of basic person following behavior for attendant robots.

1.5 Thesis Organization

This thesis is organized as follows: We first describe the proposed people tracking and identification methods, which are fundamental for reliable person following, in Chapter 2 and 3, respectively. Chapter 4 explains a people behavior measurement

system which is used to measure and analyze human behavior. The analysis result of real professional caregivers' behavior during attendance is also described in this chapter. The awareness estimation model is proposed in Chapter 5. Chapter 6 concludes the thesis and discusses an attendant robot system design.

Chapter 2

People Tracking

2.1 Related Work on People Tracking

People tracking is a fundamental function for person following robots. To follow a person, the robots have to be able to keep tracking the identity of the target person. One of the most standard and established scheme for Multiple Object Tracking (MOT) is so-called "tracking-by-detection" [27]. In this scheme, the system detects all the objects to be tracked in the sensor view using a detector, and then associates the detections over time to track the identity of each object.

As the starting point of the tracking process, the detector has a crucial impact on the tracking result. To improve the detection accuracy, various sensors, such as RGB, depth, stereo cameras [28, 15, 14], 2D, 3D LIDARs [29, 30], and their combinations [31, 32] have been investigated.

The tracking part, which follows the detection part, also plays an important role in the tracking system. For robust tracking, state estimation filters have been employed to predict people motion [33, 34] and combined with several data association methods [35, 36, 37].

2.1.1 Vision-based People Detection

Vision-based human detection has been widely studied in the computer vision community over than two decades. Early works proposed simple template matching-based methods to detect faces in images [38, 39]. To deal with facial expression and illumination changes, the combination of machine learning-based detector and the sliding window approach were proposed lately. One of the most successful method for face detection is the one proposed by Viola and Jones [40]. In this method, they trained a cascaded AdaBoost classifier on Haar-like features. The cascaded classifier quickly rejects background regions while spending more computation on object-like regions. This approach allows us to save the processing cost while keeping the detection accuracy. Since the human face is well-structured, this kind of sliding window approaches work well, and the naive face detection problem is considered to be solved nowadays. More recent works, thus, focus on further facial analysis problems, such as facial landmark detection [41] and expression recognition [42].

Since the human body has more variation on the appearance and the shape compared to the face, vision-based human detection is still considered to be challenging [43]. Classic people detection methods train cascaded classifiers on sophisticated appearance features to detect people. To model human body shape, HOG (Histogram of Oriented Gradients) [44], ICF (Integral Channel Features) [45], and their variations (*Checkerboards* [46] and *RotatedFilters* [43]) are often exploited, and by using a soft cascade classifier [47], which reuses the confidence of the rejection decision of the previous cascade to improve the false negative rate, we can detect people in

images efficiently and robustly. More recently, deep convolutional neural network-based people detection methods were proposed. Some of them take the form of the traditional sliding window approach [48, 49, 50] while the other ones take different bottom-up approaches based on fully convolutional networks [51]. The most successful ones are based on detection of body landmarks (e.g., head, chest, arms, and legs) [52, 53]. In contrast to the sliding window approach which trains a binary classifier, they train a network which outputs a set of response maps corresponding to each of body landmarks and detect people by aggregating the body landmark detection results. They can naturally deal with partial occlusion of body parts, and thus have the advantage of detection of people in crowded scenes. One notable work based on this approach is proposed by Cao et al. [54]. It estimates *Part Affinity Fields* (PAFs), a set of 2D vector fields that encode the location and orientation of limbs, and a following parsing step associates detected body part candidates on PAFs by performing bipartite matching. It is also known as *OpenPose* architecture, and its open source implementation has been widely used in many applications. Another interesting idea is to introduce the end-to-end learning fashion to the people detection problem. Stewart et al. [55] put a recurrent LSTM (Long Short-Term Memory) layer after a convolution network. The LSTM acts as a controller which decodes the feature maps encoded by the CNN and outputs a sequence of detection bounding boxes.

Although the great progress has made on visual people detection, it is still hard and costly to detect people in images. In the last decade, affordable consumer RGB-D and stereo cameras became available, and they have been widely applied to the people detection purpose [28, 15]. Typically, human candidate objects are detected using the Euclidean clustering technique, and then non-human objects are eliminated by a machine learning-based classifier. Although they show a better false positive rate than RGB cameras, the detection range is often limited (e.g., ~ 5[m]).

2.1.2 LIDAR-based People Detection

LIDARs have been widely used for real-time people detection [29, 56, 30]. They provide the distance to objects precisely, and the range information allows us to easily separate foreground objects from the background using simple clustering algorithms. Compared to vision sensors, they have the advantage of long-range and wide-area object detection and outperform vision sensors in terms of the detection accuracy and speed. In addition to that, they are useful for other tasks required for mobile robots (e.g., mapping and localization), and thus, LIDARs have been a "de facto" sensor for mobile robots.

2.1.3 People Tracking

In the "tracking-by-detection" scheme, the detection part is followed by a tracking part which typically consists of two steps: *filtering* and *data association*.

The *filtering* step estimates the current person state (e.g., position and velocity) from observations until the current frame and predicts the person state in the next frames. Kalman filter [33] has been often used for this purpose. It is a kind of recursive Bayesian filters, and it allows us to estimate a state, which may contain latent variables like velocity, by recursive prediction and correction steps. The constant velocity model is commonly exploited to predict people motion. Although this model is very naive, it works well in the short-term. Several works proposed more sophisticated non-linear motion models based on Social Force Model (SFM) and collision

avoidance constraints, and combined them with non-linear filters, such as extended Kalman filter [34].

The data association step associates tracks and observations at a frame based on the distances between them. The Nearest Neighbor (NN) association is the simplest method which associates observations with the closest tracks. An extension of the NN association, Global Nearest Neighbor association, finds the association which minimizes the sum of the distances between the track-observation pairs using a combinatorial optimization algorithm, like Hungarian algorithm [57]. While these nearest neighbor methods consider only hard (one-by-one) associations, further methods, such as JPDAF (Joint Probabilistic Data Association Filter) [36] and MHT (Multiple Hypothesis Tracking) [37], consider multiple association hypotheses for each track. JPADF considers all possible combinations of track-observation associations and update the target states based on the joint data association probabilities. MHT keeps association hypothesis over frames, and it allows us to postpone to make data association decisions until data association ambiguities are resolved. Since they consider all possible assignments of measurements, they suffer from combinatorial complexity. Thus, some techniques to reduce the computational complexity, such as gating and hypothesis pruning are required.

Although the advanced data association methods with sophisticated motion models may improve the tracking accuracy, the choice of data association method matters less. For practical applications like service robots with limited computational resources, well-tuned simple data association methods can be a better choice than the complex tracking methods [58].

2.2 Proposed System

We propose two people tracking methods. One relies solely on a monocular camera (Sec. 2.2.1) while the other one combines a 2D LRF and a monocular camera (Sec. 2.2.2). We can choose either of them depending on the equipment on the robot and the use case. Both the methods take the form of the standard people tracking scheme based on Kalman filter with the constant velocity model and the global nearest neighbor data association. Although there are advanced tracking methods in the literature, it is unavoidable that tracking methods suffer from occlusion of people. Thus, we decided to keep the tracking methods relatively simple and build a re-identification mechanism on the top of the tracking module to achieve a reliable person following capability.

2.2.1 Monocular Vision-based People Tracking

Here, we propose a people tracking method which relies solely on a monocular camera. As the starting point of the people tracking process, we use *OpenPose*, a deep convolutional neural network-based human detector [59]. 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 [60] with depth-wise separable convolution filters [1]. Then, inspired by [61] and [62], 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 where the persons are actually moving [61]. In addition to that, person positions in the robot space are very useful for service robots

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

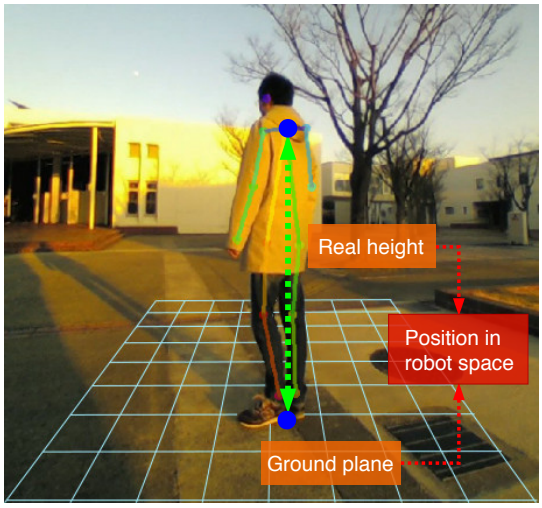


FIGURE 2.1: The proposed tracking method takes advantage of the ground plane information.

to interact with them. For instance, with the estimated position in the robot space, we can easily control the robot so that it keeps the distance to the person constant while avoiding other persons in a following task.

Fig. 2.1 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) [63] 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, if the real height is available, we can update the UKF with only a neck detection when the ankle is not visible to the camera.

State Estimation

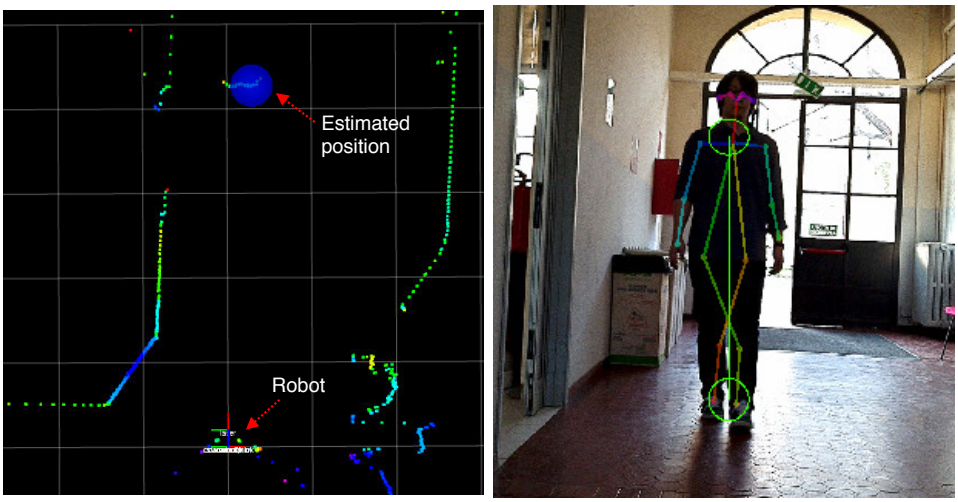
We define the state space to be estimated as $\boldsymbol{x}_t = [\boldsymbol{p}_t, \boldsymbol{v}_t, h_t]^T$, which consists of the position, velocity, and height of a person in the robot space. With UKF, we estimate the state from observations of neck and ankle positions in the image space $\hat{\boldsymbol{z}}_t = [\boldsymbol{p}_t^{neck}, \boldsymbol{p}_t^{ankle}]^T$.

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

$$f(\boldsymbol{x}_t) = \boldsymbol{x}_{t+1} = [\boldsymbol{p}_t + \Delta t \cdot \boldsymbol{v}_t, \boldsymbol{v}_t, h_t]^T, \tag{2.1}$$

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

$$h(\boldsymbol{x}_t) = \boldsymbol{z}_t = [Proj(\boldsymbol{p}_t + [0, 0, h_t]^T), Proj(\boldsymbol{p}_t)]^T, \tag{2.2}$$



(A) In the robot space. (B) In the image space.

FIGURE 2.2: 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.

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:

$$h'(\boldsymbol{x}_t) = \boldsymbol{z}'_t = [Proj(\boldsymbol{p}_t + [0, 0, h_t]^T)]^T. \tag{2.3}$$

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 the Unscented Transform [63]:

$$\mu_t^z, \sigma_t^z = UT(\mu_t^{x_t}, \sigma_t^{x_t}, h). \tag{2.4}$$

μ_t^z and σ_t^z are the expected observation distribution, $\mu_t^{x_t}$ and $\sigma_t^{x_t}$ 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:

$$Dist(track_i, obs_j) = \begin{cases} \infty, & \text{if } D_M(\mu_t^z, \sigma_t^z, \hat{\boldsymbol{z}}_t) > th_{gate} \\ -\mathcal{N}(\mu_t^z, \sigma_t^z, \hat{\boldsymbol{z}}_t), & \text{otherwise} \end{cases}, \tag{2.5}$$

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 [35]. Note that in the data association algorithm, a constant is added to the calculated distances to make them positive.

Fig. 2.2 shows a tracking result. The blue sphere in Fig. 2.2 (a) shows the estimated person position in the robot space, and the green ellipses in Fig. 2.2 (b) indicate the neck and ankle positions distribution calculated from the person state in the robot space.

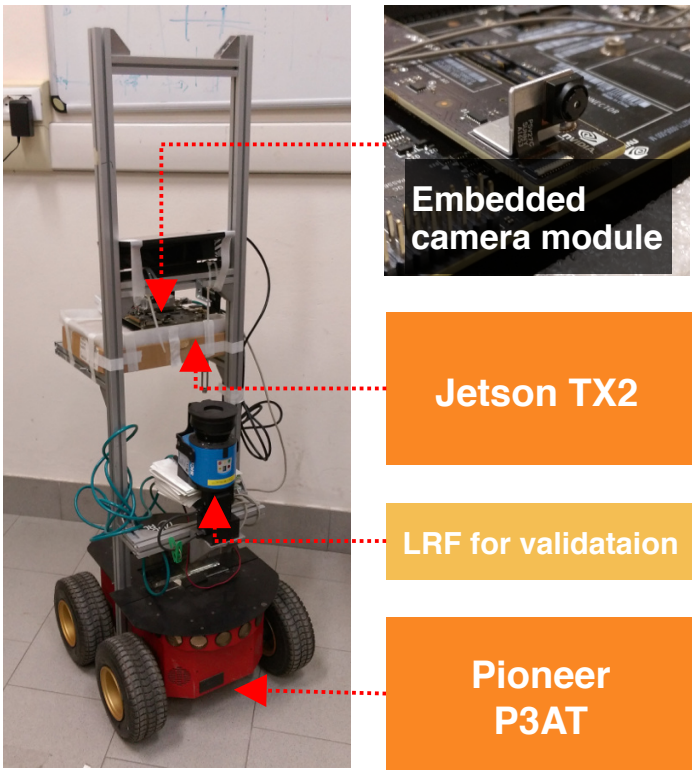


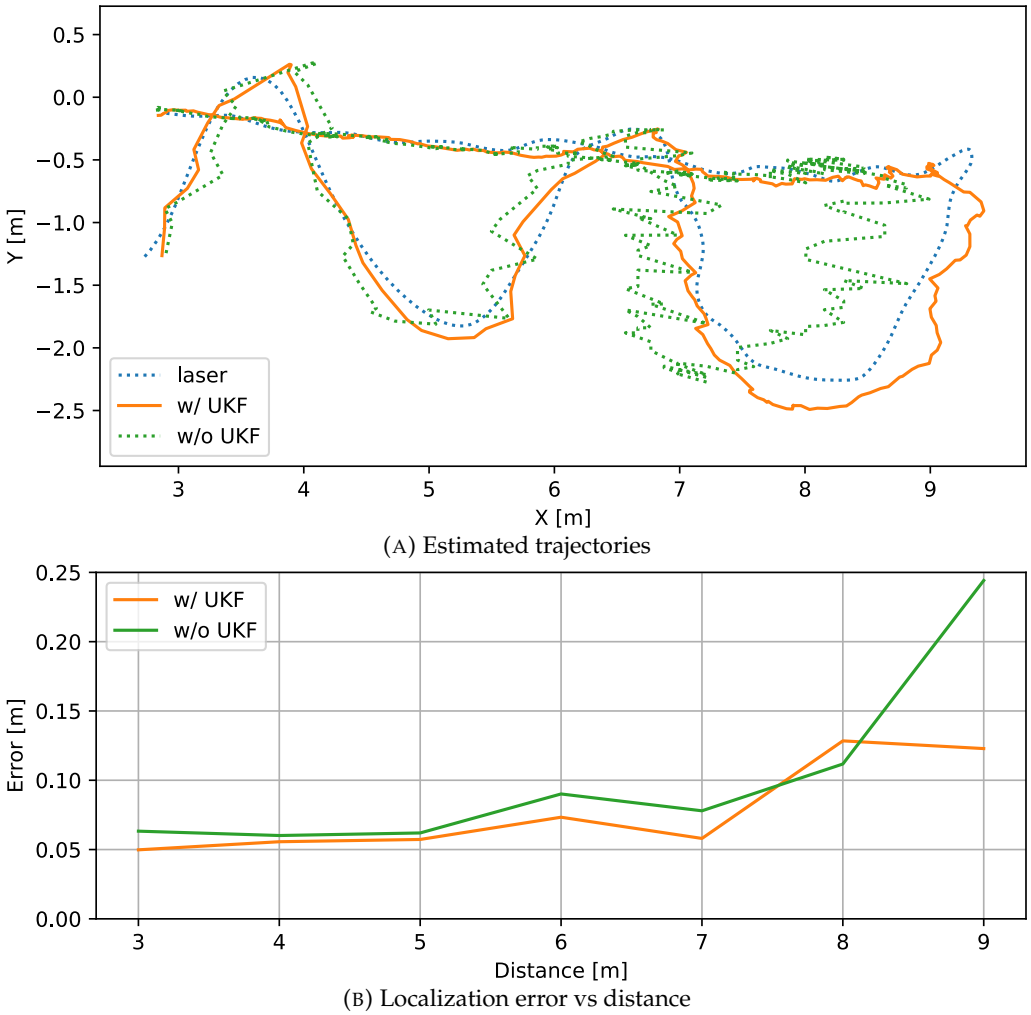
FIGURE 2.3: A mobile robot equipped with a Nvidia Jetson TX2 development board. An embedded camera module is bundled with the development board. An LRF is also mounted on the robot for the validation of the tracking system.

Evaluation

To evaluate the accuracy of the proposed tracking method, we recorded an image sequence with the robot shown in Fig. 2.3. A Jetson TX2 development board with an embedded monocular camera module is mounted on the robot. The camera pose with respect to the ground plane is calibrated by observing a chessboard pattern put on the ground. For evaluation, 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 [23]. We consider the laser-based result as the ground truth in this evaluation.

Fig. 2.4 (a) shows the trajectories estimated by the proposed and the laser-based methods. 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 the trajectory estimated by the proposed method well matches with the one estimated by the laser-based one. On the other hand, without the UKF, the error gets larger when the person is distant from the camera (around $7 \sim 9$ [m]) due to the calibration error on the camera pose with respect to the ground. With the proposed UKF-based method, the estimation in the distant place gets improved thanks to the height information which allows us to estimate the distance to the person without the ground plane.

Fig. 2.4 (b) shows the plot of the localization error versus the distance between the camera and the person. We can see that, with the UKF, the estimation in the distant place is significantly improved, and the error is smaller than 0.15 [m] in the range between $3 \sim 9$ [m]. We consider that this accuracy is enough to perform the basic following behavior of person following robots (e.g., keeping the distance to the



(A) Estimated trajectories

(B) Localization error vs distance

FIGURE 2.4: The tracking accuracy evaluation result.

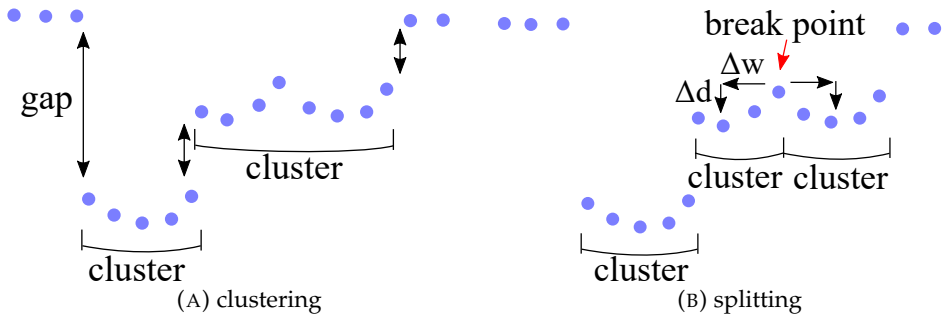
person constant).

2.2.2 LRF-based People Tracking

People Detection

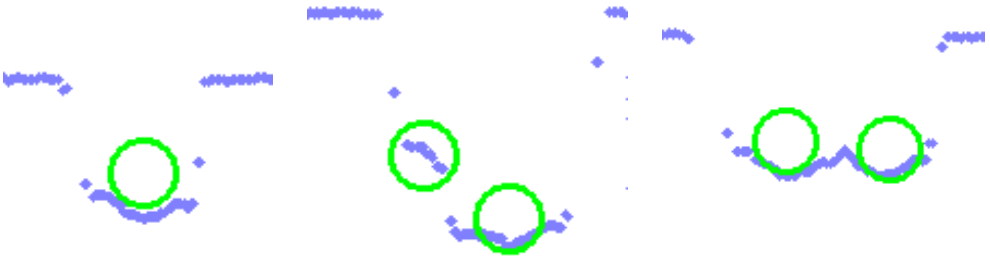
Multiple layered LRFs are sometimes used for human detection [64]. Typically these sensors are put at the height of torsos and legs, and then both detection results are combined. They assume that the torso of a person is always detected, and if one or two legs are found under a torso, the torso is judged as a true positive. By combining detection results of multiple layered LRFs, we can reduce the number of false positives.

Torsos and legs are typically detected as a segment separated from background by finding gaps in range data [65, 56]. However, in populated environments, torsos and legs are not always separated from background or another torso/leg. They are also often partially occluded by another objects. Our method first detects gaps of range data for clustering (see Fig. 2.5(a)), and then finds break points of merged torsos/legs in range data using two threshold values Δw and Δd (see Fig. 2.5(b)). For a point in a cluster, if two points separated from the point by Δw on both sides are closer by Δd to the robot than the point, the point is treated as a break point,



(A) clustering (B) splitting

FIGURE 2.5: Torso and leg detection procedure.



(A) usual (B) partially-occluded (C) merged

FIGURE 2.6: Detected torso candidates. Each green circle indicates the position of a torso candidate.

and the cluster is split at that point. We then apply a size filtering to all the clusters to detect torso/leg candidates. Fig. 2.6 shows examples of detected candidates for torso.

The detected candidates are classified into torso/leg and other objects using Arras's method [29] and Zainudin's method [56], respectively. Features which represent the shape of the clusters are extracted, and then the classification is performed by machine learning method, such as SVM [66] and Adaboost [67].

People Tracking

We adopt a simple procedure for temporal data association of detected persons, based on Kalman filter with a constant velocity model and a nearest neighbor (NN) data association. This works well in the majority of tracking cases. If a person is occluded by another person for several seconds, however, it often fails to track the person due to an incorrect data association. We thus take occlusions of persons into account in data association as follows.

We model each person by a circle located at the position predicted by the Kalman filter, and test whether it is occluded or not. We first predict the range data which should be obtained from the circle, and then the predicted range data for the circle are compared with the actual observed range data. If more than a half of the actual range data are closer to the robot than the predicted range data, the person is considered as occluded. The occluded persons are not associated with the detected persons to prevent incorrect data association.

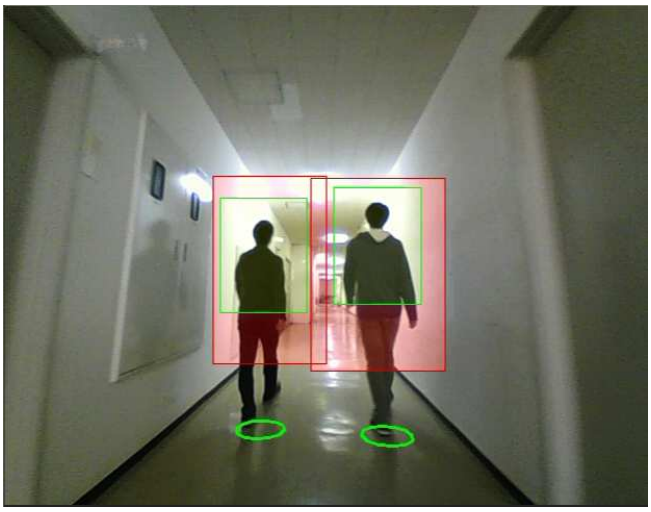


FIGURE 2.7: Detecting the person region. The green circles indicates the person position obtained by the LRF-based tracking. The red transparent regions are the ROI calculated from the person position. The green transparent regions are the detected person regions.

Finding People Regions on the Image

A person region on an image is required to extract features for person identification. We first calculate a Region of Interest (ROI) from a person position obtained by the LRF-based tracking, and then detect the upper body of the person from the ROI using the cascaded HOG classifier [68]. To calculate the ROI, we model the person as a cylinder located at the person position and project the cylinder into the image (see Fig. 2.7). The detected regions are used for extracting the person features.

Chapter 3

Person Identification

Person identification is one of the fundamental functions for attendant robots. Robots have to keep following and watching the target person during attendance. If they lose the track of the target person, they have to find and re-identify the target to continue the service. In this chapter, we propose three person identification methods for person following robots. We consider a non-cooperative scenario, where we can use only devices on the robot, and a cooperative scenario, where we can let the target hold a device (e.g., smartphone) and use the signal obtained by the device to identify the target. Depending on the use case, we can choose one of them to realize a robust person following service.

In Sec. 3.1, we propose an identification method based on appearance features improved by a deep convolution network approach. In Sec. 3.2, by combining range and vision data, illumination independent gait and height features are introduced. They are be incorporated with appearance features to reliably identify the target under severe illumination conditions. Sec. 3.3 describes a method based on the matching of foot strike timings obtained by LRFs on the robot and a smartphone held by the target.

3.1 Person Identification based on Convolutional Channel Features

3.1.1 Related Work on Appearance Feature-based Person Identification

In cases of mobile robots, the most standard feature for person identification is the appearance, since it is descriminative and easy to obtain. Many appearance features are used in image-based identification, for example HSV, Lab and XYZ color space histogram [69, 70], Haar-like [28], HOG [71], LBP [71] and SIFT [14] features. It has been proven that the combination of such appearance features and online learning methods works very well for the person following task [72, 28, 71]. 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 non-similar 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 [28], Local Binary Patterns (LBP) [72], edge features [71] 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 [73, 74]. However, a few works [75] applied such CNN-based methods to mobile

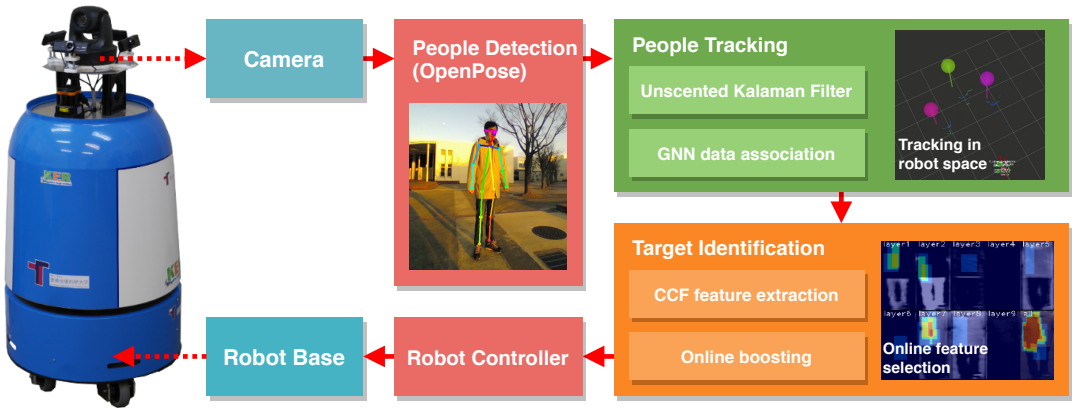


FIGURE 3.1: The proposed person tracking and identification framework with a monocular camera.

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 [76], 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) [77]. 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 work, we introduce CCF to person identification for mobile robots to take advantage of deep representation while keeping the processing cost low.

3.1.2 Proposed System

Here, we propose a person identification method based on Convolutional Channel Features. By combining the identification method with the monocular vision-based tracking method described in Sec 2.2.1, we realize a person following framework which relies solely on a monocular camera (see Fig. 3.1).

In this framework, we first detect persons with *OpenPose*, a deep neural network-based skeleton detector [59], and the detected people are tracked by the vision-based people tracker in Sec. 2.2.1. Then, a person identification method based on the combination of Convolutional Channel Features [77] and Online boosting [78] 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-identify the target person among surrounding persons with the online learned appearance model. The entire system is designed so that it can be 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.

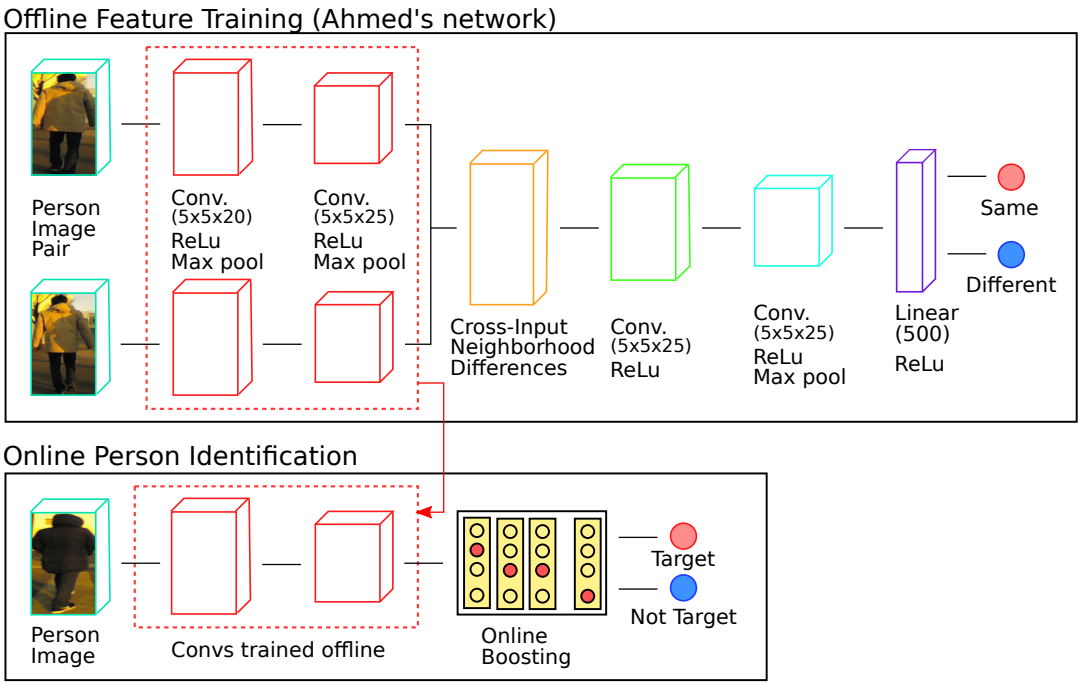


FIGURE 3.2: Convolutional Channel Features-based person identification framework. We take the first two layers of a network for person re-identification and use them to extract features for online person identification.

3.1.3 Convolutional Channel Features

To take advantage of deep CNN-based feature representation, we employ Convolutional Channel Features (CCF) [77] instead of traditional appearance features which have been used for mobile robots, such as color histograms [23], haar-like [29], and edge features [71]. 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.

In this work, we train Ahmed's network for person re-identification [73] 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.3.2). 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 differences 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 second convolution filters 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 [79] and CUHK03 [80]. 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 section, the CCFs taken from the original and the tiny version networks are denoted as CCF25 and CCF10, respectively.

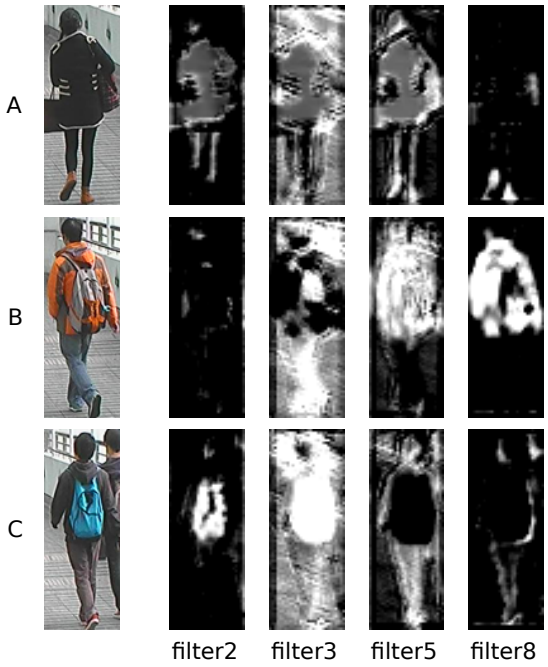
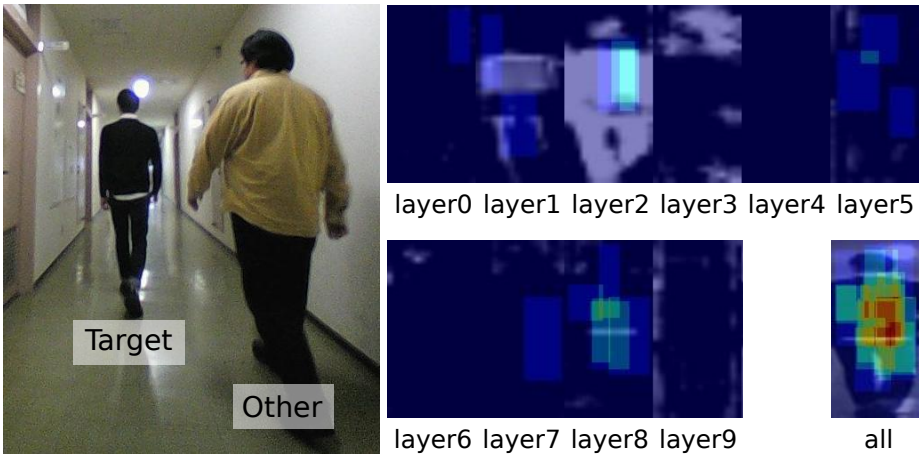


FIGURE 3.3: Feature maps extracted by CCF10. Each filter shows strong responses for different color properties.



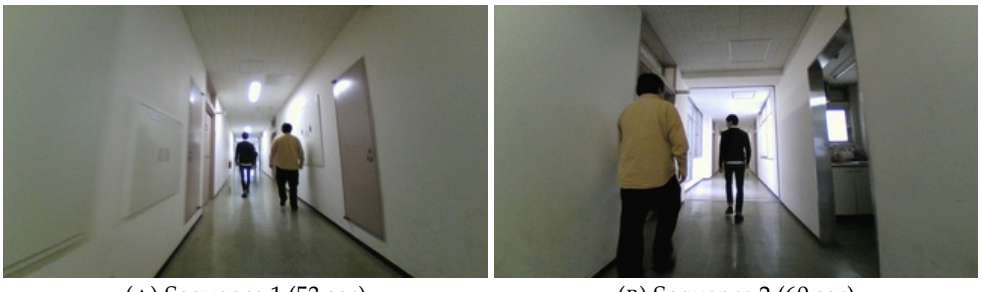
(A) A snapshot of the environment. (B) Features selected by online boosting. The selected regions are superimposed on the feature maps.

FIGURE 3.4: An example of features selected by online boosting. The discriminative regions, the upper body regions in this case, are automatically selected.

Fig. 3.3 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.

3.1.4 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 [28], we employ online boosting [78] to construct the classifier. Online boosting constructs an ensemble



(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 3.5: 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.

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. 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 of them are selected to construct an ensemble. Fig. 3.4 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.

3.1.5 Evaluation

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. 3.1). For validation, we also recorded LRF data in addition to the images in this dataset. Fig. 3.5 shows snapshots of the dataset. We controlled the robot manually and made it follow a target person in indoor and outdoor environments. We

TABLE 3.1: Person identification evaluation result. Bold indicates best results.

Sensors		Duration [sec] LRF + Camera			Camera
Features		Haar Lab [28] *	CCF10 [24]	CCF25 [24]	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)
* [28] without depth images.

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.

We evaluate the proposed monocular person identification framework with CCF10 on this dataset. To compare the proposed vision-based tracking method with the laser-based method, we also run the laser-based system with CCF10 and CCF25-based identification. On the laser-based system, the combination of Haar-like features on intensity images and *Lab* color histograms is also evaluated. This is almost identical to [28] except that it does not use Haar-like features on depth images.

TABLE 3.2: 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

FIGURE 3.6: The scene where the proposed system failed to detect the target person.

Table 3.1 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 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 processing 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

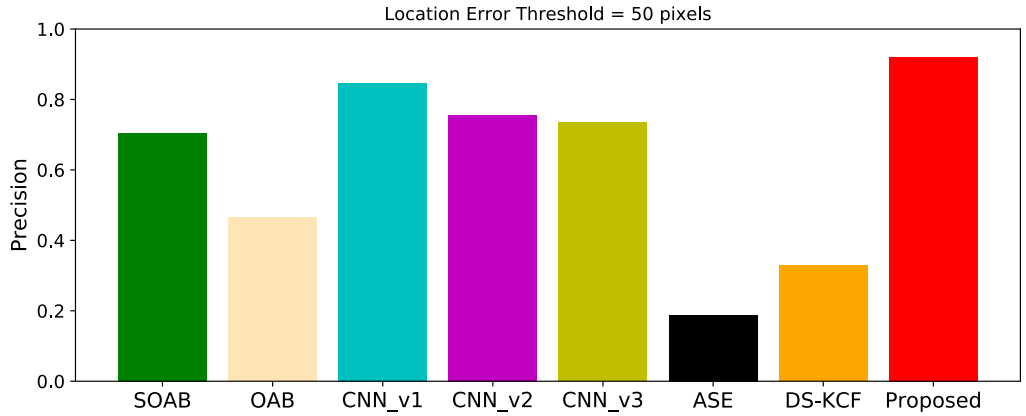


FIGURE 3.7: Target person localization evaluation result on the dataset [75].

filters. As a result, CCF10 shows a better result than CCF25 in this case.

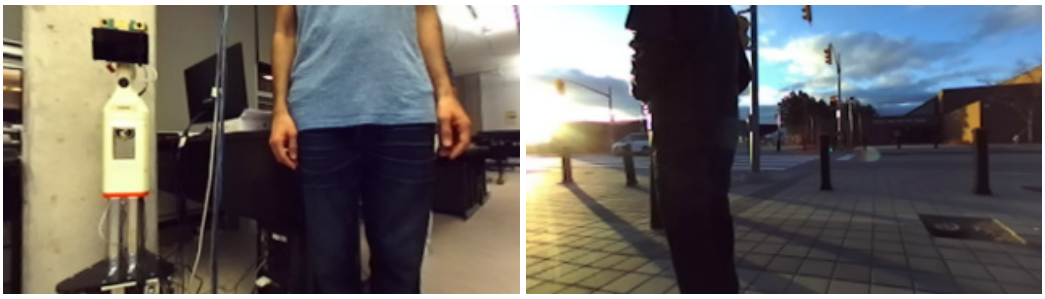
The result of the monocular vision-based system with CCF10 is comparable but a bit worse than the result of the laser-based system because the vision-based system failed to detect the target person when he was very distant from the robot (see Fig. 3.6). This result suggests that the laser-based system 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. 2.2.1, the vision-based method can track persons up to 10 [m] depending on the camera characteristics, and we consider that, during a following task, the distance between the target person and the robot would not get so long. Furthermore, a target person search approach like [81] 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 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 took about 1 sec for each frame and was far from real-time performance.

Table 3.2 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).

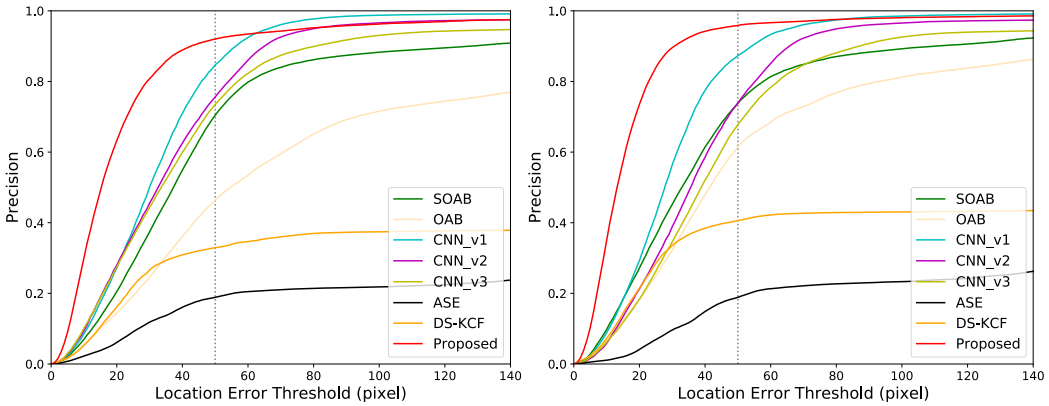
Person Identification Evaluation on a Public Dataset

We evaluated the proposed monocular vision-based framework on a public dataset for person following robots [75]. This dataset consists of 11 sequences acquired with



(A) "*lab_and_seminar*" (B) "*sidewalk*"

FIGURE 3.8: 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. 3.8.

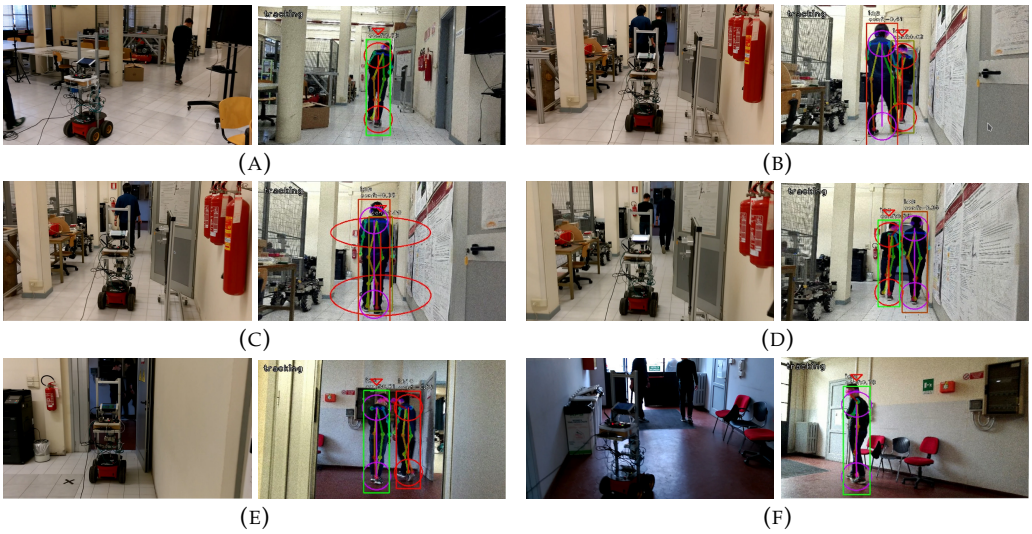
FIGURE 3.9: Target localization precision vs location error threshold.

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 severe 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 [75]. *OAB* [82] and *ASE* [83] are object tracking algorithms for monocular cameras, while *SOAB* [72], *DS-KCF* [84] are tracking algorithms for stereo cameras. There is also a convolutional neural network-based tracking algorithm for stereo images and its variations [75]. *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 network for regular RGB images. All the networks output the similarity of an input image region to the target person.

Fig. 3.7 shows the evaluation result. Following the evaluation procedure in [75], 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



(A) (B) (C) (D) (E) (F)

FIGURE 3.10: 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.

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. 3.8), 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. 3.9 (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 the target is too close to the camera, the precision under large thresholds is worse than the other state-of-the-art method (CNN_v1). Fig. 3.9 (b) shows the evaluation result where the two sequences shown in Fig. 3.8 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.

This result and the result in Sec. 3.1.5 suggest that a drawback of the vision-based method is the detection of too close and too distant persons. However, we consider that we can improve the detection rate of close persons by using a wide view angle camera, and we can employ a target person search approach [81] to re-identify a distant person.

3.1.6 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. We used the mobile robot shown in Fig. 2.3 equipped with a Jetson TX2. All the modules including the person detection, tracking, identification, and robot controller run on this board, thus, we did not use any other computers in this experiment.

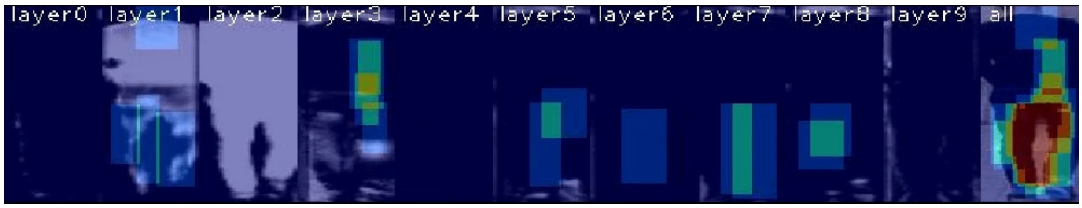


FIGURE 3.11: Features selected by online boosting during the person following experiment.

Fig. 3.10 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. 3.10 (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. 3.10 (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. 3.10 (d)). Although there was a significant illumination change when the target moved out from the room (Fig. 3.10 (e)(f)), the appearance model was updated adeptly, and as a result, the robot successfully followed the target person.

Fig. 3.11 shows the features selected by online boosting during the experiment. We can see that the classifier focused on the trousers region to robustly identify the target in this case.

3.2 Person Identification using Color, Height, and Gait Features

3.2.1 Related Work on Soft-Biometric Features for Person Identification

Many appearance features are used in image-based identification, for example HSV, Lab and XYZ color space histogram [69, 70], Haar-like [28], HOG [71], LBP [71], SIFT [14], and deep learning-based [77] features. Those features are, however, not applicable to severe illumination environments such as a strong backlight or darkness, where colors and edges are not reliably obtained (see Fig. 3.12, for example). It is therefore necessary to combine other features, including those from other sensors, for more robust identification.

Person identification using gait analysis has recently become popular [85, 86, 87]. These works extract and use frequency components from silhouette images of a walking person for identification. Since they assume a static background, these methods cannot be directly applied to mobile robots. Little has been proposed for gait analysis using range data [88, 89, 90]. Cifuentes et al. [88] measured the gait features, such as leg distance and leg orientation, from a mobile robot to realize a smooth human-robot interaction. The relative position between the robot and the person is, however, very limited for avoiding that legs are occluded by the opposite leg. Nakamura et al. [89] and Song et al. [90] put several LRFs on the ground and extracted the gait feature from these data. Since a mobile robot has a single viewpoint, a leg is often occluded by the other leg; the measured gait may be degraded due to this occlusion.

Height features are used for well calibrated and fixed cameras [91]. Since the height of a person is fixed and specific to the person, it is suitable for person identification. In the case of mobile robots, however, it is difficult to measure the height of a person using only one camera because the distance to the person can change largely. In order to use the height feature for mobile robots, another sensor which provides the distance is necessary.

(A) outdoor. (B) indoor.

FIGURE 3.12: Extremely severe illumination environments which mobile robots may face.

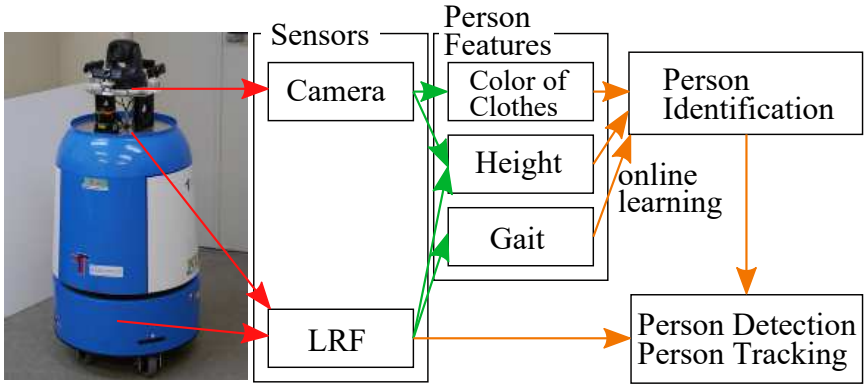


FIGURE 3.13: Person tracking and identification system.

3.2.2 Proposed System

Here, we propose a method of robustly identifying a specific person using LRFs and cameras. In order to ensure the redundancy of features in identification, we introduce two illumination-independent features, height and gait, in addition to appearance features. We combine these features to realize a robust person identification even in severe illumination environments.

Fig. 3.13 shows an overview of the proposed system. The method first tracks people in range data obtained from LRFs by using the method described in Sec. 2.2.2, and then identifies a specific person. The color feature is extracted from images while the gait feature is extracted from range data. The height feature is obtained by combining images and range data. The proposed method combines these features in order to identify the person in any environmental conditions.

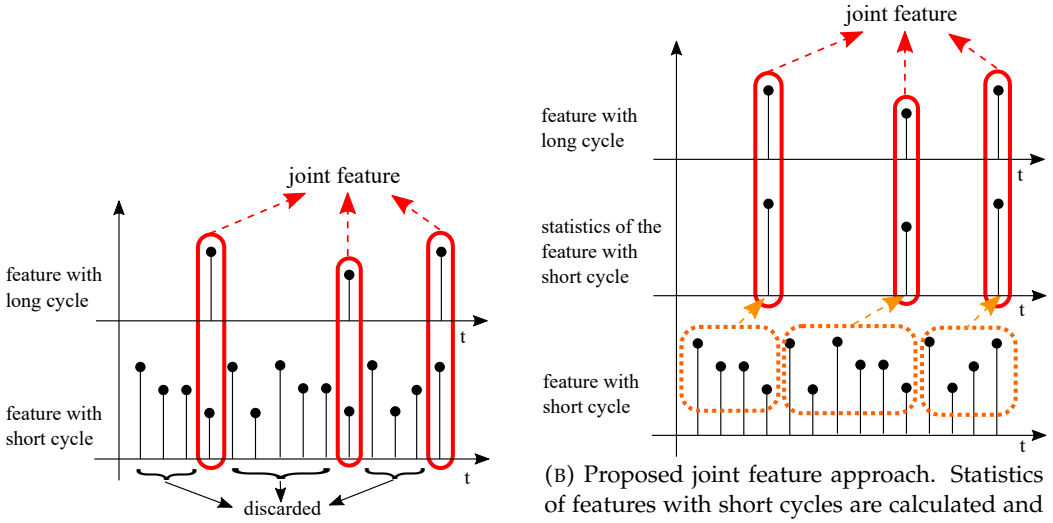
3.2.3 Person Identification Framework

Here, we briefly describe our person identification framework and a joint feature approach. To identify the person, we employ the color, the height, and the gait feature as it will be explained in detail in Sec. 3.2.4 and 3.2.5. Those features are merged into a joint feature and learned by online boosting [78].

Person Identification with Online Boosting

Online boosting is one of the online learning methods which constructs an ensemble of weak classifiers and uses it as a strong classifier. In our case, each weak classifier uses only one of the three features; appearance, gait or height. Since online boosting selects the best weak classifiers, only the effective features are used for person identification. For example, when they are in a severe illumination environment, the color feature is not effective and only the height and the gait features are used in the classifier. As a result, we can obtain a reliable person classifier even in a severe illumination environment (see experiments in Sec. 3.2.6).

While the target person is tracked by the LRF-based tracker, the person classifier is updated with observed features. While the LRF-based tracker is losing the target, updating of the person classifier stops and the robot looks for the target person using the latest person classifier. If a person who is judged as the target person by the classifier is found, the robot sets the person as the target to track, and resumes tracking.



(A) Traditional joint feature approach. Latest features are simply concatenated.

(B) Proposed joint feature approach. Statistics of features with short cycles are calculated and then combined with the feature with the longest cycle.

FIGURE 3.14: Comparing joint feature approaches.

Joint Feature for Online Boosting

The observation cycles of the proposed features are largely different. To apply the features to online boosting, they have to be synchronized and merged into a single joint feature. There are basically two approaches to synchronize features: synchronizing features to the one with the shortest or the longest cycle.

If we take the first approach, while the feature with the shortest cycle is varying, features with long cycles are kept constant or interpolated. Weak classifiers using those with long cycles are updated by one observation until a new observation is obtained. It may cause an overfitting. We thus take the second approach.

In a traditional way for the second approach, every time the feature with the longest cycle is obtained, latest feature values are simply concatenated to construct a feature vector [92]. In our system, however, the observation cycles of the proposed features are very different from each other; those of the color and the height feature are about 30 msec long, while that of the gait feature is about 500 msec long. By using only the latest feature values, a large amount of observations with short cycles are discarded (see Fig. 3.14(a)), and the identification result may be degraded. Therefore, we also make use of the values of features with shorter cycles obtained during an interval of the feature with the longest cycle by calculating their statistics and concatenating them with the feature with the longest cycle (see Fig. 3.14(b)).

In this framework, the statistics of two features, the height and the gait, are calculated. Since the height of a person is fixed, the distribution of the height feature can be expected to be unimodal. On the other hand, if we observe a person for a while, the distribution of the color feature may become multimodal due to illumination changes. However, in our case, the duration for summarizing the features is about 0.5 [sec] (i.e., the observation cycle of the gait feature). We assume that the duration is small enough to model the distribution of the color features as unimodal. We thus employ mean and standard deviation to summarize the features.

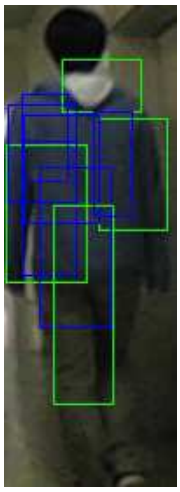


FIGURE 3.15: An example of regions for color histogram extraction selected by online boosting. The blue rectangles are the regions for hue histogram extraction, and the green rectangles are the regions for saturation histogram extraction.

3.2.4 Image-based Person Identification

Color Feature

Color features can easily be extracted from an image and are effective for identifying a person by their clothing color [69, 28, 15]. Texture and shape features, such as HOG [71] and SIFT [14], are also used for a more robust person identification. However, all such appearance features are weak under severe illumination environments [93]. We thus use only a color feature as an appearance feature for simplicity.

Color histogram is one of the most popular representations for color modeling. We use a hue-saturation histogram (HS-histogram) to reduce the effect of light intensity changes. To obtain a histogram, we follow Luber's histogram extraction approach [28]. A HS-histogram is constructed from pixels in a rectangular region with randomized positions and sizes in the person region. By online boosting [78], histogram extraction regions are sampled randomly, and regions with better identification rate are used for constructing an ensemble of classifiers. An example of histogram extraction regions generated by online boosting is shown in Fig. 3.15.

Height Feature

The height of a person can be used as another feature for person identification. Even if there are multiple persons with similar heights, the height is useful for reducing the number of candidates for the target person. To calculate the height of a person, we first determine the topmost position (i.e., *sinciput* of the head region) in the image, and then estimate the height using the camera geometry.

A saturation-intensity histogram of a hair region is computed from the hair images in advance, and then a Gaussian mixture model (GMM) is fitted to the histogram. Hair images are collected from about fifty people in various environments, and the total number of hair images are about two hundred. Since we collected hair images from Asian people, most of pixels will be the ones with zero saturation (black or gray pixels). We thus fit a separate univariate GMM to the intensity distribution of zero saturation pixels. The resultant GMM is used as the hair color model (see Fig. 3.16 and Fig. 3.17). Currently, the hair color model is specialized for people

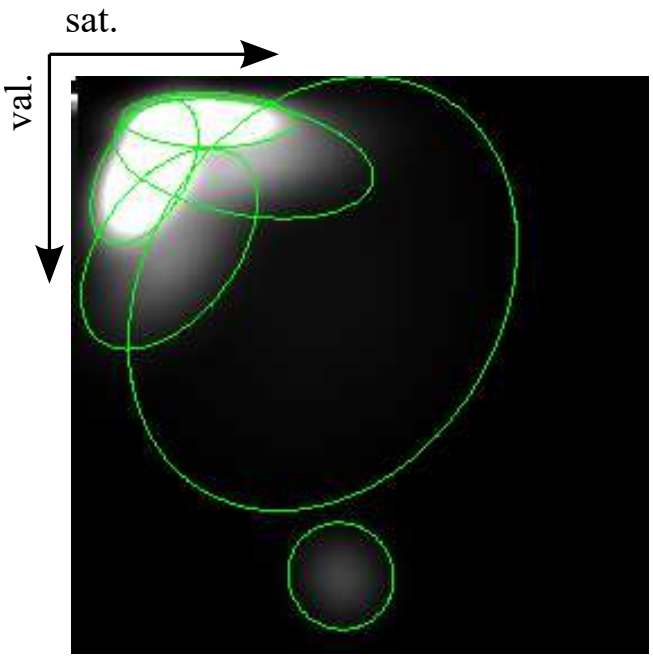


FIGURE 3.16: Hair color model. The green circles indicate gaussian distributions. The gaussian in the high value region corresponds to bright pixels caused by direct reflections of light.

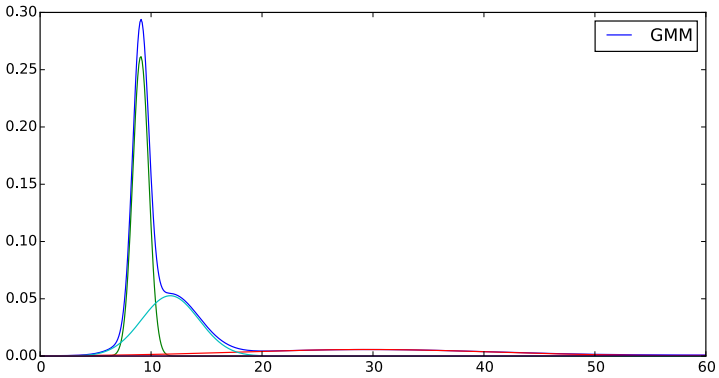


FIGURE 3.17: Hair color model for zero saturation pixels. The blue line indicates the hair color model and the rest lines indicate the gaussian distributions which compose the model.

with black or gray hair. However, the model can be extended for other people by adding their hair images.

We make two images from an input image, one representing the similarity of hair color and the other representing the magnitude of the gradient, and calculate the pixel-wise product of the images. The pixel which has the highest product value is considered as the sinciput of the person (see Fig. 3.18).

The sinciput position in the image is combined with the person position obtained by the LRF-based tracking to calculate the person height. The relationship between

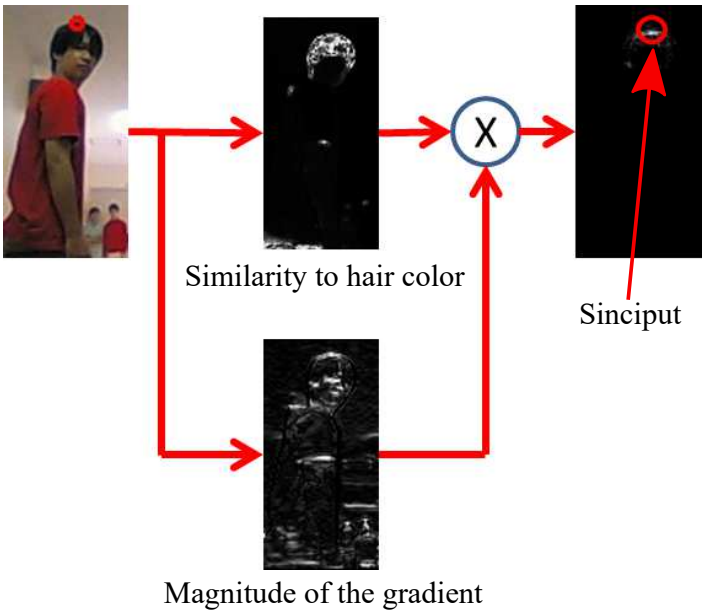


FIGURE 3.18: Sinciput detection procedure.

the 3D coordinate relative to the camera (X, Y, Z) and the projected screen coordinate (u, v) in the pinhole camera model is given by:

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}, \tag{3.1}$$

where (f_x, f_y) is the focal length, and (c_x, c_y) is the center point of the image. From this equation, we obtain:

$$Y = \frac{Z(v - c_y)}{f_y}. \tag{3.2}$$

The depth Z between the camera and the person is obtained by the LRF-based tracking, and v is the sinciput height in the image. By putting these values into eq. (3.2), we obtain the persons height.

To reduce the effects of a failure of the sinciput detection, we apply a robust estimation to the person height calculation. We adopt the M estimation with Tukey's biweight function [94] to estimate the person's height.

3.2.5 LRF-based Person Identification

Gait Feature

In computer vision, gait recognition has been studied widely [85, 86, 95]. It is, however, difficult to apply their methods to mobile robots since they assume a static background to extract silhouette images of a walking person. By using an RGB-D camera, such as Kinect, we can separate the person region from the background region, and then extract gait features [96, 97]. However, Stone et al. reported that the gait analysis using depth images shows a lower accuracy than those using RGB images [96]. Furthermore, infrared depth cameras, like Kinect, are not usable in outdoor scenes.

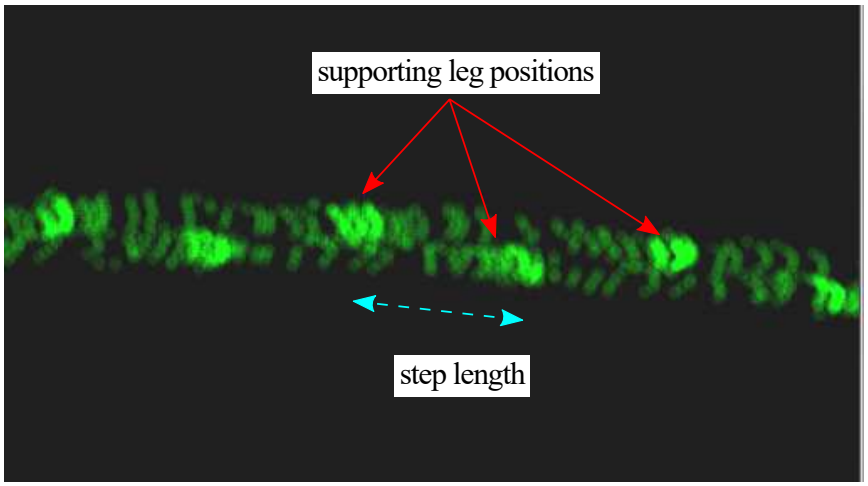


FIGURE 3.19: Accumulated range data of the legs of a walking person. High-density regions are considered as the over around the supporting legs.

When a person is walking, the legs of the person swing and stop alternately. The interval when a leg is stopping is referred to as *stance phase*, and the interval when a leg is swinging is referred to as *swing phase* [98]. During the stance phase, the leg which stops and supports the body of the person is referred to as a *supporting leg*. If we can obtain the supporting leg positions (where the leg touches the ground), we can calculate gait features, such as a step length and a stance width, from these positions.

Nakamura et al. [89] proposed a method of detecting the supporting leg positions from LRF data. They observed the legs of walking persons by several LRFs from different directions at a railway station and accumulated range data over time. Since the supporting leg positions have high accumulated values, they are extracted by Mean shift method [99]. Fig. 3.19 shows an example accumulation of range data; supporting leg positions can be found at high-density positions. We basically use their approach but a difference is that a mobile robot has a single viewpoint for LRF. This causes occlusion of supporting legs by the other ones, which may degrade the spotting of supporting leg positions in the accumulated range data.

We thus develop a method of reliably spotting support leg positions based on maximum likelihood estimation which takes such occlusions into account.

Let $X = [x_1, y_1, \cdots, x_n, y_n]$ be positions of supporting legs, $Y = [x'_1, y'_1, \cdots, x'_n, y'_n]$ be their observed positions, and $\Sigma = [\sigma_1^2, \cdots, \sigma_n^2]$ be the observation variances. The Likelihood function L is defined as:

$$L = \prod_{i=1}^{n} \frac{1}{2\pi\sigma_i^2} \exp\left(-\frac{(x'_i - x_i)^2 + (y'_i - y_i)^2}{2\sigma_i^2}\right). \tag{3.3}$$

We minimize the following objective function J.

$$\begin{aligned} J &= -\log L, \\ &= \sum_{i=1}^{n} \log 2\pi\sigma_i^2 + \sum_{i=1}^{n} \frac{1}{2\sigma_i^2} \left\{(x'_i - x_i)^2 + (y'_i - y_i)^2\right\}. \end{aligned} \tag{3.4}$$

Since the step length of a person at a stationary walk is constant [95], we assume that and obtain:

$$(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2 = \text{const.}, \quad i = (1, 2, \cdots, n-1). \tag{3.5}$$

From this equation, we obtain the following constraint function g_i:

$$g_i = (x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2 - (x_i - x_{i-1})^2 - (y_i - y_{i-1})^2 = 0, \quad i = (2, 3, \cdots, n-2, n-1). \tag{3.6}$$

According to the method of Lagrange multiplier, we define the following function.

$$F = J - \sum_{i=2}^{n-1} \lambda_i g_i. \tag{3.7}$$

Then we find a set of leg positions which satisfies the following equations:

$$\frac{\partial F}{\partial x_i} = 0, \ \frac{\partial F}{\partial y_i} = 0, \ \frac{\partial F}{\partial \lambda_i} = 0. \tag{3.8}$$

The partial differentiations of F are introduced as following equations. Note that $\lambda_i = 0$ for $i \leq 0$.

$$\frac{\partial F}{\partial x_i} = -\frac{1}{\pi\sigma_i^2}(x_i' - x_i) - 2\lambda_i(x_{i+1} - x_i) + 2\lambda_{i-1}(x_{i+1} - x_{i-1}) - 2\lambda_{i-2}(x_i - x_{i-1}), \tag{3.9}$$

$$\frac{\partial F}{\partial y_i} = -\frac{1}{\pi\sigma_i^2}(y_i' - y_i) - 2\lambda_i(y_{i+1} - y_i) + 2\lambda_{i-1}(y_{i+1} - y_{i-1}) - 2\lambda_{i-2}(y_i - y_{i-1}), \tag{3.10}$$

$$\frac{\partial F}{\partial \lambda_i} = x_{i+2}^2 - 2x_{i+1}(x_{i+2} - x_i) - x_i^2 + y_{i+2}^2 - 2y_{i+1}(y_{i+2} - y_i) - y_i^2. \tag{3.11}$$

We use five walking steps for estimation of supporting leg positions and the duration of the observation is about 2.5 [sec]. We assume that the walking speed is constant for this duration.

When the robot observes a walk from a side position, a leg on the robot side is always visible while the other is sometimes occluded. We thus give the observation of the supporting leg on the robot side a small variance (i.e., high reliability) and that of the other leg a large variance (low reliability).

Fig. 3.20 shows how to determine the side of a supporting leg. We draw a line every two positions and see if the point between them is on the same side as the robot. In the case of the figure, p_{t-1} is given a small variance while p_t a large variance.

We use the pair of step length and walking speed as the gait feature since those are determined by physical characteristics of the person (e.g. weight, height, and lengths of limbs) and specific to an individual [95].

Gait Estimation Evaluation

We describe an evaluation of our gait estimation method. We placed markers evenly on the ground every 0.6 m and a person walked by stepping at every marker so that we could obtain a constant step length. The robot observed the walk both from the front and the side of the person for comparison.

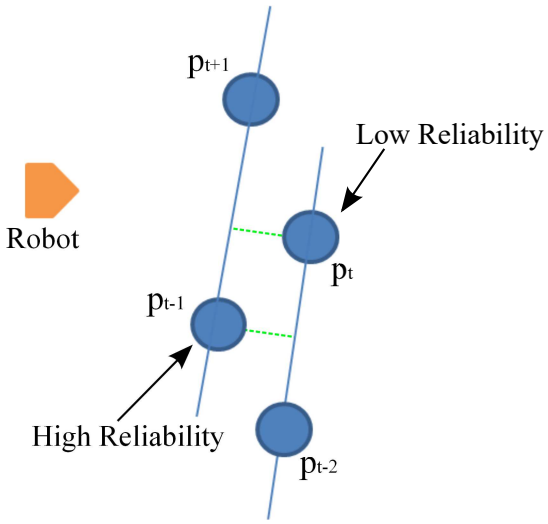
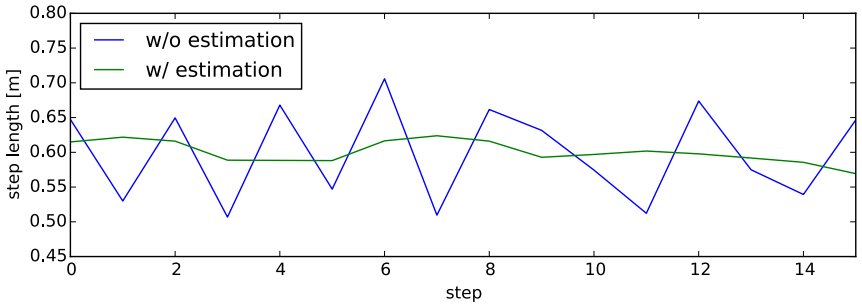
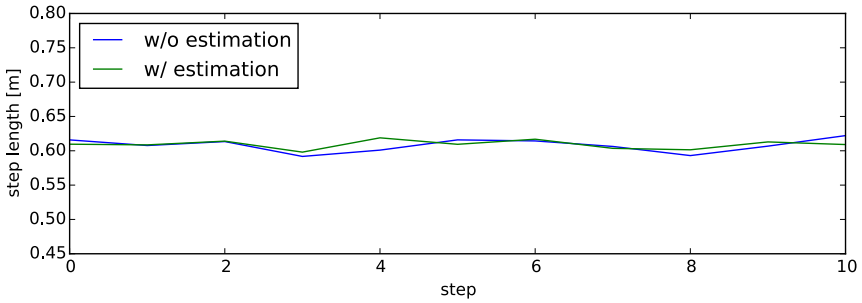


FIGURE 3.20: Assigning reliabilities to the measurement of supporting legs.



(A) Estimation from observation from side of a person.



(B) Estimation from observation from front of a person.

FIGURE 3.21: Estimated step length.

When the robot observes the person from the side (see Fig. 3.21(a)), the measured step lengths fluctuate due to the occlusion of the supporting leg. The effect of occlusion is then largely reduced by the proposed estimation method. On the other hand, when the robot observes from the front (see Fig. 3.21(b)), the measured step length is much more stable since no occlusions occur.

Table 3.3 summarizes the evaluation. The fluctuation of the observation from the side is larger than the observation from the front obviously due to the occlusion. The proposed estimation could reduce the fluctuation in the both cases.

Gait Identification Experiment

A gait identification experiment was conducted. We recorded gait data of about 30 steps long (about 20 seconds long) for eight persons at a normal walking speed as a

TABLE 3.3: Step length estimation result.

		w/ estimation		w/o estimation	
	No. of data	mean [m]	SD [m]	mean [m]	SD [m]
observation from side	42	0.6006	0.01387	0.5988	0.06469
observation from front	32	0.6078	0.00862	0.6049	0.0114

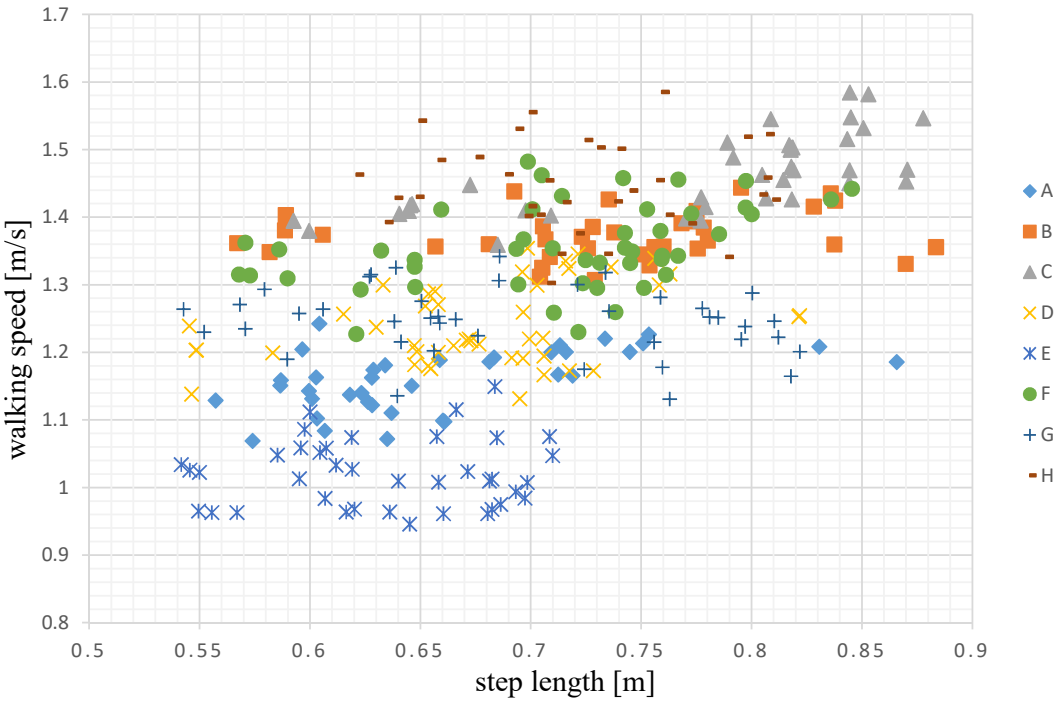


FIGURE 3.22: Observed gait data.

TABLE 3.4: Gait-based Identification Results.

test data \ model	A	B	C	D	E	F	G	H
A	0.946	0.486	0.000	0.676	1.000	0.054	1.000	0.270
B	0.711	0.658	0.553	0.789	0.184	0.947	0.500	0.474
C	0.361	0.667	0.944	0.694	0.056	0.722	0.167	0.972
D	0.978	0.609	0.022	0.848	0.761	0.500	0.783	0.174
E	0.564	0.231	0.000	0.103	1.000	0.000	1.000	0.282
F	0.854	0.563	0.375	0.833	0.313	0.917	0.500	0.542
G	0.878	0.366	0.000	0.805	0.780	0.512	0.854	0.268
H	0.529	0.412	0.882	0.500	0.118	0.824	0.206	0.882

training set and constructed a classifier for each person using online boosting [78]. In the experiments, the number of weak classifier selectors is five and each selector contains ten weak classifiers. Fig. 3.22 shows the gait data for training; we can see some persons (e.g., persons C, E, and H) have distinctive gaits.

We recorded another set of data in the same settings for evaluating the identification performance. Table 3.4 shows the result of the experiment. The first row indicates the constructed models, and the first column indicates the test data. Each cell

(A) Experiment 1. Persons with the similar heights and the different color clothes.

(B) Experiment 2. Persons with the different heights and the similar color clothes.

(C) Experiment 3. Persons with the different heights and the different color clothes.

FIGURE 3.23: The environments of the person identification experiment.

TABLE 3.5: Precision of person identification.

	height	gait	color	all features (traditional)	all features (proposed)
exp. 1	0.542	0.712	0.949	0.949	0.966
exp. 2	0.924	0.532	0.684	0.937	0.937
exp. 3	0.810	0.726	0.903	0.921	0.948

indicates the acceptance rate of the test data by the constructed model. For model C and D, the correct person shows a higher identification rate than the others. For model A, B, E, F, and H, the correct person's are the second highest. These results show that the gait feature is mostly effective to identify a person or to reduce the number of possible identities of a person.

The model B, however, shows the lower identification rate for the correct person, since the gait data of person B is in the most dense area. It is difficult to identify the person using only the gait feature in some cases, such as person B. This will be dealt with by combining with the other features.

3.2.6 Experiments

Person Identification Experiment

In order to compare the effectiveness of the features, we conducted person identification experiments. In the experiments, two people walk side by side while the robot is controlled manually and follows both persons and measures their person

features. To evaluate the effectiveness of the each feature, five person classifiers are constructed. These classifiers use the following features, respectively.

1. Height feature
2. Gait feature
3. Color feature
4. All proposed features with the traditional joint feature approach
5. All proposed features with the proposed joint feature approach

For the classifiers with all the proposed features, we tested two methods: one with a traditional joint feature approach and the other with the proposed one. In the all experiments, online boosting contains 10 weak classifier selectors, and each selector contains 10 weak classifiers.

The experiments are conducted in three different cases (see Fig. 3.23); two persons are with similar colors and different heights in case (a); those with different colors and similar heights in case (b); those with different colors and heights in case (c). The learning process of the classifier with all the proposed features takes about 20 msec long for one person. We tested the proposed system in this experiment, and calculate the precision of the identification. Table 3.5 shows the result of the experiments.

The classifiers using a single feature show a good precision in specific cases but not in the others. The classifiers with all the proposed features show superior performances in all cases. In addition, the classifier with the proposed joint feature shows equal or greater precision than the one with traditional one. This shows the effectiveness of the proposed joint feature.

Person Identification Experiment in Severe Illumination Environments

We conducted person identification experiments for two target persons and for two different illumination environments. Fig. 3.24 shows snapshots of the experiments. In experiments 1 and 3, most of the persons were wearing similarly colored clothes and sometimes entered shadowed areas. In experiments 2 and 4, color information is almost lost due to a strong backlight. In all cases, it is very difficult to identify the target person using color information only.

Fig. 3.25 shows snapshots of the experiment 2; The experiment was conducted in the most severe illumination environment. Green rectangles in the images indicate detected persons and the red triangles above them indicate the target person. At the beginning of the experiment, the robot learned the features of the target person and created a person classifier (Fig. 3.25 (a)), and then some persons occluded the target person (Fig. 3.25 (b)(c)). The LRF-based tracker failed to track the target person several times due to the occlusion of the person (Fig. 3.25 (d)(e)). The robot however, found the correct target person using the person classifier, and resumed correct tracking (Fig. 3.25(f)). In this experiment, the robot successfully continued to track a specific person in spite of temporarily-lost situations thanks to the height and the gait feature.

Table 3.6 shows the result of the four experiments. The total time for the experiments was about 765 [sec] and the target person was occluded by others 43 times through all of the experiments. The robot lost track of the target person 16 times due to occlusions. The person classifier, however, found the correct target person and

(A) Experiment 1. Direct light condition. Target person is subject A.

(B) Experiment 2. Backlight condition. Target person is subject A.

(C) Experiment 3. Direct light condition. Target person is subject B.

(D) Experiment 4. Backlight condition. Target person is subject B.

FIGURE 3.24: The environments of the person identification experiment.

TABLE 3.6: Result of the experiments in the severe environments.

		exp. 1	exp. 2	exp. 3	exp. 4	total	
	time [sec]	157	213	195	200	765	
	occlusion of the target [times]	11	11	8	13	43	
all features	successfully tracked [sec]	128	168	167	157	620	(81.0%)
	lost track of the target [sec]	29	45	22	43	139	(18.2%)
	tracked wrong person [sec]	0	0	6	0	6	(0.8%)
	lost track of the target [times]	3	6	3	4	16	
	wrong association [times]	0	0	1	0	1	
color feature	successfully tracked [sec]	128	102	74	105	409	(53.5%)
	lost track of the target [sec]	29	61	24	21	135	(17.6%)
	tracked wrong person [sec]	0	50	97	74	221	(28.9%)
	lost track of the target [times]	3	1	1	2	7	
	wrong association [times]	0	0	1	0	1	

resumed the tracking every time. The robot tracked a wrong person as the target for 6 [sec] (0.8% of the experiments) due to a wrong data association. However, that person was then judged not to be the target and the robot then found the correct person. Among the rest of the time, the robot correctly tracked the target person for 620 [sec] (81.0%) and looked for him while calculating the gait feature values for 139 [sec] (18.2%)

(A) (B) (C) (D) (E) (F)

FIGURE 3.25: Person identification experiment in a severe illumination environment: Red triangles above green rectangles indicate the identified target person.

The person classifier with only the color feature was also tested in the experiments. The robot with the classifier successfully tracked the target person in experiment 1. In the other experiments, however, the robot tracked wrong persons in many frames (221 [sec] (28.9%)) due to severe illumination environments.

3.2.7 Person Following Framework

Tracking Strategy

The LRF-based tracking method described above may sometimes fail to track the target person. The robot has to be able to recover from such a failure situation. We therefore define three states which switch in the operation as follows (see Fig. 3.26).

In the *initial* state, the robot measures the person features while following the target person. If a sufficient number of person features are measured, the robot

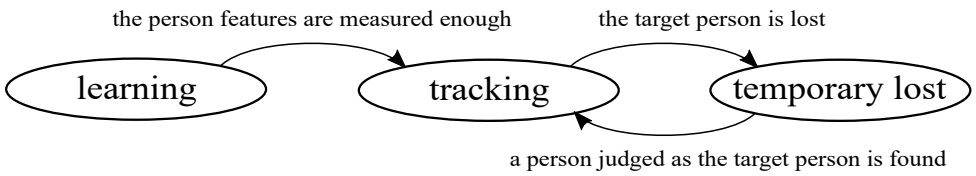


FIGURE 3.26: State machine for person following behavior.

constructs a person classifier from the features and transits to the *tracking state*. In the *tracking* state, the robot performs the usual tracking and identification. When the LRF-based tracking loses the target, the robot transits to the *temporary lost* state. In this state, while the robot is looking for the target person using the person classifier, the position of the target person is predicted from the most recent person movement, and the robot moves toward the position. If a person is judged as the target, that is, the target person is re-identified, the robot transits to the *tracking* state.

Person Following Experiment

We applied the proposed system to person following experiment. The experiment was conducted in both indoor and outdoor environments. The experimental environment is a public space in Toyohashi university of technology, and there were many ordinary persons. Fig. 3.27 shows snapshots of the experiment. The left images show experimental scenes. The right images show the images captured by the robot. The rectangular region in the upper right corner of the right images indicates range data and the conditions of the LRF-based tracker. The circles in the region indicate the tracked persons by the LRF-based tracker. The circles under the persons in the images also indicate the position of the tracked persons. Green rectangles in the images indicate detected person regions and the red triangles above them indicate the target person.

The experiment started in a populated outdoor environment. The robot followed a target person while measuring his features (Fig. 3.27(a)). Then, the robot constructed the person classifier and continued the following behavior. Several persons walked with the target person, and often occluded the target person (Fig. 3.27(b)). The LRF-based tracker lost the target person due to the occlusion (Fig. 3.27(c)). The green circle in the upper right rectangular region in the right image of Fig. 3.27(c) indicates the predicted target person position to witch the robot was moving. Once the target person appeared and walked for several steps (Fig. 3.27(d)), the robot realized that the person was the correct target to track (Fig. 3.27(e)). After the robot followed the person for a while, the target person moved to the indoor environment (Fig. 3.27(f)). While the person and the robot were moving into the indoor environment, a strong illumination change occurred (Fig. 3.27(g)) and the target person was also occluded by another person (Fig. 3.27(h)). However, the robot successfully found the target person (Fig. 3.27(i)(j)). After that, the person returned to the outdoor environment and continued the following behavior (Fig. 3.27(k)(l)).

The duration of the experiment is 920 [sec], and the LRF-based tracker lost the target person 11 times due to occlusions. However, the robot re-identified the target every time and successfully continued to follow the target throughout the experiment. The average re-identification time after the person appeared was 5.6 [sec]. Since during that time, the robot kept moving towards the predicted position of the target, it was able to find the target when he appeared again.

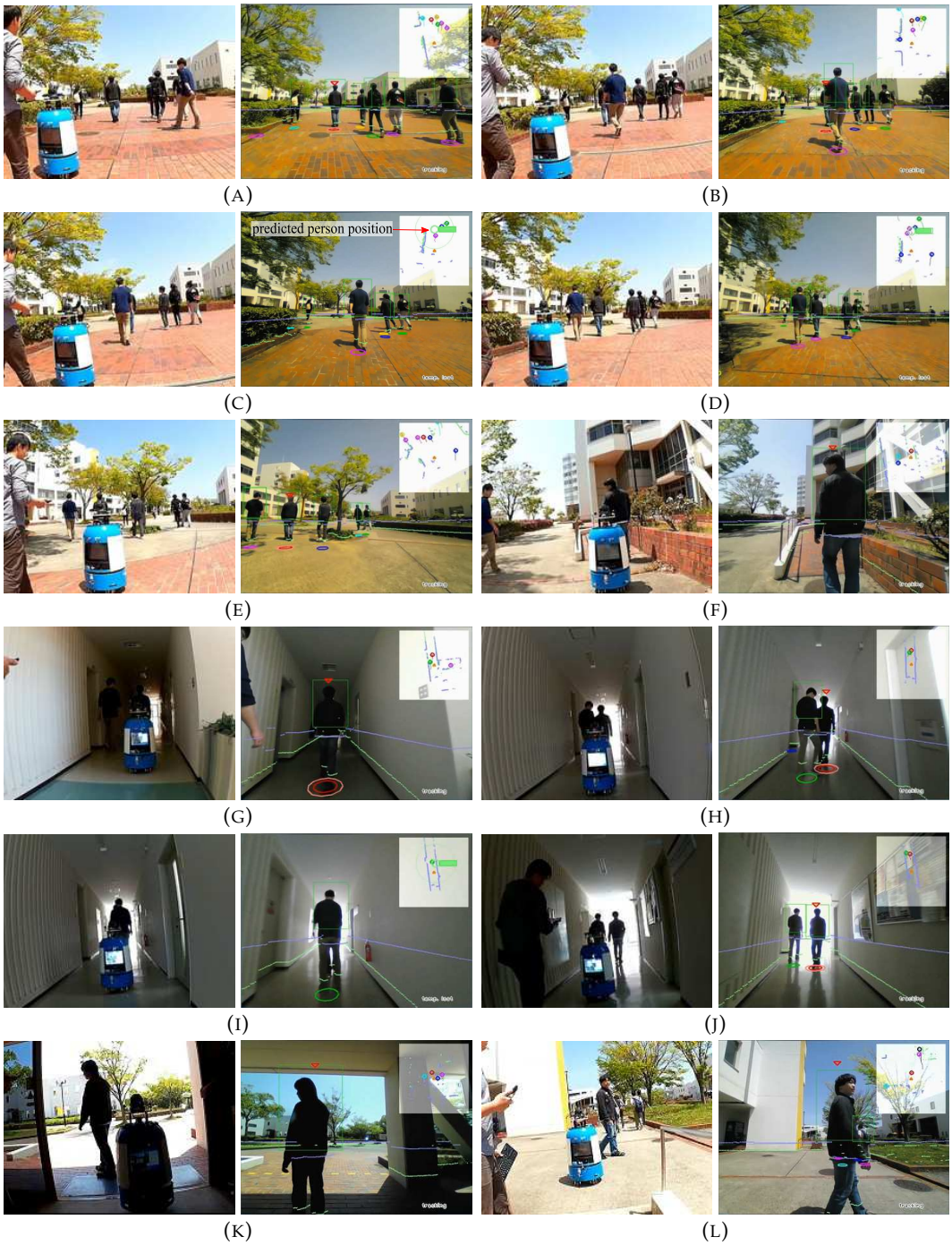


FIGURE 3.27: The person following experiment.

3.3 Person Identification based on Foot Strike Timings

3.3.1 Related Work on Wearable Device-based Person Identification

Many previous works use a combination of laser range finders (LRFs) and cameras for person tracking and identification by mobile robots [69, 28, 14]. In these works, a robot learns the appearance of the target person from cameras and uses them for identifying the person. However, if the appearance of the target person changes drastically while the person is occluded by other persons or obstacles, it is difficult for the robot to find the correct person again using only the image. If we use a device for identification and the target person holds the device, we can realize a person identification which does not suffer from any environmental changes and persistent occlusions of the target person.

Some works propose person identification methods using environment sensors and an IMU (Inertial Measurement Unit) [100, 101]. They place multiple static sensors, such as a camera and an LRF, in an environment and attach an IMU to the target person. These methods measure the walking pattern of the target person using the IMU and those of all persons in the environment using the sensors. Shiomi et al. [100] use depth cameras as environment sensors. They calculate persons' acceleration using the depth cameras and the IMU, and classify the persons' states into moving or stopping. By matching the states obtained by the depth cameras and the IMU, they identify the target person. Ikeda et al. [101] put LRFs in an environment, and tracked the legs of all the persons in the environment. They estimate the acceleration of the legs and compare them with the acceleration obtained by the IMU. By calculating the signal correlation between the accelerations, they find the person holding the IMU among others. However, we cannot apply these methods to mobile robots directly since multiple static sensors are not available.

3.3.2 System Overview

Here, we propose a person identification method based on the matching of foot strike timings for mobile robots. We estimate the foot strike timings of the target person using a smartphone held by the person and LRFs on the robot. By matching these data, the robot can reliably identify the target person in the LRF data.

Fig.3.28 shows an overview of the proposed system. The robot is equipped with two LRFs, and the target person puts a smartphone in their pocket. The smartphone is connected to the robot via wifi. The system first detects and tracks all persons around the robot using the top LRF, and then estimates foot strike timings for each

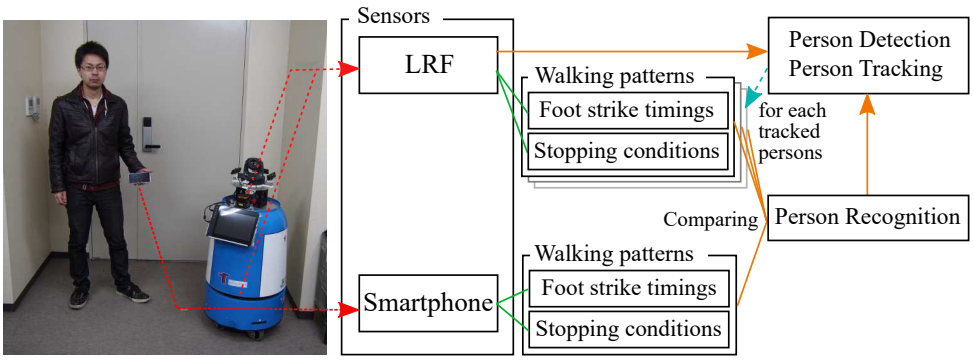


FIGURE 3.28: System overview.

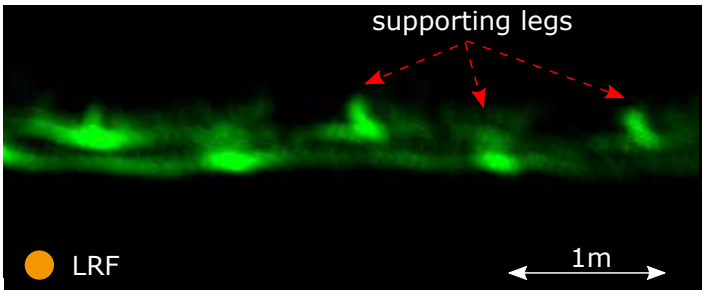


FIGURE 3.29: Accumulated range data of the legs of a walking person.

person using the bottom LRF. It also estimates the foot strike timings from the acceleration of the smartphone, and compares it with the timings of all tracked persons to identify the target person. When the target person is stopping, however, the foot strike timing is not available. We therefore judge if a person is stopping by using the LRFs and the smartphone, and the stopping states are also compared to estimate the target person. Note that the top LRF is used just to simplify the data association for people tracking. Therefore, this method essentially requires only one LRF placed at a leg height.

We define a dissimilarity measure for the foot strike timings and the stopping states from the LRFs and the smartphone, and calculate the likelihood that each person is the target. This information is integrated over time using the Bayesian inference, and the person with the highest posterior probability is judged as the target. When the LRF-based tracker loses the target person due to, for example, occlusion, it stops the estimation and starts re-identification of the target person among all surrounding persons.

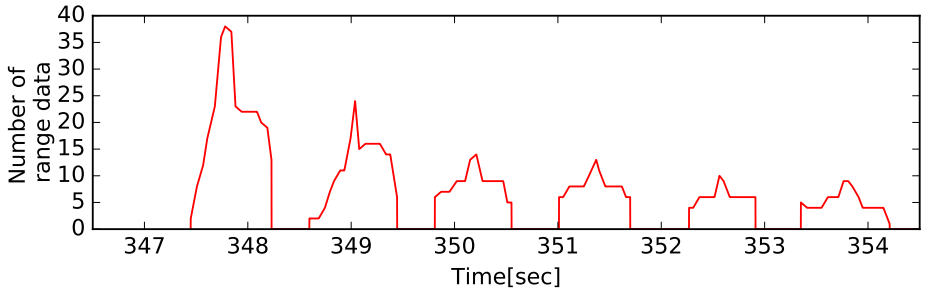
3.3.3 Estimation of Foot Strike Timings and Stopping State using LRFs

The method first detects the positions of the supporting legs of a walking person from LRF data and then estimates the strike timing from a time period where a foot is near each supporting leg position.

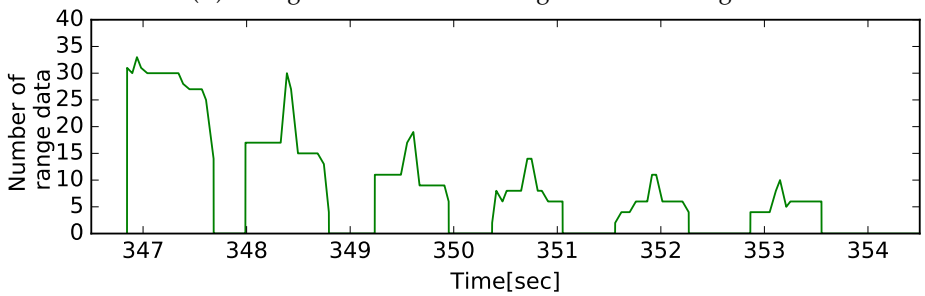
Fig. 3.29 shows an example of accumulated range data of a walking person obtained from the LRF placed at a leg height. The supporting legs of the person appear as high-density regions of range data. According to Nakamura et al. [89], we extract the supporting legs by spotting high-density regions using Mean Shift [99]. We then estimate the actual positions of the supporting legs using the maximum likelihood estimation described in Sec 3.2. We use five steps for the estimation of supporting leg positions. We assume that the walking speed is constant throughout the duration.

To estimate foot strike timings from the positions of the supporting legs, we count the number of the range data around a supporting leg at each frame, and examine how the number changes over those frames. Fig. 3.30 (a)(b) show the change of the numbers of range data for both legs. Each cluster corresponds to a foot strike. We treat the number as a weight and consider the weighted mean of times as a strike timing. Fig. 3.30 (c) shows the estimated foot strike timings.

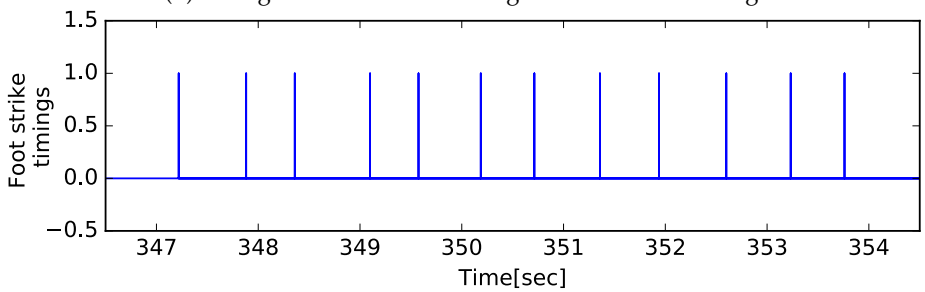
The stopping state of a person is determined by the walking speed measured by the LRF-based tracking. If the speed of a person is less than a specified threshold (currently, 0.3 [m/sec]) the person is considered to be stopping.



(A) Change of the number of range data for one leg.



(B) Change of the number of range data for the other leg.



(C) Estimated foot strike timings.

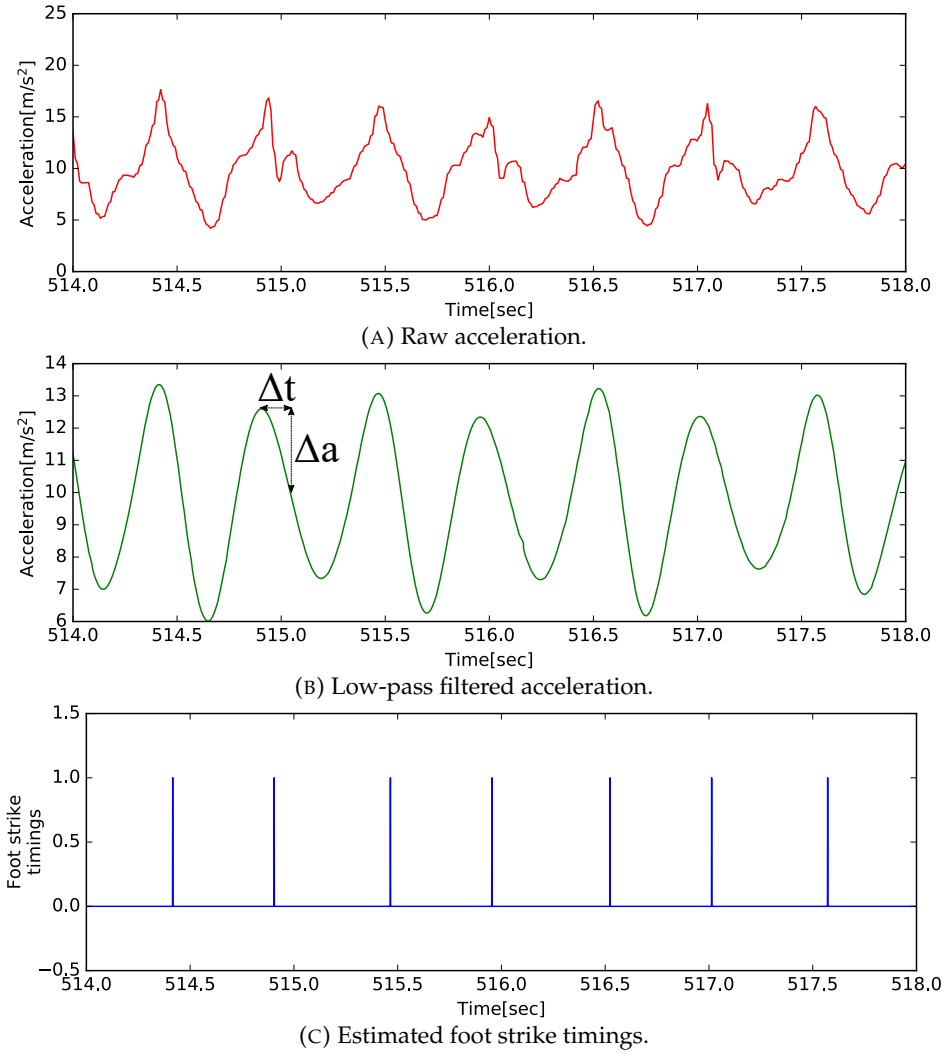
FIGURE 3.30: Estimating foot strike timings using LRFs.

3.3.4 Estimation of Foot Strike Timings and Stopping State using a Smartphone

While a person is walking, the body of the person moves up and down periodically. By detecting the peak of the acceleration of the body, we can find the foot strike timings. We use a method proposed by Li et al. [102]. They first apply an FIR low-pass filter to the acceleration data obtained by a smartphone, and then detect the peak of the filtered acceleration using two threshold values Δt and Δa (see Fig. 3.31(b)). Since their method does not depend on the position of the sensor, the smartphone can be held at any location on a person; the person can put a smartphone in a pocket or hold it in the hand. Fig. 3.31 shows an example of the acceleration of a smartphone placed in the chest pocket and the estimated foot strike timings from it. We set the cutoff frequency of the low-pass filter 3 [Hz], $\Delta t = 0.2$ [sec], and $\Delta a = 1.5$ [m/s^2].

The stopping state of a person is determined when the smartphone is judged as being stationary for a certain period of time. We use Jimenez's method [103], which uses simple thresholds of acceleration and angular velocity to judge if a sensor is stationary or not. If the smartphone is judged as stationary for 1.0[sec], we consider that the person who has the smartphone has stopped.

Fig. 3.32 shows an example of foot strike timings and stopping states when a person walks for several seconds and then stops and walks again.



(A) Raw acceleration.

(B) Low-pass filtered acceleration.

(C) Estimated foot strike timings.

FIGURE 3.31: Estimating foot strike timings using a smartphone.

3.3.5 Data Integration for Person Identification

Stopping states and foot strike timings are obtained by a smartphone for the target person and by LRFs for all persons. We compare each person's data with those of the target to find the best-matched person in the LRF data. For this purpose, we define the dissimilarity of the stopping states and the timings, which is then used for defining the likelihood function. The likelihood function is used for applying the Bayesian estimation to the target identification.

Dissimilarity Measure between LRF and Smartphone Data

To compute the dissimilarity between foot strike timings, we first associate the timings by LRF and those by a smartphone by finding the closest LRF timing for each smartphone timing. We calculate the mean of the time differences between the associated timings, and use it as the dissimilarity measure.

The dissimilarity between stopping states by LRF and smartphone is calculated by the difference between the binary patterns of stopping. We set a time window and measure the total time duration where the LRF and the smartphone patterns are

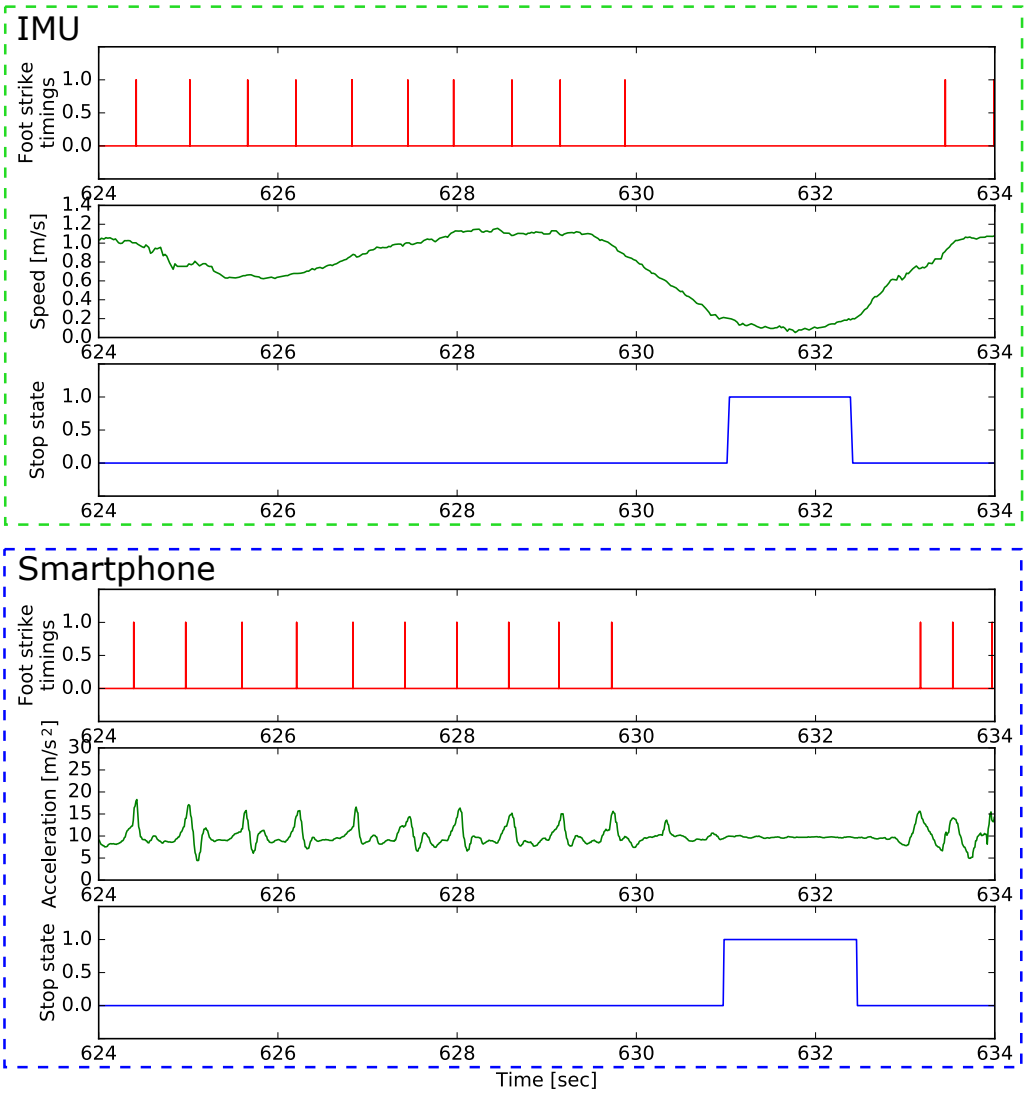


FIGURE 3.32: Foot strike timings and stopping states obtained by LRFs and a smartphone.

TABLE 3.7: Dissimilarity of foot strike timings and stopping state

	target person		other persons	
	foot strike	stopping	foot strike	stopping
# of data	381	472	1483	2486
mean	63.6 [msec]	0.0585	144.8 [msec]	0.3005
std. dev.	28.6 [msec]	0.0898	68.2 [msec]	0.2650

different. The duration is normalized by the width of the time window and used as the dissimilarity measure. We set the time window size to 5 [sec].

Table 3.7 gives statistics of dissimilarity values for foot strike timings and stopping states. This is obtained from the real experiments which were conducted under the same settings used in Sec. 3.3.6. The dissimilarities of the target person is much smaller than the dissimilarities of the others.

Bayesian Estimation for Person Identification

The target person usually shows low dissimilarities and the others high dissimilarities. The target person, however, sometimes shows a high dissimilarity due to a lack of range data or measurement errors. For a robust tracking, we use a Bayesian estimation for determining the target person.

Let $p(x_i)$ be the prior probability that the ith person in the LRF data is the target person. We define the likelihood of person x_i for an observed dissimilarity value y as:

$$p(y|x_i) = \exp(-cy), \tag{3.12}$$

where c is a constant. We calculate the likelihood values for the foot strike timing and the stopping state, and the multiplication of the two likelihood values is used as the likelihood $p(\boldsymbol{y}|x_i)$. Then the posterior probability $p(x_i|\boldsymbol{y})$ is given by:

$$p(x_i|\boldsymbol{y}) = \alpha p(\boldsymbol{y}|x_i)p(x_i), \tag{3.13}$$

where α is the normalization constant. The probability is updated every 100 [msec].

Re-detection of the Target Person

If the LRF-based tracking loses the target person, Bayesian estimation stops temporarily and the system starts to re-detect the person. This re-detection is done while the target person is walking, by searching for a person whose foot strike timings by LRFs are close enough with those by a smartphone to a high confidence.

We consider that a pair of foot strike timings matches if their difference is less than a threshold th_{tm}. Then, if the foot strike timings of a person (by LRFs) and those of the target person (by a smartphone) have at least n_{sim} matched frames in n_{test} consecutive frames, that person is considered as the target. We determine these three parameters as follows.

Fig. 3.33 shows relationship between the matching threshold th_{tm} and the matching rate of timings for the actual target person and other persons. We calculated the matching rate from a real data sequence. The experimental setting is the same as the one used in Sec. 3.3.6. In the experiments, we placed a smartphone in two difference locations, a trousers pocket and a chest pocket. As the threshold increases, the rates increase, but that for the target person much more rapidly increases. When we put the smartphone in the trousers pocket, the matching rate of the target person is less than the chest pocket case. It is strongly affected by the attached leg and the foot strike of the opposite leg becomes difficult to detect. Even in the case of the trousers pocket, however, since the matching rate of the target person is significantly larger than the others, it can be used for identifying the target person.

According to binomial distribution, we can calculate the probability that a person is identified as the target from the matching rate and arbitrarily n_{sim} and n_{test} as:

$$p_{ident}(p, n_{sim}, n_{test}) = \sum_{i=n_{sim}}^{n_{test}} \binom{n_{test}}{i} p^i (1-p)^{n_{test}-i}, \tag{3.14}$$

where p is the matching rate of the foot strike timings of the person. We calculate the probability where the correct person is identified as the target (true positive rate) and where another person is identified as the target (false positive rate).

In order to determine the appropriate parameters, we set the target true positive rate as 80 % and the target false positive rate as 5 %. Table 3.8 shows examples of

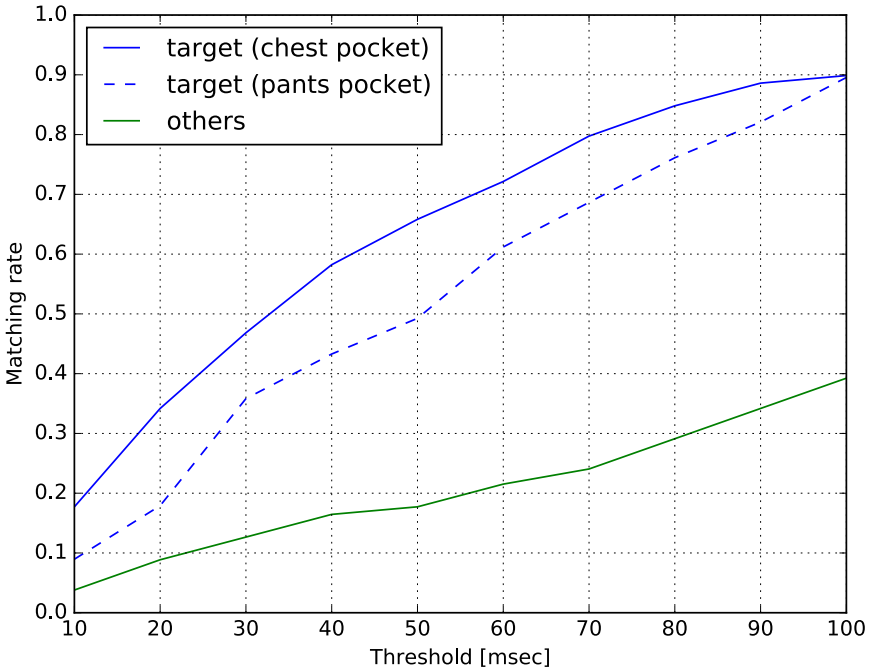


FIGURE 3.33: Matching rate.

TABLE 3.8: Candidate parameters

threshold	n_{sim}	n_{test}	true positive	false positive
70 [msec]	3	4	0.814	0.045
70 [msec]	4	5	0.831	0.027
90 [msec]	4	5	0.897	0.049
90 [msec]	5	6	0.857	0.019
90 [msec]	6	7	0.814	0.007
70 [msec]	6	9	0.910	0.008

the candidate parameters which meet the criteria. As shown in Table 3.8, if we set n_{sim} and n_{test} large enough (it means focusing on the person for a longer number of seconds), we can obtain a better re-detection performance.

For mobile robots, however, there is a limitation on re-detection time, because the target person may get far away from the robot while the robot is trying to re-detect the person. We, therefore, choose parameters $n_{sim} = 3$ and $n_{test} = 4$; then, the true positive rate and the false positive rate are estimated to be 0.814 and 0.045 respectively, and the minimum time for re-detection will be about 1.5 [sec].

Comparison with the Previous Method

We compare the identification performances of the proposed method and our previous method which uses an IMU attached on a leg of the target to detect foot strike timings [22]. We define the identification performance as the harmonic mean of the true positive rate and the inverse of the false positive rate:

$$\text{performance} = \frac{2TP \cdot (1 - FP)}{TP + (1 - FP)}, \tag{3.15}$$

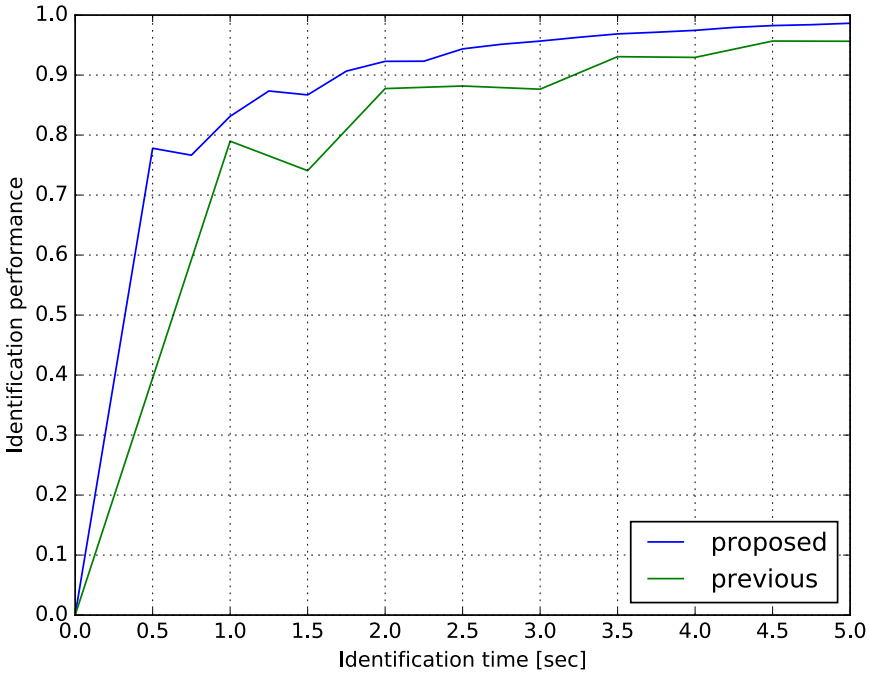


FIGURE 3.34: Relationship between the identification time and the identification performance.

TABLE 3.9: Results of the person identification experiment

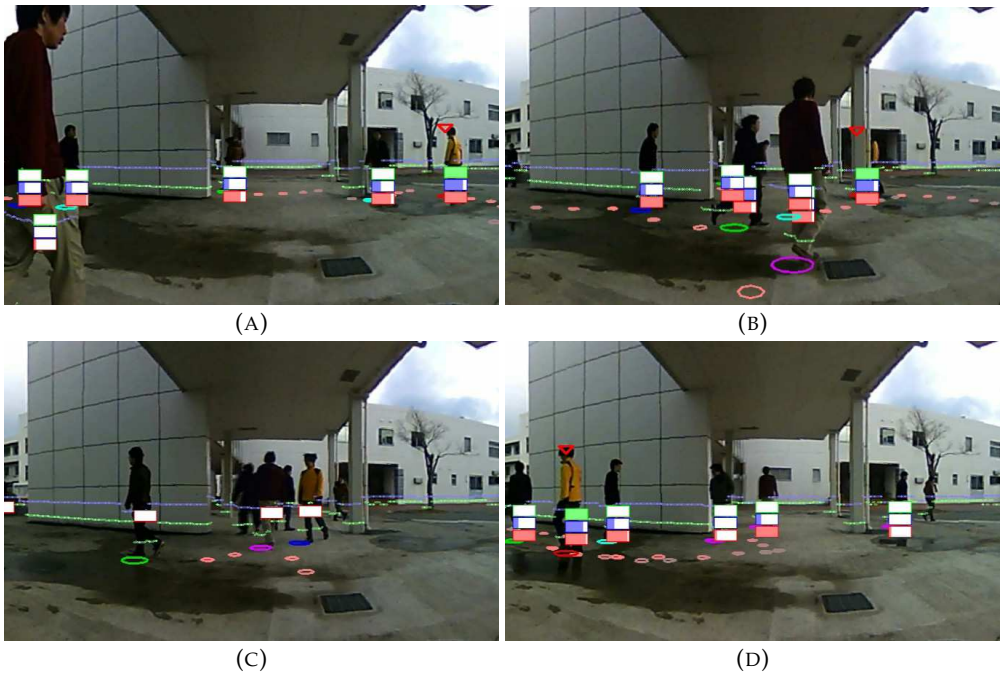
duration [sec]		smartphone		IMU
		chest	trousers	foot
successfully tracked		135.5	128.2	121.8
lost track	target appears	44.6	49.1	56.6
	target not appears	28.3	24.5	27.7
tracked wrong person		0.0	6.6	2.3
average identification time		3.4	3.5	4.7

and we define the identification time as the average of the minimum identification time (n_{sim}) and the maximum identification time (n_{test}):

$$\text{identification time} = T_{step}\frac{n_{sim} + n_{test}}{2}, \tag{3.16}$$

where T_{step} is the cycle time of foot strike. Since the proposed method uses the foot strike timings of both legs and the previous method uses only one leg, we set T_{step} of the proposed method to 0.5 [s] and the previous method to 1.0 [s].

Fig. 3.34 shows the relationship between identification time and identification performance. The identification performance values shown in Fig. 3.34 are the best ones among those with the same identification time. As shown in Fig. 3.34, if we take a longer identification time, we can obtain better performance. Since the proposed method can obtain more foot strike timings than the previous method in a certain period, it shows a better identification performance than the previous method.



(A) (B)

(C) (D)

FIGURE 3.35: Person identification experiment. The person wearing the yellow jacket is the target person; he placed two smartphones, one in his chest pocket and one in his trousers pocket and an IMU on his foot. Circles under persons indicate their positions obtained by LRF-based tracking, red triangles indicate the target person and green, blue, and red bars indicate the posterior probabilities of each person, the likelihood of the foot strike timings, and the likelihood of the stopping state, respectively.

3.3.6 Experimental Results

Person Identification Experiment

We conducted a person identification experiment. In the experiment, the target person held two smartphones (FREETEL FTJ152B, MediaTek accelerometer is embedded), one in his trousers pocket and one in his chest pocket and attached an IMU (ZMP IMU-Z2) to his foot. We used two LRFs (HOKUYO UST-20LX) to track people and estimate their foot strike timings. We apply the proposed method to those smartphones and our previous method to the IMU.

Fig. 3.35 shows snapshots of the experiment with the smartphone in the chest pocket. The person wearing the yellow jacket is the target person. Circles under persons indicate their estimated positions obtained by the LRF-based tracking. Green, blue, and red bars indicate the posterior probabilities that a person is the actual target, the likelihood of the foot strike timings, and the likelihood of the stopping state, respectively. Red triangles on a person indicate that the person is tracked as the target.

The target person and the other persons entered the field and walked around (Fig. 3.35 (a)), and then the target person was lost by the LRF-based tracking due to occlusions (Fig. 3.35 (b)(c)). Once the target person appeared again and walked several steps, however, the target person was re-detected and the tracking was resumed (Fig. 3.35 (d)). During the experiment, the robot lost the track of the target a total of 12 times due to occlusions. The target person was, however, successfully re-detected in every case.

Table 3.9 shows the result of the experiment. The total time of the experiment was 208.4 [sec]. In the case of the smartphone in the trousers pocket, while the

target person is hidden by other persons, wrong persons are re-detected as the target twice. However, the system realized that the person is not the target by the Bayesian inference and soon tracked the target person again. Since we used parameters which are optimized for a smartphone in a chest pocket, the identification performance in the case of the trousers pocket is a little worse than in the case of the chest pocket. If we optimize the parameters for a smartphone in a trousers pocket, we could improve identification performance.

Since the previous method takes longer identification time than the proposed method, its duration of target loss is longer than the others. It shows that the proposed method has improved its responsiveness from the previous method.

During the experiment, the whole procedure except visualization took less than 1 [msec] per frame. Since the processing cost is very low, the method can be extended to track multiple targets by using a smartphone of every target person.

Person Following Experiment

We applied the person identification method to a person following task. While the robot is tracking the target person, it moves toward the person. If the track of the person is lost, the robot moves toward the target person position predicted from the latest observed position and the velocity of the person in order to keep the robot close to the person. We used a path planning method by proposed Ardiyanto and Miura [17] to make the robot avoid obstacles while following the person.

Fig. 3.36 shows snapshots of the person following experiment. During the experiment, the robot lost the target person several times due to occlusion by others (Fig. 3.36 (b)). However, once the target person appeared and walked for several steps, the robot successfully re-detected him among the others and continued to follow him (Fig. 3.36 (c)).

(a)

(b)

(c)

FIGURE 3.36: Specific person following experiment: The left column shows experiment scenes and the right column shows view of the camera on the robot. The meanings of the markers (triangle, circle, bars) are the same as in Fig. 3.35.

Chapter 4

A Portable People Behavior Measurement System using a 3D LIDAR

Measuring and analyzing people behavior are essential to construct human social models, such as awareness estimation models. Here, we propose a system for long-term and wide-area people behavior measurement using a 3D LIDAR. In this chapter, we also provide the result of a field test conducted in Sawarabikai Fukushimura hospital. In this test, we collected the behavior data of professional caregivers attending elderly persons. Through this test, we analyzed how the professional caregivers attend eldely persons. The measured behavior data is useful to make attendant robots socially acceptable by letting them mimic the human behavior.

4.1 Motivation

Several models which describe the social interaction between persons, such as social distance [26] and social force model [25], have been proposed, and a number of works have applied those models to service robots [20, 104, 21]. However, since those models are based on the simple analysis of the distance between persons, they cannot describe the influence of the surrounding environment and the other persons. Such limitations may yield unnatural behavior of the robots in complex situations. To realize a robot with natural and acceptable behavior, it is necessary to measure person behavior in diverse situations and construct a sophisticated interaction behavior model.

There are several datasets which provide people behavior in indoor [105] and outdoor environments [106, 107]. However, to our knowledge, no dataset provides people behavior involving interaction between followed and following persons even though such a situation is very common in daily services. Most of existing robots just keep the distance to the target person constant, and this naive following strategy could make people feel uncomfortable. We believe that it is necessary to measure and analyze people attendant behavior to design the behavior of attendant robots, and it triggered us to develop a system which enables long-term and wide-area people behavior measurement and create a dataset which consists of real professional human's attendant behavior data.

Fig. 4.1 illustrates the proposed system for people behavior measurement. The system is based on a 3D LIDAR, and a human observer carries the system and follows the persons to be observed while keeping them in the sensor view. The system simultaneously estimates the sensor pose in a 3D environmental map and tracks the

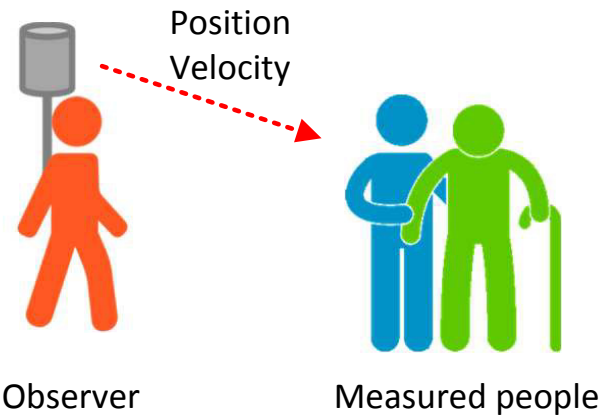


FIGURE 4.1: The proposed system to measure people behavior using a 3D LIDAR. The observer carries the backpack with a 3D LIDAR and follows the persons to be measured.

target persons. The proposed system can be applied to long-term and wide-area people behavior measurement tasks.

4.2 Related Work

Systems to measure people behavior can be categorized into two groups: 1) systems using static sensors which are fixed at the environment, and 2) systems using wearable sensors attached to the target persons.

People tracking using static sensors, such as cameras and laser range finders, have been widely studied. In particular, people tracking using cameras for surveillance is a major research topic in the computer vision community. A lot of works have proposed people detection [43] and tracking methods [108] using RGB cameras. Recent inexpensive consumer RGB-D cameras allow us to reliably detect and track people [28], and a camera network system for people tracking using RGB-D cameras has been proposed [109]. Although such works provide reliable people tracking, a capability of recovering the track of a person, who left the camera view once, is necessary. This problem (i.e., person re-identification) has been one of the main research topics of vision-based people tracking systems. A lot of re-identification methods based on people appearance [93, 14, 23, 110], and soft biometric features [111, 112] have been proposed. They enable reliable people re-identification over time and over cameras.

Laser range finders have also been used for people tracking systems [90, 89]. Such systems can very accurately localize people, and the measurement area of each sensor is larger than cameras. While the reliability and the detection accuracy of those static sensor-based systems are very good, they can measure people behavior only in an area limited by the sensor view. In order to cover a large environment, they require the placement of a lot of static sensors, thereby increasing the time and cost of installing and calibrating all the sensors.

Another way to measure the behavior of specific persons for a long time over a wide area is to attach a wearable sensor to each target person and measure their behavior with the sensor. Several kinds of sensors, such as INS (Inertia Navigation System) and GPS (Global Positioning System), have been used for this purpose. Recent small wearable GPS sensors allow us to track a person in outdoor environments, and they have been applied to several applications of people behavior measurement and analysis [113, 114]. As an application, GPS-based wearable devices for helping

elderly or visually impaired people have been proposed [115, 116]. The combination of GPS and INS improves tracking accuracy under low-level GPS radio power [117]. However, GPS signals are not available in places close to buildings and indoor environments.

Recently, WiFi signal-based localization has been widely studied [118, 119, 120]. Some of them are based on triangulation of WiFi signal strength and show decimeter or centimeter accuracy in ideal situations [118, 119]. However, they require to place multiple antennas in the environment to accurately estimate the device position, and thus, it is hard to be applied to a large environment. Other ones are based on the matching of WiFi fingerprint matching [120]. While they do not rely on external antennas and can be applied to large environments where WiFi signal is available, the estimation accuracy is very limited.

Behavior measurement systems for indoor environments based on pedestrian dead reckoning have also been proposed [102, 121]. Those methods estimate the target person position by integrating acceleration and angular velocity obtained by an INS (attached to the person). In order to prevent estimation drift, Li et al. combined pedestrian dead reckoning with map-based localization [102]. Those methods can keep track the position of the person as long as they hold the sensor. Since they utilize smartphones which are very common and inexpensive in recent years, those methods are cost effective and easy to adopt. However, since INS is an internal sensor and it cannot sense the surrounding environment, it is hard to accurately measure the person position with respect to the environment and other persons positions. Thus, they cannot be applied to the measurement of the interaction between persons and that of person's behavior affected by the environment.

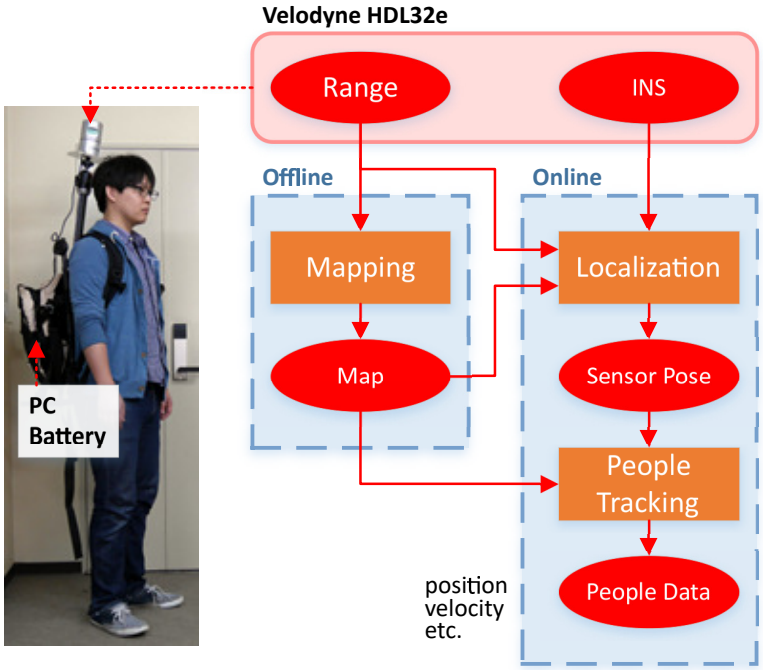


FIGURE 4.2: System overview.

4.3 System Overview

Fig. 4.2 shows an overview of the proposed system. In this system, the observer carries the backpack equipped with a 3D LIDAR (velodyne HDL-32e) and a PC, and follows the persons to be measured. The 3D LIDAR provides 360 degree range data at 10 Hz, and from the range data, the system estimates its pose while tracking the target persons. The process of the proposed system consists of two phases: 1) offline environmental mapping and 2) online sensor localization and people detection/tracking.

In the offline mapping phase, we create a 3D environmental map which covers the entire measurement area. For the mapping, we employ a graph optimization-based SLAM approach (i.e., Graph SLAM [122]). In order to compensate accumulated rotational errors of the scan matching, we introduce ground plane and GPS position constraints for indoor and outdoor environments, respectively.

In the behavior measurement phase, the system estimates its pose on the map created offline by combining a scan matching algorithm with an angular velocity-based pose prediction using Unscented Kalman filter [63]. Simultaneously, the system detects and tracks the target persons.

4.4 Offline Environmental Mapping

4.4.1 Graph SLAM

Graph SLAM is one of the most successful approaches to the SLAM problem. In this approach, the SLAM problem is solved by constructing and optimizing a graph whose nodes represent parameters to be optimized, such as sensor poses and landmark positions, and edges represent constraints, such as relative poses between sensor poses and landmarks. The graph is optimized so that the errors between the parameters and the constraints are minimized. Following [122, 123], let $\boldsymbol{x}_k$ be the node k. Let $\boldsymbol{z}_k$ and Ω_k be the mean and the information matrix of the constraints relating to $\boldsymbol{x}_k$. The objective function is defined as:

$$F(x) = \sum \boldsymbol{e}_k(\boldsymbol{x}_k, \boldsymbol{z}_k)^T \Omega_k \boldsymbol{e}_k(\boldsymbol{x}_k, \boldsymbol{z}_k), \tag{4.1}$$

where, $\boldsymbol{e}_k(\boldsymbol{x}_k, \boldsymbol{z}_k)$ is an error function between the parameters $\boldsymbol{x}_k$ and the constraints $\boldsymbol{z}_k$. Typically, eq. (4.1) is linearized and minimized by using Gauss-Newton or Levenberg-Marquardt algorithms.

However, if the parameters span over non-Euclidean spaces (like pose parameters), those algorithms may lead to sub-optimal or invalid solutions. One way to deal with this problem is to perform the error optimization on a manifold which is a minimal representation of the parameters and acts as an Euclidean space locally. In order to enable it, an operator $\boxplus$ is introduced, which transforms a local variation Δx on the manifold.

Typically, in the 3D SLAM problem, node $\boldsymbol{x}_k$ has parameters of the sensor pose at k (a translation vector $\boldsymbol{t}_k$ and a quaternion $\boldsymbol{q}_k$). A manifold of the quaternion $\boldsymbol{q}_k = [q_w, q_x, q_y, q_z]^T$ can be represented as $[q_x, q_y, q_z]^T$, and the operator $\boxplus$ is described as:

$$\boldsymbol{q}_k \boxplus \Delta\boldsymbol{q} = \left[\sqrt{1 - \|q'_x + q'_y + q'_z\|^2}, q'_x, q'_y, q'_z\right], \tag{4.2}$$

where, $q'_* = q_* - \Delta q_*$.

In the proposed system, we first estimate the sensor trajectory by iteratively applying NDT (Normal Distributions Transform) scan matching [124] between consecutive frames. For 3D LIDARs, NDT shows a better performance than other scan matching algorithms, such as Iterative Closest Points [125], in terms of both the reliability and the processing speed [126]. Let $\boldsymbol{p}_t$ be the sensor pose at t, consisting of a translation vector $\boldsymbol{t}$ and a quaternion $\boldsymbol{q}$, and $\boldsymbol{r}_{t,\,t+1}$ be the relative sensor pose between t and $t+1$ estimated by the scan matching. We add them to the pose graph as nodes $[\boldsymbol{p}_0, \cdots, \boldsymbol{p}_N]$ and edges $[\boldsymbol{r}_{0,\,1}, \cdots, \boldsymbol{r}_{N-1,\,N}]$. Then, we find loops in the trajectory and add them to the graph as edges (i.e., loop closure) to correct the accumulated error of the scan matching with Algorithm 1.

Algorithm 1 Loop-detection

Input: $\mathcal{P} = \{\boldsymbol{p}_0, \ldots, \boldsymbol{p}_N\}$, pose nodes
Input: $\mathcal{R} = \{\boldsymbol{r}_{0,1}, \ldots, \boldsymbol{r}_{N-1,N}\}$, odometry edges
Output: $\mathcal{L} = \{\boldsymbol{l}_0, \ldots, \boldsymbol{l}_M\}$, loop edges

1: $\mathcal{L} \Leftarrow \{\}$
2: **for** $i = 0 \ldots N-1$ **do**
3: $\quad\mathcal{C} \Leftarrow \{\}$ ▷ Loop candidates
4: $\quad accum\_d \Leftarrow 0$ ▷ Accumulated distance
5: $\quad$**for** $j = i+1 \ldots N$ **do**
6: $\quad\quad d \Leftarrow \|\boldsymbol{p}_i.\boldsymbol{t} - \boldsymbol{p}_j.\boldsymbol{t}\|$
7: $\quad\quad accum\_d \Leftarrow accum\_d + d$
8: $\quad\quad$**if** $d < th_d$ and $accum\_d > th_a$ **then**
9: $\quad\quad\quad$Add loop candidate $l = \{\boldsymbol{p}_i, \boldsymbol{p}_j\}$ to $\mathcal{C}$
10: $\quad\quad$**end if**
11: $\quad$**end for**
12: $\quad$**for** $l = \{\boldsymbol{p}_i, \boldsymbol{p}_j\}$ in $\mathcal{C}$ **do**
13: $\quad\quad m \Leftarrow scan\_matching(p_i, p_j)$
14: $\quad\quad$**if** $m.score < th_s$ **then**
15: $\quad\quad\quad\mathcal{L} \Leftarrow \mathcal{L} \cup \{l\}$
16: $\quad\quad$**end if**
17: $\quad$**end for**
18: **end for**

The loop detection algorithm is similar to [127]. First, we detect loop candidates based on the translational distance and the length of the trajectory between nodes (Line 2 $\sim$ 11). Then, to validate the loop candidates, a scan matching algorithm (in our case, NDT) is applied between the nodes of each candidate. If the fitness score is lower than a threshold (e.g., 0.2), we add the loop to the graph as an edge between the nodes (Line 12 $\sim$ 17). Every time a loop is found, the pose graph is updated such that eq. (4.1) is minimized. We utilize g2o, a general framework for hypergraph optimization [123], for the pose graph optimization.

As a generated map gets larger, it tends to be bent due to the accumulated rotational error of the scan matching (see Fig. 4.7). In order to compensate the error, we introduce ground plane and GPS position constraints for indoor and outdoor environments, respectively. Fig. 4.3 shows an illustration of the graph structure of the proposed system.

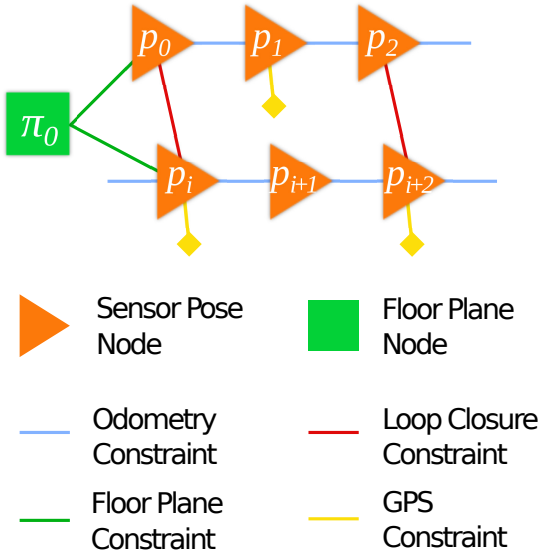


FIGURE 4.3: The proposed pose graph structure.

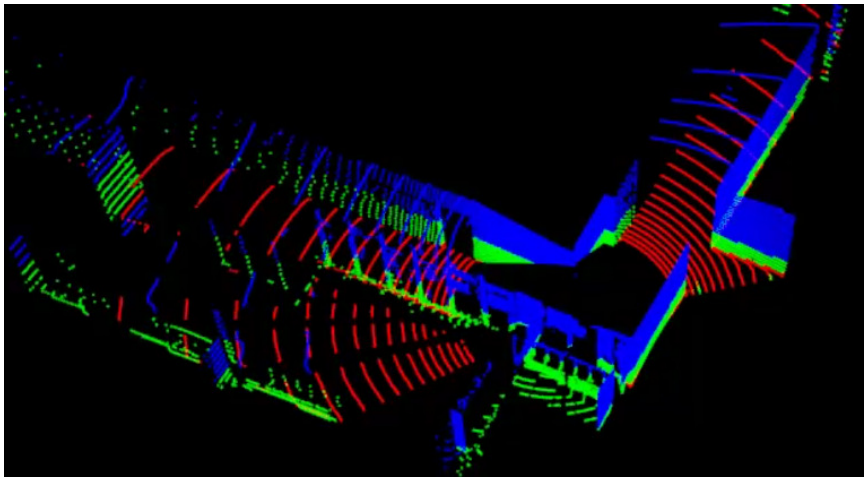


FIGURE 4.4: Ground plane detection. Points within a certain height range are extracted by height thresholding (green points), and then RANSAC is applied to them to detect the ground plane (red points). The horizontality of the ground plane is validated by checking the plane normal.

4.4.2 Ground Plane Constraint

To reliably generate the map of a large indoor environment, we assume that the environment has a single flat floor, and introduce the ground plane constraint which optimizes the pose graph such that the ground plane detected in each observation becomes the same plane. This assumption is valid in many indoor public environments, such as schools and hospitals.

We assume that the approximate height of the sensor is known (e.g., 2m) and extract points within a certain height range which should contain the ground plane points (e.g., [-1.0, +1.0]m from the ground). Then, we apply RANSAC [128] to the extracted point cloud and detect the ground plane. If the normal of the detected plane is almost vertical (the angle between the normal and the unit vertical vector is lower than 10 deg), we consider that the ground plane is correctly detected and add a ground plane constraint edge to the graph. Fig. 4.4 shows an example of

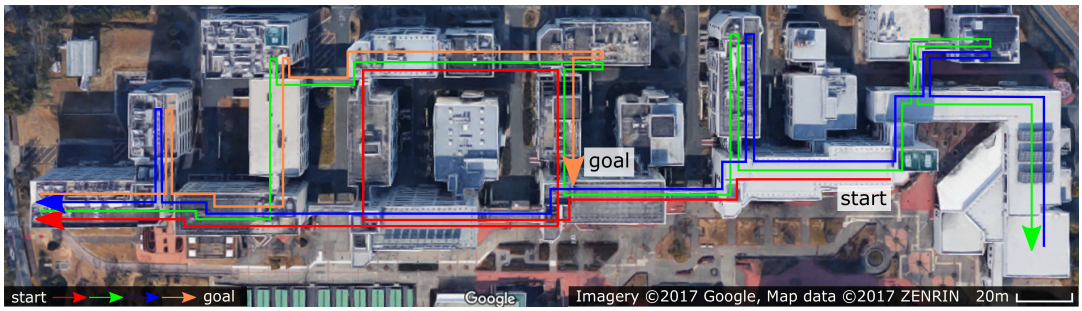


FIGURE 4.5: The experimental environment. The duration of the sequence is about 45 minutes, and the length of the trajectory is about 2400 m.

the detected ground planes. Green points are the points extracted by the height thresholding, and red points belong to the ground plane detected by RANSAC. We detect the ground plane every 10 seconds and connect the corresponding sensor pose node $\boldsymbol{p}_i$ with the fixed ground plane node where the plane coefficients $\boldsymbol{\pi}_0 = [n_x, n_y, n_z, d]^T = [0, 0, 1, 0]^T$.

To calculate the error between sensor pose $\boldsymbol{p}_t$ and the ground plane $\boldsymbol{\pi}_0$, we first transform the ground plane into the local coordinate of the sensor pose $\boldsymbol{p}_t$:

$$[n'_x, n'_y, n'_z]^T = \boldsymbol{R}_t \cdot [n_x, n_y, n_z]^T, \tag{4.3}$$

$$d' = d - \boldsymbol{t}_t \cdot [n'_x, n'_y, n'_z]^T, \tag{4.4}$$

where, $\boldsymbol{\pi}'_0 = [n'_x, n'_y, n'_z, d']$ is the ground plane in the local coordinate, and $[\boldsymbol{R}_t|\boldsymbol{t}_t]$ is the sensor pose at time t.

Following Ma's work [129], we employ the minimum parameterization $\tau(\boldsymbol{\pi}) = (\phi, \psi, d)$, where ϕ, ψ, d are the azimuth angle, the elevation angle, and the length of the intercept, respectively. The error between a pose node and the ground plane node is defined as:

$$\tau(\boldsymbol{\pi}) = \left[\arctan\left(\frac{n_y}{n_x}\right), \arctan\left(\frac{n_z}{|\boldsymbol{n}|}\right), d\right], \tag{4.5}$$

$$\boldsymbol{e}_{i,0} = \tau(\boldsymbol{\pi}'_0) - \tau(\boldsymbol{\pi}_t), \tag{4.6}$$

where $\boldsymbol{\pi}_t$ is the detected ground plane at t.

4.4.3 GPS Constraint

In outdoor environments where the ground is not flat, we use the GPS-based position constraint instead of the ground plane constraint. For ease of optimization, we first transform GPS data into the UTM (Universal Transverse Mercator) coordinate, where a GPS data has easting, northing, and altitude values in a Cartesian coordinate. Then, each GPS data is associated with the pose node, which has the closest timestamp to the GPS data, as an unary edge of the prior position information.

The error between the translation vector $\boldsymbol{t}_t$ of a pose node $\boldsymbol{p}_t$ and a GPS position $\boldsymbol{T}_t$ is simply given by:

$$\boldsymbol{e}_i = \boldsymbol{t}_t - \boldsymbol{T}_t. \tag{4.7}$$

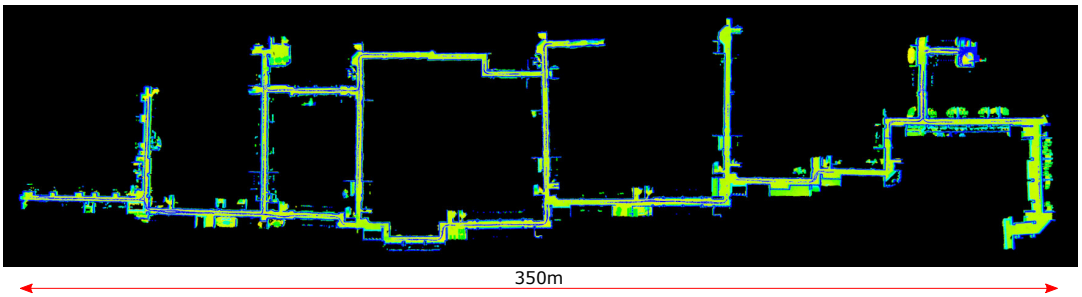


FIGURE 4.6: The created environmental map. The color indicates the height of each point. The height of the floor is consistent thanks to the plane constraint.

4.4.4 SLAM Framework Evaluation

In order to validate the proposed SLAM system, we recorded a 3D point cloud sequence in an indoor environment. Fig. 4.5 shows the experimental environment and the trajectory of the sequence. The duration of the sequence is about 45 minutes (2700 sec), and the length of the trajectory is about 2400 m (estimated by the proposed method).

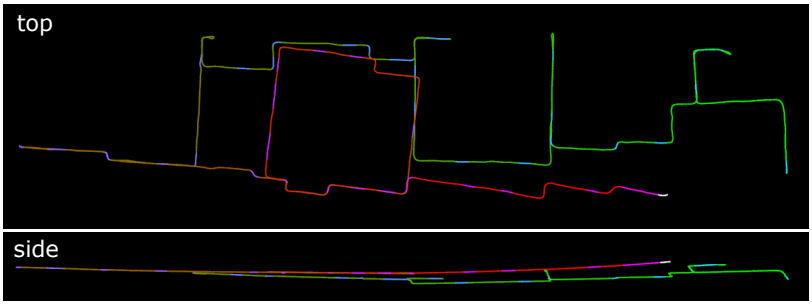
For comparison, we generated 3D environmental maps using the proposed method with and without plane constraints. We also applied existing publicly available SLAM frameworks, BLAM [127] and LeGO-LOAM [130], to this dataset.

Fig. 4.7 shows the trajectories estimated by the different SLAM algorithms. BLAM and LeGO-LOAM were aborted in the middle of the sequence when they failed to estimate the trajectory and did not recover. BLAM failed to find the loops due to the accumulated rotation error of the scan matching, and generated a warped and inaccurate trajectory. Since LeGO-LOAM maintains the local consistence of the ground plane between consecutive frames, the estimated trajectory is flatter than the one estimated by BLAM. However, it still suffer from the accumulated rotational error due to the lack of the global ground constraint. Eventually, it failed to estimate the trajectory when the observer made a u-turn at the end of a narrow corridor.

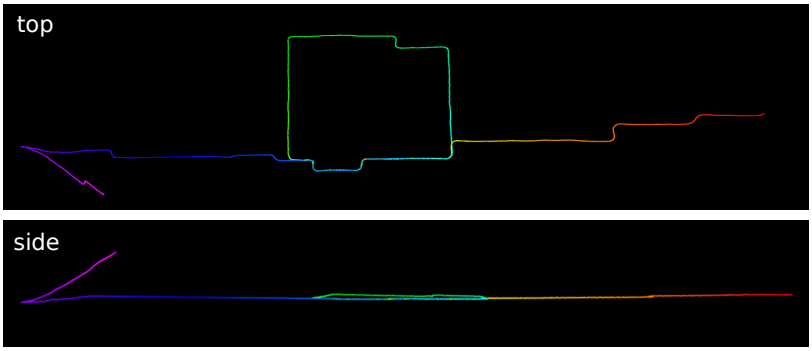
With and without the plane constraint, the proposed method could construct pose graphs properly thanks to the reliability of NDT, and it generated consistent maps. However, without the plane constraint, the resultant map is warped due to the accumulated rotational error which is hard to be corrected by loops on a plane. With the ground plane constraint, the accumulated rotational error is corrected, and the resultant map is completely flat. Fig. 4.6 shows the generated environmental map. The color indicates the height of each point. The floor has the consistent height thanks to the plane constraint. The result shows that the proposed plane constraint is effective to compensate the accumulated rotational error in a large indoor environment.

Table 4.1 shows the processing time of the proposed method and BLAM. The processing time of LeGO-LOAM is not available here, since it provides only real-time processing. While BLAM took about 15,327 [sec] to generate the map, the proposed method took about 5,392 [sec] thanks to the computational efficiency of NDT.

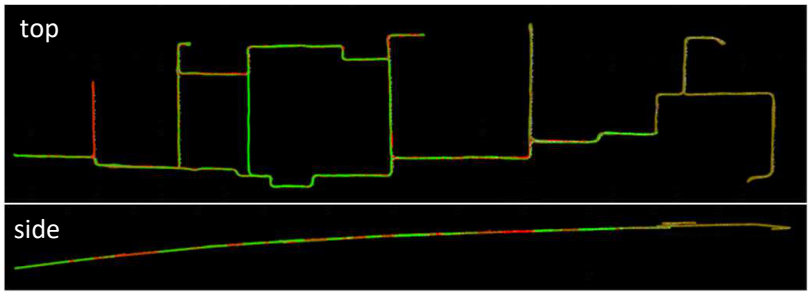
We also validated the proposed method in an outdoor environment. Fig. 4.8 (a) shows the environment and the trajectory of the sequence. The duration of the sequence is about 42 minutes (2500 sec). Fig. 4.8 (b) shows the map generated by the proposed method with the GPS constraint. Although there were large undulations, the system correctly found loops and constructed a proper pose graph thanks to the



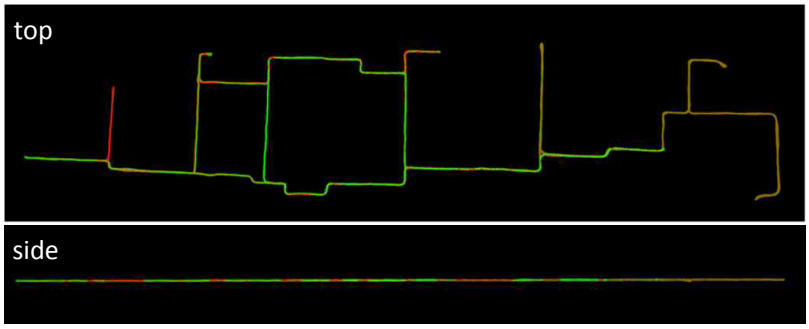
(A) BLAM



(B) LeGO-LOAM



(C) ours w/o plane constraints



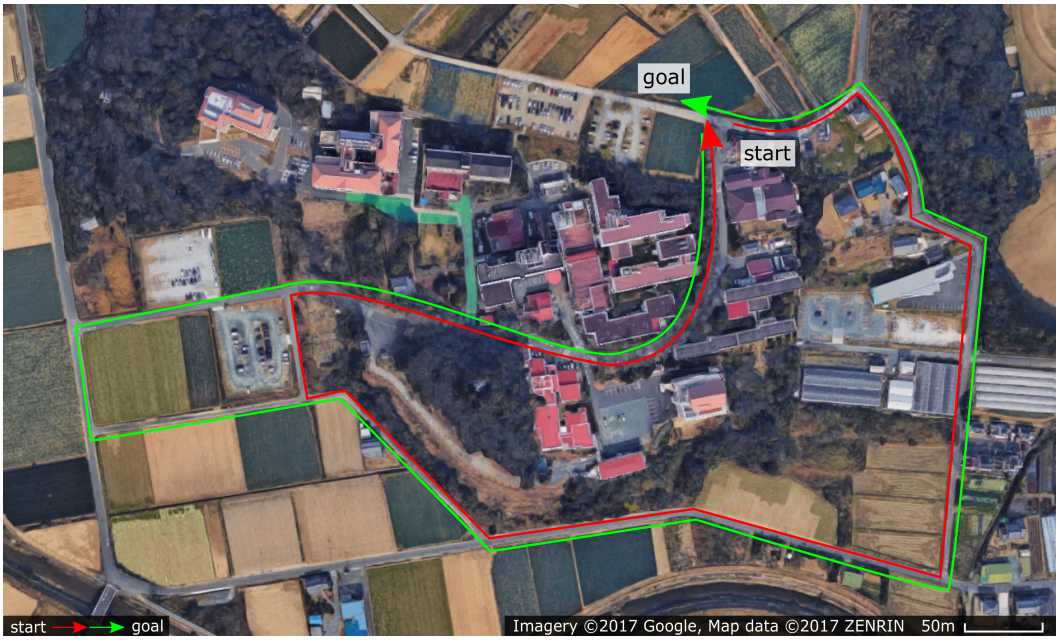
(D) ours w/ plane constraints

FIGURE 4.7: Comparison of the sensor trajectories estimated by the existing method and the proposed method.

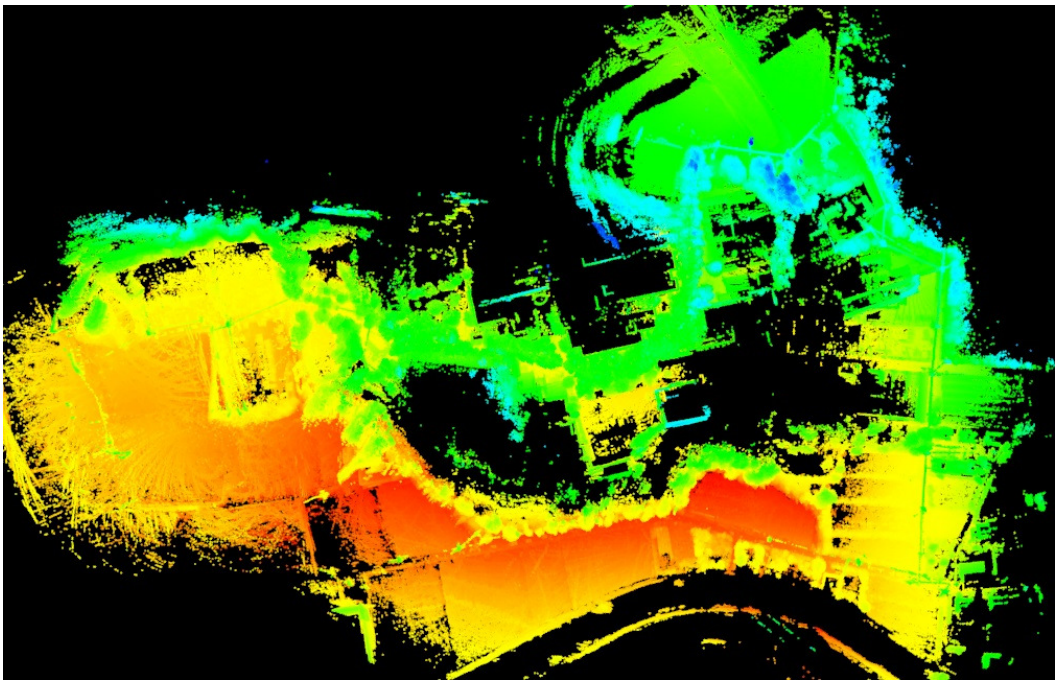
GPS constraint. Note that, without the GPS constraint, the system could not find the loop due to the scan matching error and failed to create the environmental map.

4.5 Online People Behavior Measurement

In order to measure people behavior, the system simultaneously estimates the sensor pose on the 3D environmental map and tracks people around the observer. Fig. 4.9 shows an overview of the online sensor localization and people tracking system.



(A) The outdoor environment. The duration of the sequence is about 42 minutes, and the length of the trajectory is about 3000 m.



(B) The 3D map of the outdoor environment generated by the proposed method with GPS constraints. The color indicates the height of each point.

FIGURE 4.8: The SLAM system validation in an outdoor environment.

By integrating angular velocity and range data provided by the LIDAR, the system estimates the sensor pose. Then, it detects and tracks people to know people positions with respect to the environmental map. Note that the initial pose of the sensor is given by hand to avoid the global localization problem.

TABLE 4.1: Processing time of BLAM and our SLAM system.

method		time [sec]
ours	scan matching	1542
	floor detection	231
	loop closing	3619
	total	5382
BLAM	total	15327

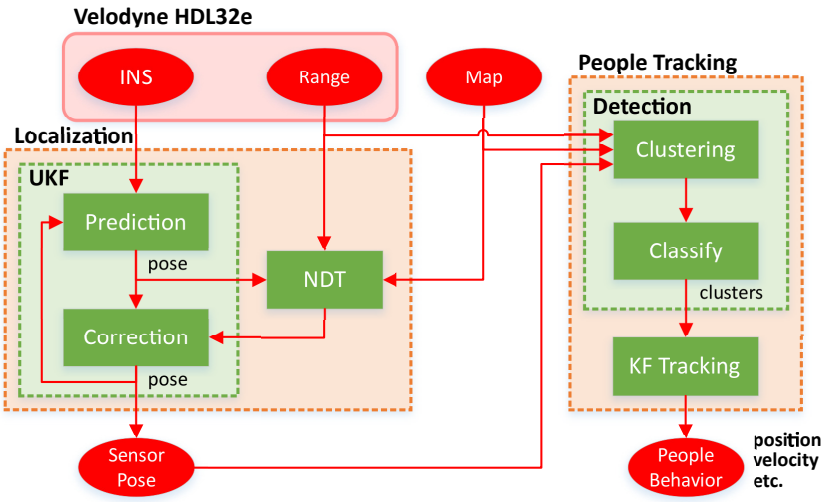


FIGURE 4.9: The online sensor pose estimation and people detection and tracking system.

4.5.1 Sensor Localization

We can estimate the sensor ego-motion by iteratively applying a scan matching algorithm as in the SLAM part. However, in contrast to the SLAM scenario, the observer has to follow the target persons during the measurement and sometimes has to move quickly to keep them in the sensor view. In such cases, the sensor motion between frames gets very large and the scan matching may wrongly estimate the sensor ego-motion due to the large displacement. In order to deal with this problem, we integrate the NDT scan matching with angular velocity data provided by the 3D LIDAR using Unscented Kalman filter [63].

We define the sensor state to be estimated as:

$$\boldsymbol{x}_t = [\boldsymbol{p}_t, \boldsymbol{q}_t, \boldsymbol{v}_t, \boldsymbol{b}_t^a]^T, \tag{4.8}$$

where, $\boldsymbol{p}_t$ is the position, $\boldsymbol{q}_t$ is the rotation quaternion, $\boldsymbol{v}_t$ is the velocity, $\boldsymbol{b}_t^a$ is the bias of the angular velocity of the sensor at time t. Assuming constant translational velocity for the sensor motion model, and constant bias for the angular velocity sensor, the system equation for predicting the state is defined as:

$$\boldsymbol{x}_t = [\boldsymbol{p}_{t-1} + \Delta t \cdot \boldsymbol{v}_{t-1}, q_{t-1} \cdot \Delta \boldsymbol{q}_t, \boldsymbol{v}_{t-1}, \boldsymbol{b}_{t-1}^a]^T, \tag{4.9}$$

where, Δt is the duration between t and $t-1$, $\Delta \boldsymbol{q}_t$ is the rotation during Δt caused by the bias-compensated angular velocity $\boldsymbol{a}_t' = \boldsymbol{a}_t - \boldsymbol{b}_{t-1}^a$:

$$\Delta \boldsymbol{q}_t = \left[1, \frac{\Delta t}{2} a_t^{x\prime}, \frac{\Delta t}{2} a_t^{y\prime}, \frac{\Delta t}{2} a_t^{z\prime}\right]^T. \tag{4.10}$$

With eq. (4.9), the system predicts the sensor pose by using Unscented Kalman filter, and then applies NDT to match the observed point cloud with the global map with the estimated $\boldsymbol{x}_t$ and $\boldsymbol{q}_t$ as the initial guess of the sensor pose. Then, the system corrects the sensor state with the sensor pose estimated by the scan matching $\boldsymbol{z}_t = [\boldsymbol{p}'_t, \boldsymbol{q}'_t]^T$. The observation equation is defined as:

$$\boldsymbol{z}_t = [\boldsymbol{p}_t, \boldsymbol{q}_t]^T. \tag{4.11}$$

We normalize the quaternion in the state vector after each of the prediction and correction steps to prevent its norm from changing due to the unscented transform and the accumulated calculation error. It is worth mentioning that we also implemented pose prediction which takes acceleration into account. However, the estimation result got worse due to the strong noise on acceleration observations.

4.5.2 People Detection and Tracking

We first remove the background points from an observed point cloud to extract the foreground points. Then, we create an occupancy grid map with a certain voxel size (e.g., 0.5m) from the environmental map. The input point cloud is transformed into the map coordinate according to the sensor pose estimated by UKF, and then each point at a voxel containing environmental map points is removed as the background. The Euclidean clustering is then applied to the foreground points to detect human candidate clusters. However, in case persons are close together, their clusters may be wrongly merged and are detected as a single cluster. To deal with this problem, we employ Haselich's split-merge clustering algorithm [131].

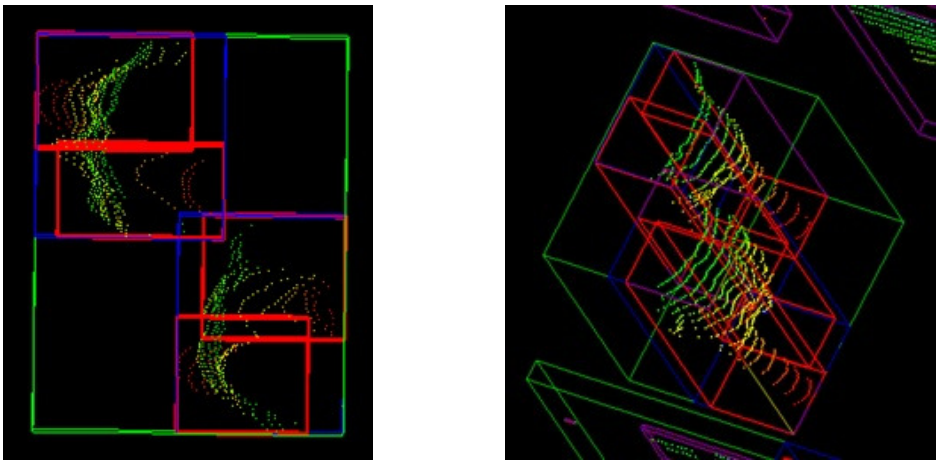
The algorithm first divides a cluster into sub-clusters until each cluster gets smaller than a threshold (e.g., 0.45m) by using dp-means [132] so that every cluster does not have points of different persons. Then, if there is no gap between those sub-clusters, the clusters are considered to belong to a single person and re-merged into one cluster. Fig. 4.10 shows an example of the detection results. The person clusters are correctly separated even when they are very close together thanks to the split and the re-merge process.

The detected clusters may contain non-human clusters (i.e., false positives). To eliminate non-human clusters among detected clusters, we judge whether a cluster is a human or not by using a human classifier trained with slice features by Kidono et al. [30] and Adaboost [67]. Assuming that persons walk on the ground plane, we track persons on the XY plane without the height. We employ the combination of Kalman filter with the constant velocity model and global nearest neighbor data association [35] to track persons. The tracking scheme works well as long as the tracked persons are visible from the sensor and are correctly detected.

4.5.3 Sensor Localization Evaluation

To show how the pose prediction improves the sensor localization, we conducted a sensor localization experiment. Fig. 4.11 shows the experimental environment. An observer carries the system and moves along the corridor, and the system estimates its pose from the range and angular velocity data. We conducted the experiment twice. In the first trial, the observer walked (about 1.5 m/sec) to avoid the sensor being moved quickly. In the second trial, the observer ran (about 3.0 m/sec) and the sensor got shaken very strongly.

Fig. 4.12 shows the results of the first trial. Fig. 4.12 (a) shows the estimated trajectories with and without the pose prediction. Since the observer moved slowly



(A) Top view. (B) Bird's eye view.

FIGURE 4.10: Haselich's clustering algorithm. The green bounding box indicates the Euclidean clustering result. Two persons are wrongly detected as a single cluster. The cluster is divided into small sub-clusters (red bounding boxes) and then re-merged if there is no gap between those sub-clusters. The blue bounding boxes are the final detection result.

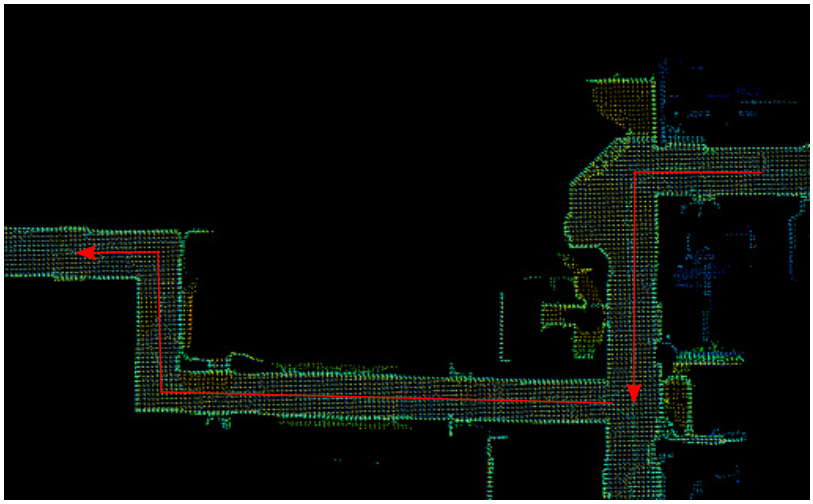
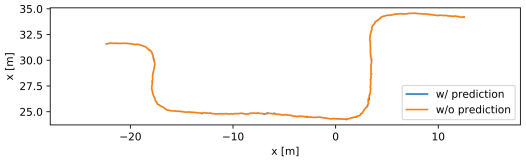


FIGURE 4.11: The experimental environment of the sensor localization experiment.

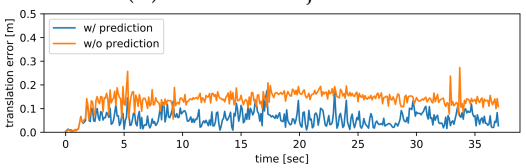
TABLE 4.2: The summary of the sensor localization experiment.

seq.	w/ prediction			w/o prediction		
	error[m]	error[deg]	time[msec]	error[m]	error[deg]	time[msec]
1st (walk)	0.0588	1.0913	38.88	0.1367	2.1625	40.06
2nd (run)	0.1851	4.2845	45.14	0.3330	6.6798	56.11

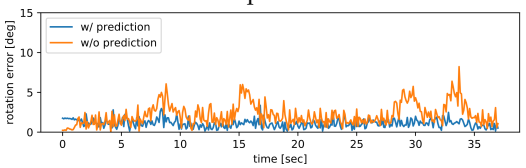
during the first sequence, both the results show the same correct trajectory. To assess the effect of the sensor pose prediction, we assume that the trajectories estimated by NDT are mostly correct, and we compare the predicted sensor poses with the poses estimated by NDT since measuring the ground truth of the sensor trajectory is difficult. Fig. 4.12 (b) and (c) show the difference between the predicted sensor pose (initial guess pose) and the one estimated by NDT. In the case without the pose



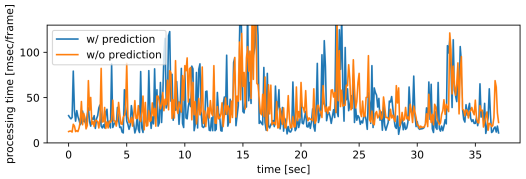
(A) Estimated trajectories.



(B) Difference between the predicted and the corrected positions.

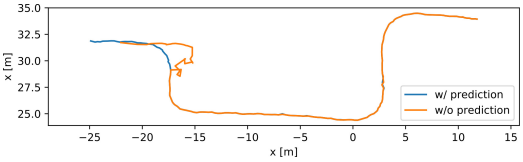


(C) Difference between the predicted and the corrected rotations.

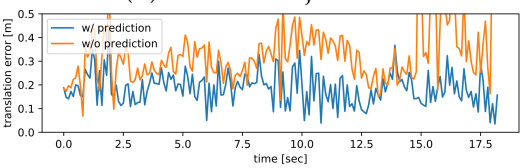


(D) Processing time.

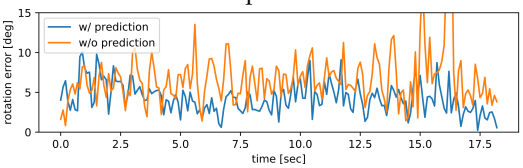
FIGURE 4.12: The results of the first trial of the sensor localization experiment. The observer walked during the trial (about 1.5 m/sec). Both the trajectories with and without the angular velocity-based pose prediction are correctly estimated. With the prediction, the initial guess for NDT significantly gets closer to the correct pose.



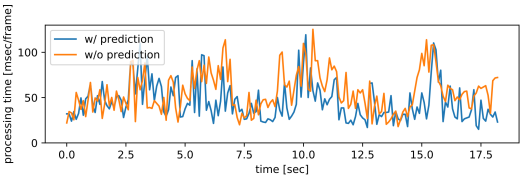
(A) Estimated trajectories.



(B) Difference between the predicted and the corrected positions.



(C) Difference between the predicted and the corrected rotations.



(D) Processing time.

FIGURE 4.13: The results of the second trial of the sensor localization experiment. The observer ran during the trial (about 3.0 m/sec). Without the pose prediction, the system could not correctly estimate the pose due to the very quick motion.

prediction, the previous matching result is used as an initial guess. With the prediction, the translational and rotational pose prediction errors significantly decrease thanks to the constant velocity model and the consideration of angular velocity, respectively.

The results of the second trial are shown in Fig. 4.13. The system failed to estimate the sensor pose without the pose prediction (see. Fig. 4.13 (a)) since the observer moved very quickly, and the sensor displacement between frames got larger. The NDT matching took a longer time (about 56 msec per frame) without the pose prediction since the large displacement between frames makes NDT need more iterations to converge to a local solution. With the prediction, the matching took about 45 msec per frame thanks to the good initial guess (see Table 4.2). The results show that the angular velocity-based pose prediction makes the pose estimation robust to quick motions and fast to converge.

4.5.4 People Detection Evaluation

To analyze the effect of the split-merge clustering [131] and the human classifier [30], we recorded a 3D range data sequence, in which two persons are close together and

TABLE 4.3: The people detection evaluation result.

Split-merge Clustering [131]	Human Classifier [30]	precision	recall	F-measure
w/o	w/o	1.000	0.834	0.909
w/o	w/	1.000	0.809	0.894
w/	w/o	0.902	0.995	0.946
w/	w/	0.961	0.961	0.961

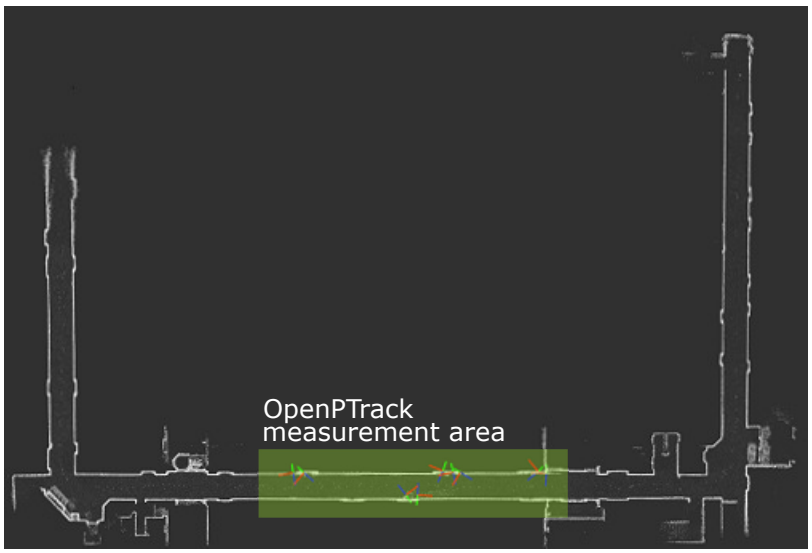


FIGURE 4.14: The experimental environment and the configuration of RGB-D cameras for OpenPTrack. Nine Kinect v2's are placed in the corridor. While OpenPTrack can measure only the limited area covered by cameras (about 2m $\times$ 20m area), the proposed system can cover the whole of the floor.

walking side by side. It is a hard situation for the usual Euclidean clustering since the persons' clusters may be merged into a single cluster. The number of frames is 102, and we applied the human detection method with and without the split-merge clustering and the human classifier to this sequence.

Table 4.3 shows the evaluation result. Without both the techniques, the recall value is low (0.834), since clusters of the persons are sometimes detected as a single cluster due to the Euclidean clustering. With the split-merge clustering, the wrongly merged clusters are split into sub-clusters, and the recall value gets higher (0.995). With both the split-merge clustering and the human classifier, over split sub-clusters are eliminated by the classifier, and the highest F-measure value is achieved (0.961). This result shows that, in situations where persons are close together, the split-merge clustering [131] effectively increases the recall of human detection, and by combining it with the human classifier [30], we can obtain reliable human detection results.

4.5.5 Comparison with a Static Sensor-based People Tracking System

In order to reveal the pros and cons of the proposed system, we compared the proposed system with a publicly available static sensor-based people tracking framework, OpenPTrack [109]. The framework is designed for people tracking using static RGB-D cameras, and it is scalable to a large camera network. Moreover, it uses cost effective hardware and is easy to setup. It has been operated by people including non-experts in computer vision, such as artists and psychologists.

TABLE 4.4: The difference of the observer and the subject positions measured by the proposed system and OpenPTrack.

	difference [m]			
	min	max	mean	std. dev.
observer	0.0008	0.2126	0.0768	0.0448
subject	0.0035	0.2837	0.0990	0.0445

Fig. 4.14 shows the experimental environment and the configuration of the RGB-D camera network. The map is created by the proposed SLAM method. We placed nine Kinect v2's so that they cover about 2m $\times$ 20m area. We calibrated the camera network according to the procedure provided by OpenPTrack and then estimated the transformation between the environmental map and the camera network by performing ICP registration between point clouds of the Kinects and the environmental map.

While a subject walked in the corridor, an observer carrying the proposed system followed him. The trajectories of both the persons were measured by the proposed system and OpenPTrack. Table 4.4 shows the summary of the differences between the people positions measured by the proposed system and OpenPTrack. The differences sometimes became larger (about 0.2 $\sim$ 0.3m) due to detection errors of OpenPTrack at the border of the camera view. However, the difference is lower than 0.1m on average, and the result shows that the measurement accuracy of the proposed system and the static sensor-based people tracking system are comparable.

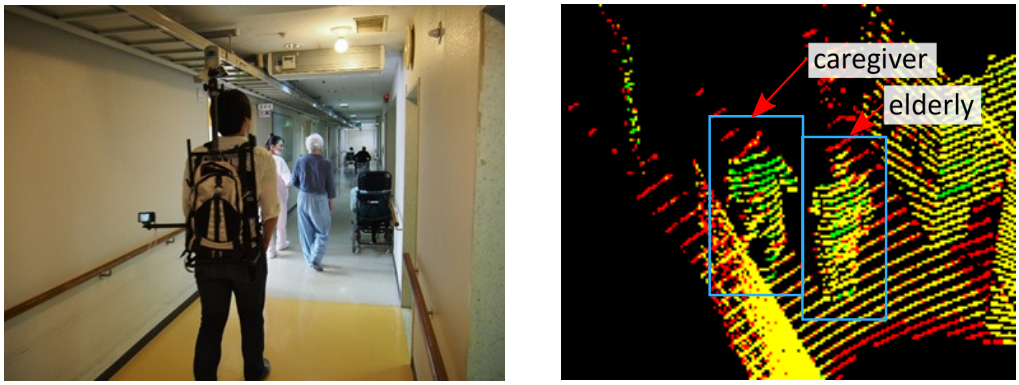
In summary, the tracking accuracy of the proposed portable system is comparable to the static sensor-based system, and the measurement area of the proposed system can be extended easily. For instance, the system can measure the people behavior over the whole area of the map shown in Fig. 4.6 (200 m $\times$ 50 m). We would need hundreds of cameras to cover the whole area of the map if we used a static sensor-based system in the environment. On the other hand, static sensor-based systems can measure behavior of all people in the covered area simultaneously while the proposed system covers only the surrounding area. Thus, we can say that the proposed system is suitable to measure the behavior of specific people over a large area, while static sensor-based systems are suitable for behavior measurement of all the people in a relatively small environment.

4.6 Field Test in a Hospital

4.6.1 Measuring Behavior of Caregivers Attending Elderly Persons

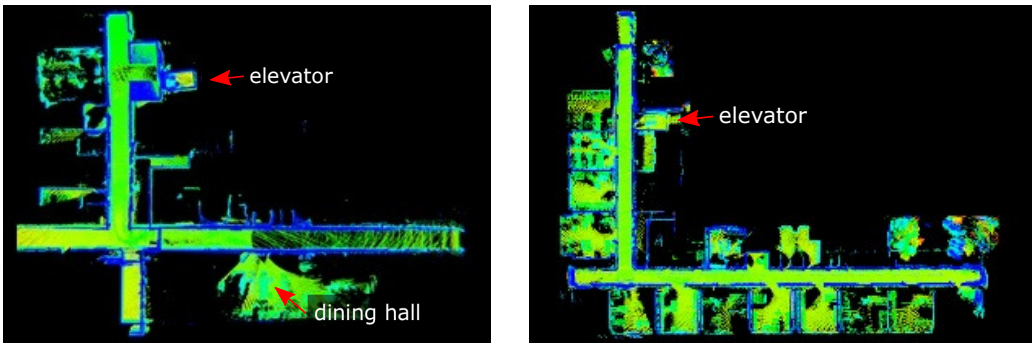
To show that the proposed system can be applied to real people behavior measurements, we conducted a field test in Sawarabikai Fukushimura hospital. The hospital is specialized for elderly care, and hundreds of elderly patients are hospitalized and receiving care and rehabilitation in the hospital. Under permission granted by the hospital, we recorded professional caregivers' behavior while they attend elderly persons with dementia. Fig. 4.15 shows a snapshot of the field test. The caregiver attends the elderly to prevent accidents (such as stumbling, colliding, and falling) and sometimes guides him/her to their room.

The number of sequences is 33, and the total duration is about 52 minutes. We also recorded an attendant behavior sequence in an outdoor environment shown in Fig. 4.8. The duration of the outdoor sequence is about 22 minutes. Note that, for privacy reasons, we captured images during only the sequence shown in Fig. 4.15



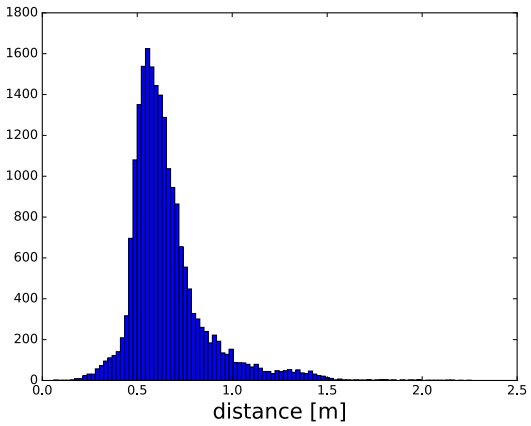
(A) Image. (B) Range data.

FIGURE 4.15: A snapshot of the field test. The behavior of the care giver attending an elderly is recorded by using the proposed system.

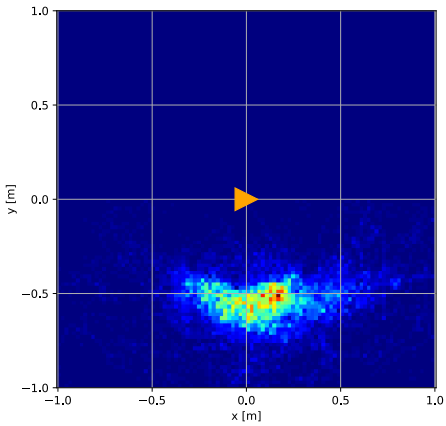


(A) Hallway (1F). (B) Ward (2F).

FIGURE 4.16: The environments of the field test.



(A) The distribution of the distance between the elderly person and the care giver.

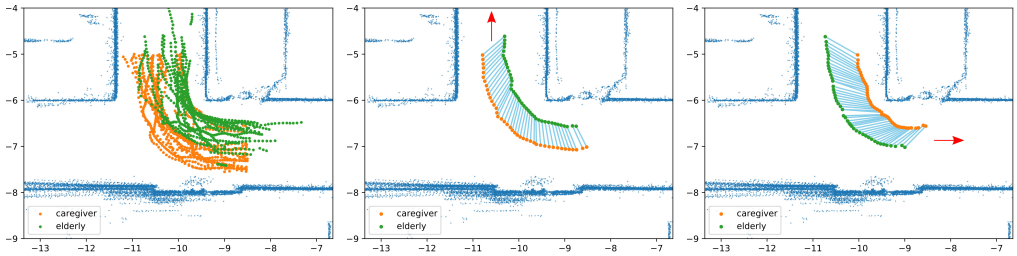


(B) The distribution of the relative position of the care giver with respect to the elderly person.

FIGURE 4.17: An analysis of the people attending behavior during the field test in an indoor environment.

with the special permission from the hospital, the subject, and his family. In the other sequences, we recorded only range data. It is a merit of the proposed system that it can measure people behavior without privacy problems.

Fig. 4.16 shows the created indoor environmental maps through the field test. The elderly persons take rest at the dining hall on the first floor and then return to their hospital room on the second floor with a caregiver using the elevator. After



(A) All the trajectories of the caregivers and the elderly persons.

(B) An example of the cases where the caregiver walks on the outer side of the corner.

(C) The case where the caregiver walks on the inner side of the corner.

FIGURE 4.18: The trajectories of the caregivers (in orange) and the elderly persons (in green) at a corner. The light blue lines indicate that the connected points are measured at the same time. In most of the cases, the caregivers walked on the outer side of the corner (15 of 17). In a few cases, the caregivers walked on the inner side. In such cases, they preceded the elderly persons to ensure outlook of the corridor (2 of 17).

they ride the elevator, we switch the map from the one of the first floor to the second floor.

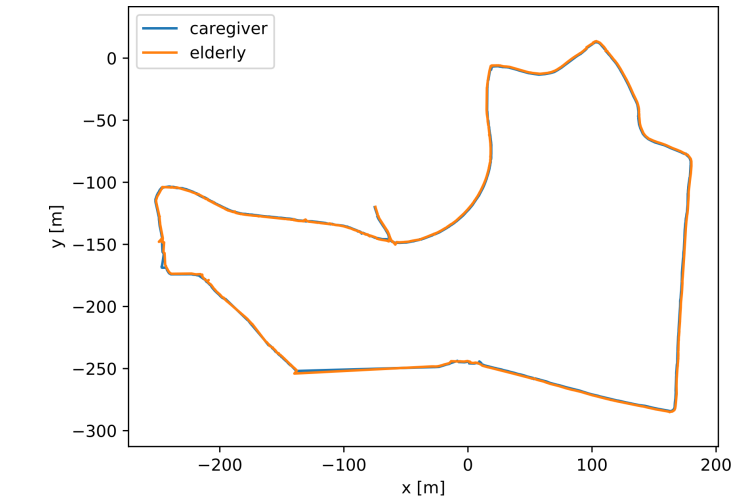
During the measurement, there were other patients and objects, such as wheelchairs and medicine racks, and the observer sometimes had to move quickly to keep the subjects in sensor view. However, the proposed system could correctly localize itself through all the sequences thanks to the wide measurement area of the 3D LIDAR and the integration of the scan matching and the angular velocity-based pose prediction.

Regarding people tracking, the system failed to keep track of the subjects when a patient came between the observer and the subjects to be observed, and new IDs were assigned to the subjects after they re-appeared. In such cases, the system notifies that it lost the track of subjects, and we re-assigned correct IDs to them by hand. Since we saw those cases only a few times, the system could keep track of the subjects for the most part of the sequences, and we could re-assign all the IDs with the minimum effort.

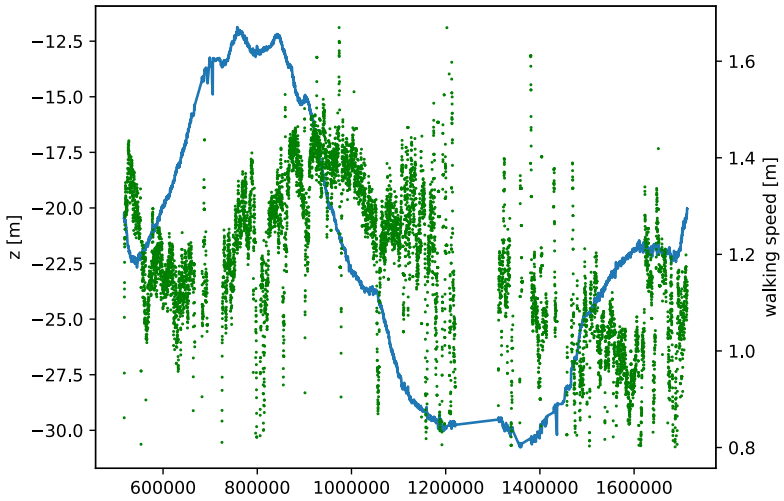
4.6.2 Preliminary Analysis of the Attendant Behavior

To show the possibility of the behavior analysis with the proposed system, we provide preliminary analysis of the measured behavior sequences.

Fig. 4.17 (a) shows the distribution of the distance between a caregiver and an elderly person in the indoor environment. The distribution is unimodal, and the peak is at about 0.6m. In proxemics, this distance is categorized as "*Personal distance* (0.45m - 1.2m)", and people allow only familiar people to be within this distance [26] while they keep more distance (i.e., "*Social distance* (1.2m - 3.6m)") when meeting or interacting with unfamiliar people. It implies that people maintain a closer relationship while attending another person comparing to usual people interaction, such as meeting. Fig. 4.17 (b) shows the distribution of the caregivers' position with respect to the elderly persons. The caregivers usually locate at the side of the elderly persons. In order to lead the elderly persons, they slightly precede the patients. The distribution is a bit anisotropic: when a caregiver is following an elderly person, the distance between them tends to be larger since the caregivers see the elderly person and the surrounding environment at the same time. From this preliminary analysis we can find that the caregivers decide their attending position in order to keep the elderly person in the view and look ahead in the environment.



(A) People trajectory.

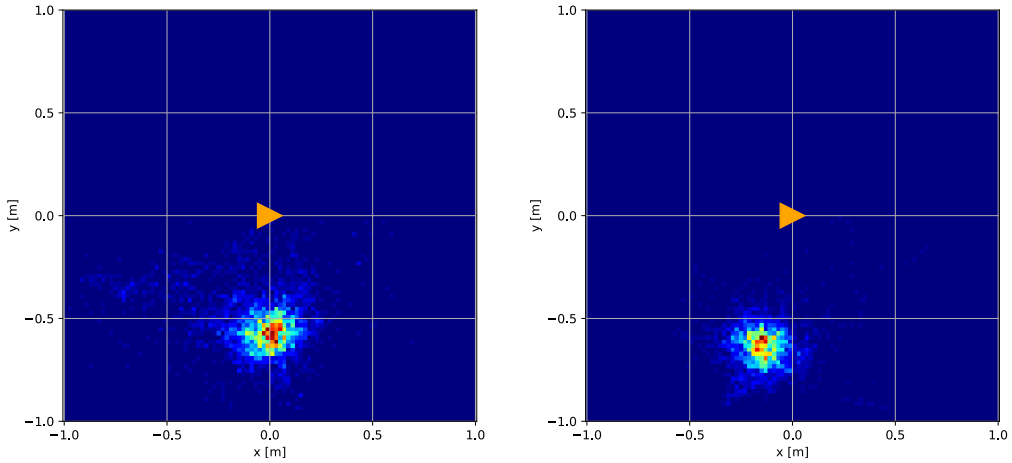


(B) The caregiver's walking speed (green), and altitude (blue).

FIGURE 4.19: The recorded attendant behavior in the outdoor environment.

Fig. 4.18 (a) shows the trajectories of the caregivers and the elderly persons at a corner, and it also suggests the importance of visibility for deciding the attending position. The number of the trajectories is 17. The caregivers tend to walk on the outer side of the corner (15 of 17). We can consider that, by walking at the outer side, the caregivers keep the outlook of the corridor to prevent accidents, such as stumbling and colliding. The caregivers walk on the inner side in a few cases (2 of 17). However, they preceded the elderly persons in order to check the safeness before the elderly persons enter the corner. These results suggest that the caregivers always check the existence of other surrounding people and objects, such as wheelchairs, to prevent accidents.

Fig. 4.19 (a) shows the recorded trajectories in the outdoor environment. In this sequence, the elderly was fine to walk, and the caregiver did let him walk relatively freely while navigating him to return back to the hospital. Fig. 4.19 (b) shows the caregiver's walking speed and the elevation of her position in the global map. When the caregiver (and the elderly) was going up a slope, they got slow down to 1.0 $\sim$ 1.2 m/sec while they walked at 1.2 $\sim$ 1.4 m/sec in down slopes. Slopes influence not only their walking speed but also their position relationship. We extracted their behavior in up slopes and down slopes, respectively, and calculated the distributions



(A) Up slopes. (B) Down slopes.

FIGURE 4.20: The distribution of the relative position of the care giver with respect to the elderly person in an outdoor environment.

of the caregiver's relative position with respect to the elderly (see Fig. 4.20). We can see that, in down slopes, the elderly led the caregiver while they walked side by side in up slopes due to the change of the walking speed. Although the caregiver's "X-axis" position varies depending on the walking speed, he almost always stays at 0.6 m side from the elderly. This is also observed in indoor environments (see Fig. 4.17). These results suggest that, during attendance, professional caregivers adjust their position depending on the elderly persons' status and the surrounding environment, while keeping their side distance to the elderly persons constant. This can be applied to designing of person following robots. Most of existing person following robots just keep the distance to the target constant. However, it might be unnatural behavior for people. We can make the robot keep the side distance to the target constant, and it may contribute the naturalness of the following behavior of the robot.

Those analysis results are difficult to obtain using existing measurement systems which use static sensors or wearable devices, such as INS and GPS since it requires accurately measure people behavior with respect to other people and the surrounding environment. The results show that we can capture and analyze such people behavior with the proposed system.

Chapter 5

Awareness Estimation-based Attendant Robot Framework

5.1 Robotic Attendant based on Awareness Estimation

One of the significant symptoms of dementia is the lack of attention to objects [133]. Even though elderly persons with dementia keep ordinary body functions, the lack of attention leads them to get injured by, for example, bumping into obstacles and falling from steps. We consider that if we properly alert them to dangerous objects and situations, they can avoid such accidents by themselves. However, if we always inform them of the existence of obstacles, they may feel it annoying. Therefore, we need to assess the risk of an accident, and inform the elderly of that only when it is necessary.

In our framework, the robot assesses the risk of an accident by estimating person's awareness. If an elderly is not aware of an obstacle, he/she may bump into it. In this case, the robot should take an action, such as notifying the elderly of the obstacle, to prevent the accident. On the other hand, if he/she is aware of it, they avoid the obstacle by himself/herself, thus, the robot lets the elderly walk freely to avoid disturbing him/her.

Fig. 5.1 shows the proposed robotic attendant framework;

1. The robot observes the target person's behavior while following him/her.
2. The robot estimates his/her awareness of an obstacle and assesses the risk of an accident.
3. If the person is not aware of the obstacle and it's a dangerous situation;
4. The robot takes an action to prevent the accident.

To realize this, we propose a model to estimate a person's awareness from the person's behavior and the surrounding environment information. We take a machine learning approach to construct this model (see Fig. 5.2). We first collect a set of behavior data where a person is aware/unaware of an object, and train a machine learning model to discriminate them. Then, when the robot is in operation, it observes a person's behavior and judges if he/she is aware of an object by using the trained model. Unlike traditional models for person behavior [25, 26], our model does not rely on hand-crafted parameters, and it is able to model the complex relationship between a person's awareness, surrounding objects, and environments.

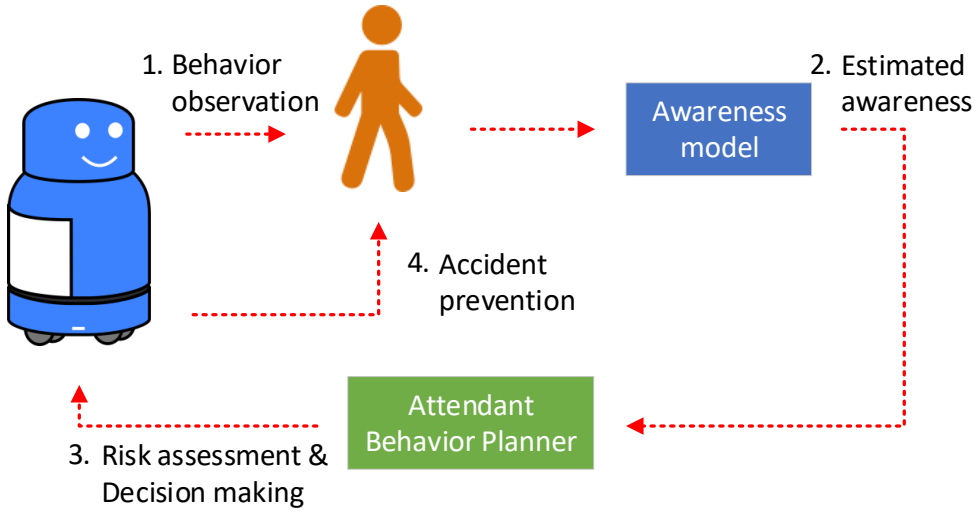
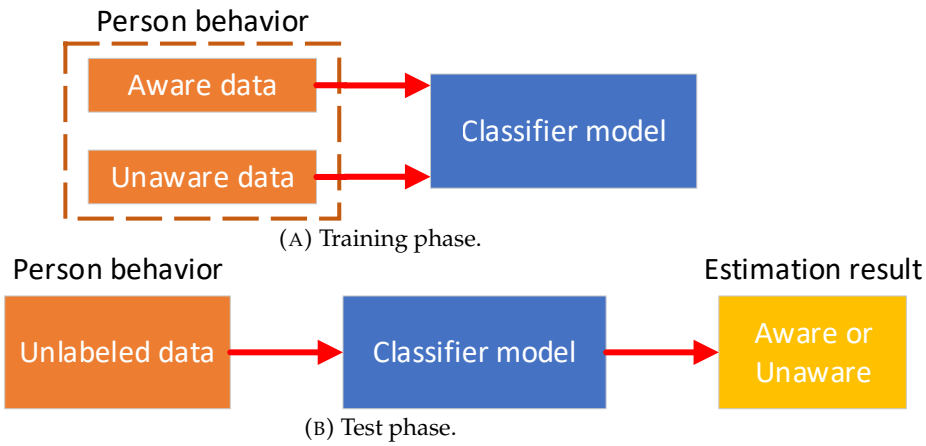


FIGURE 5.1: Proposed robotic attendant framework.



(A) Training phase.

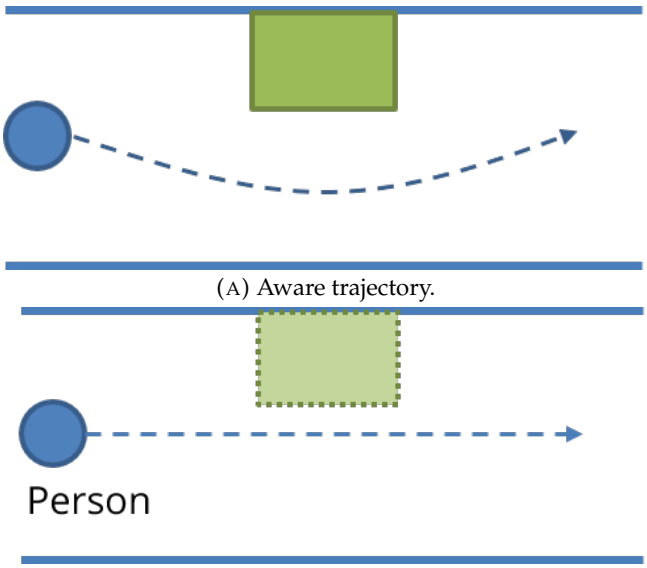
(B) Test phase.

FIGURE 5.2: Machine learning-based awareness estimation approach.

5.2 Simplifying Awareness Estimation Problem

An important issue in this approach is that it is very hard to obtain persons' unaware behavior data due to ethical and technical reasons. If a person is not aware of an obstacle, there is a risk that the person bumps into the obstacle and get injured. We need to manage and control the experiment very carefully, and it is not feasible to collect a bunch of behavior data required for the training of the model.

To deal with the problem of collecting unaware behavior data, we introduce a simple assumption; if a person is not aware of an object, the person acts as if there were no object. For instance, if a person is walking in a corridor and he/she is aware of an obstacle, the person changes his/her trajectory to avoid the obstacle (Fig. 5.3 (a)). On the other hand, if he/she is not aware of it, he/she moves as if there were no obstacle, and as a result, the person's behavior becomes the same as the one where the obstacle does not exist (Fig. 5.3 (b)). Our idea is that the person's aware/unaware state corresponds to the existence of the object, and it can be observed in the person's behavior. We collect a set of behavior data where an object exists/not exist, and then train a classifier from the behavior data. We consider that, by using the classifier which estimates the existence of an object from a person's behavior, we can judge whether the person is aware of the obstacle or not from his/her behavior.



(A) Aware trajectory.

(B) Unaware trajectory.

FIGURE 5.3: Influence of person's awareness on person's behavior. The unaware trajectory is the same as the trajectory where there is no obstacle.

5.3 Proof-of-concept: Estimating Person's Awareness of an obstacle

As a proof-of-concept, we introduce one of our works for awareness estimation. The purpose of this work is to show that a person's awareness of an obstacle can be estimated by only observing his/her motion. In this method, we first extract motion features from the person trajectory and model the relationship between the awareness and the motion using HCRF (Hidden Conditional Random Fields) [134]. We then estimate the person's awareness of the obstacle from the observed motion using the model. To simplify the problem, in this work, we assume a person walking in a straight corridor, and estimate his/her awareness of an obstacle (cardboard box) in the corridor (like shown in Fig. 5.3). Although, this model is designed for the limited situation, through this work, we validate that it is possible to estimate persons' awareness by observing their behavior.

5.3.1 Estimating the Awareness of an Obstacle

Person's Motion Features

In order to describe a person's motion with respect to an obstacle, we define the following four features (see Fig. 5.4).

1. Distance to the obstacle: When the person is close to the obstacle, the person's motion is affected strongly by the obstacle.

2. Distance to the skeleton of the hallway: This feature is designed to describe how the person's trajectory is affected by the obstacle. Since the person will move along the hallway if there is no obstacle, this feature will be changed by the existence of an obstacle.

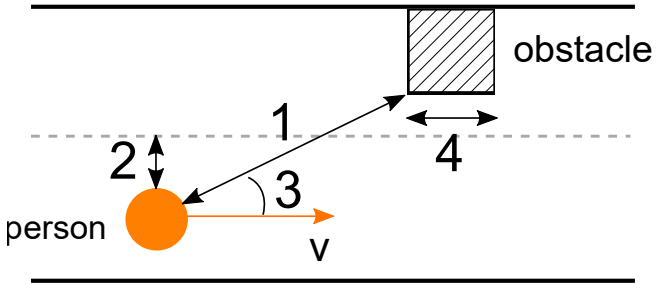


FIGURE 5.4: Person's motion features.

3. Angle between the velocity vector and the vector from the person to the obstacle: This feature represents whether the person moves toward the obstacle or not. If the person is avoiding the obstacle, this feature will be large.

4. Size of the obstacle: The person's motion may be affected by several characteristics of the obstacle. We simply use its size to model the obstacle.

Person's Awareness Model using HCRF

We represent the motion of a person as $\mathbf{x} = \{x_1, x_2, \cdots, x_t\}$ which is a sequence of motion features with length t. Let y be a binary label of a sequence denoting whether the person is aware of the obstacle or not. We assume that the person's motion is influenced by the condition of the person's awareness. This relationship can naturally be modeled using a sequence classifier, such as CRF (Conditional Random Fields) [135] and HCRF (Hidden Conditional Random Fields) [134]. In this work, we use HCRF to construct the model. We also use CRF as a baseline.

By introducing HCRF, we can model the relationship between the person's awareness and the person's motion as shown in Fig. 5.5. Following the work of [136], the relationship is modeled as:

$$P(y|\mathbf{x}, \theta) = \sum_{\mathbf{h}} P(y, \mathbf{h}|\mathbf{x}, \theta) = \frac{\sum_{\mathbf{h}} \exp^{\psi(y,\mathbf{h},\mathbf{x};\theta)}}{\sum_{y',\mathbf{h}} \exp^{\psi(y',\mathbf{h},\mathbf{x};\theta)}}, \tag{5.1}$$

where θ is the parameter of the model, ψ is a potential function parameterized by θ. A sequence of hidden states $\mathbf{h} = \{h_1, h_2, \cdots, h_t\}$ is introduced as the possible hidden labels inside the model. In our model, the number of possible values of each hidden state is set to three.

The parameter θ is optimized using a stochastic descent method [134], and then, we estimate the label of the sequence as follows:

$$\arg\max_{y} P(y|\mathbf{x}, \theta). \tag{5.2}$$

We obtain observations every 0.5 [s] and use six consecutive observations as one sequence. The duration of a sequence is 3 [s]. We assume that the duration is long enough to describe the person's obstacle avoiding motion.

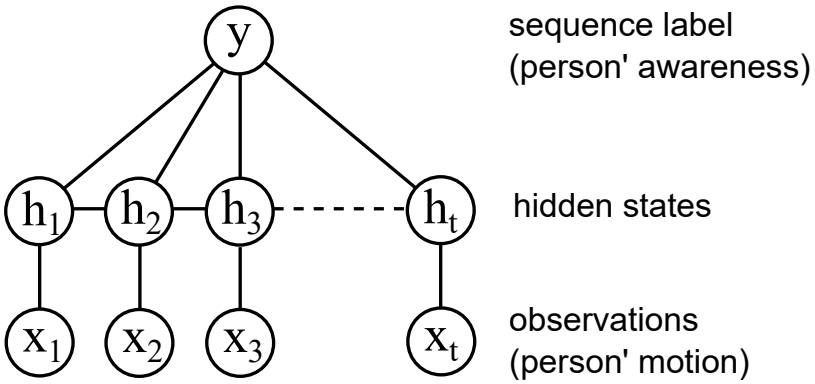


FIGURE 5.5: Person's awareness model.

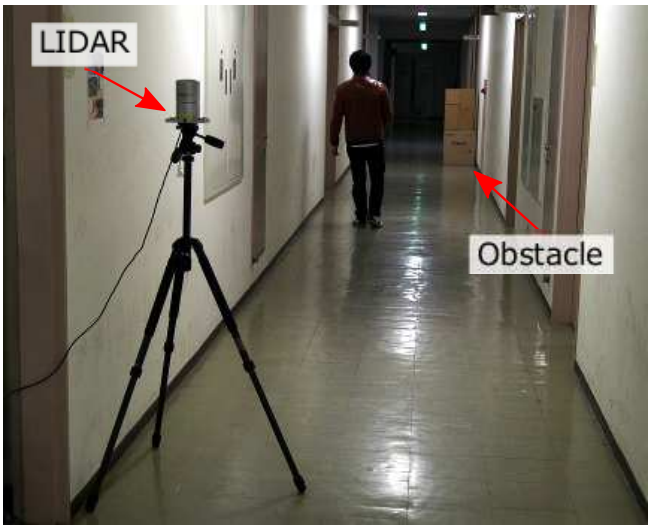


FIGURE 5.6: Experimental environment.

5.3.2 Experiments

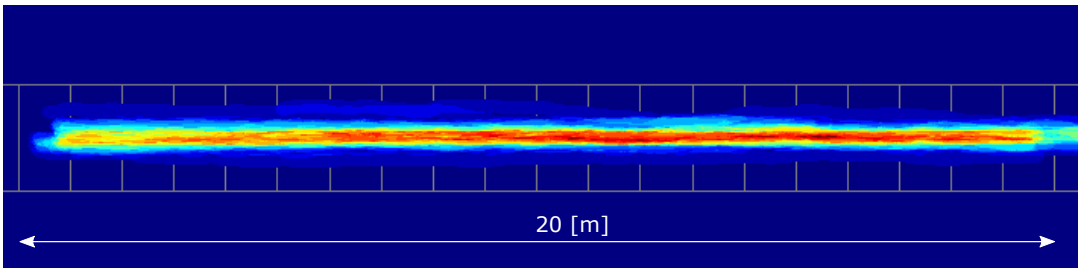
Awareness Estimation Experiments

As explained in Sec 5.2, we collect people behavior with and without obstacles instead of aware/unaware hahavior, and train a classifier which classifies them.

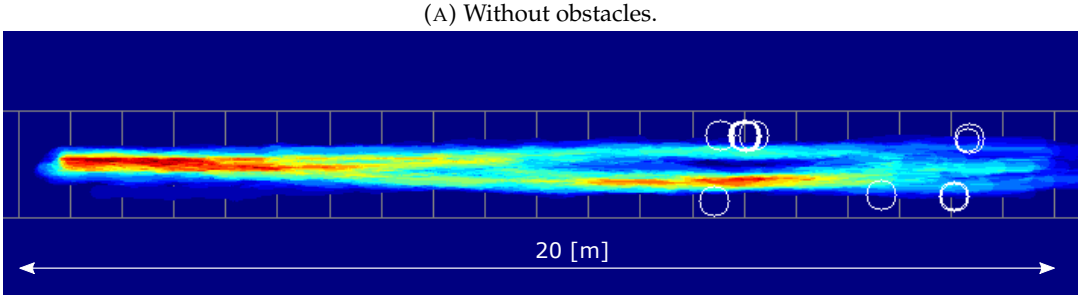
We first collected a set of person trajectories with and without obstacles. Fig. 5.6 shows the experimental setting. A person walking in the corridor is tracked by the way described in Sec. 4.5.2. The experiments were conducted under two settings; in the first one, an obstacle was placed at a random position in the corridor, and in the second, no obstacle was placed. Five persons walked in the corridor and avoided the obstacle if there was an obstacle. We measured the person's trajectory 30 times for each person with and without obstacles, respectively.

Fig. 5.7 shows the heatmap created from the measured trajectories. Red indicates where the persons passed on frequently, and blue indicates where the persons did not pass. The white circles indicate the size and the position of the obstacles. As we can see in Fig. 5.7, the person's motion is affected by the obstacles. If there is no obstacle, persons move straight along the hallway. On the other hand, if there is an obstacle, persons change their trajectories to avoid the obstacle.

In situations where a person is unaware of an obstacle, the person's motion is independent of the obstacle. To simulate the situation using the situations without



(A) Without obstacles.



(B) With obstacles.

FIGURE 5.7: Heatmap of persons' trajectories. Red indicates where the persons passed on frequently, and blue indicates where the persons did not pass. The white circles indicate the size and the position of the obstacles.

TABLE 5.1: Estimation Results.

Method	Precision	Recall	F1
CRF	0.743	0.745	0.744
HCRF	0.921	0.941	0.931

obstacles, we randomly choose obstacle data from the situations with obstacles and extract the person's motion features as if there were a chosen obstacle. We train the HCRF model using the extracted features.

The set of trajectories is divided into five parts, and one of them is used as a test set, and the rest are used as a training set. The number of the motion sequences in the test set is 785, and the number of the sequences in the training set is 3146. Table 5.1 shows the estimation results. HCRF shows a better estimation performance than CRF, and in the case of HCRF, we achieve an estimation accuracy of 92.1%. Fig. 5.8 shows the relationship between the distance to the obstacle and the estimation accuracy. As a person get closer to an obstacle, the person's motion is influenced by the obstacle strongly, and the motion becomes distinguishable from when the case without the obstacle. As a result, the estimation accuracy increases. When the distance between the person and the obstacle is less than 4 [m], the method can estimate the person's awareness with an estimation accuracy of over 90%.

Online Awareness Estimation Experiments

We collected additional three persons' trajectories without obstacles and nine trajectories with obstacles for an online test. In order to validate the applicability of the proposed method to real attendant robots, we examine the point where the method judged that the person was aware of the obstacle.

Fig. 5.9 shows examples of the estimation results. Thick lines indicate the trajectory of a person and estimation results. Blue color indicates that the system is accumulating motion data and is not classifying the motion due to an insufficient

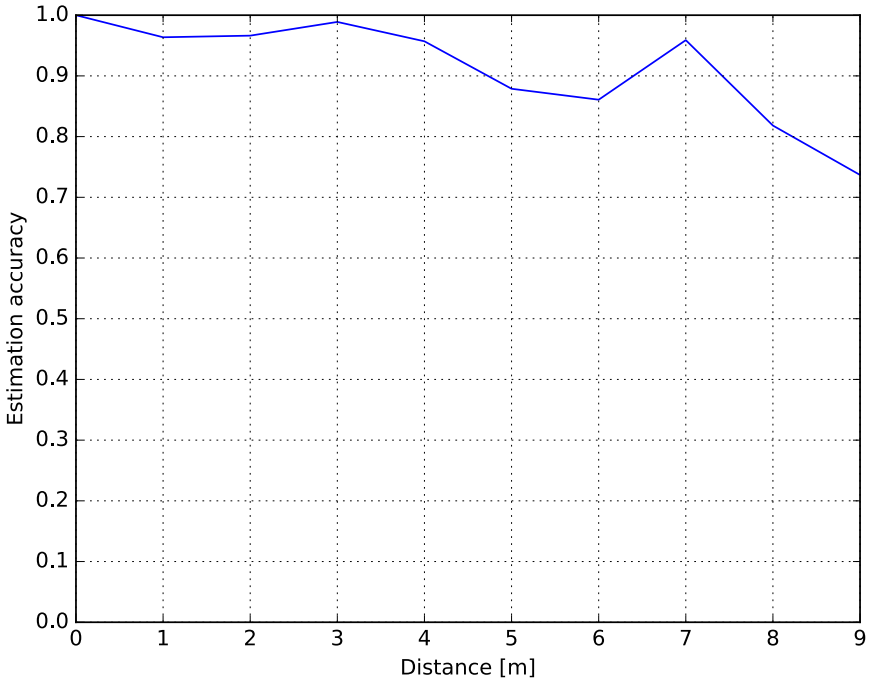
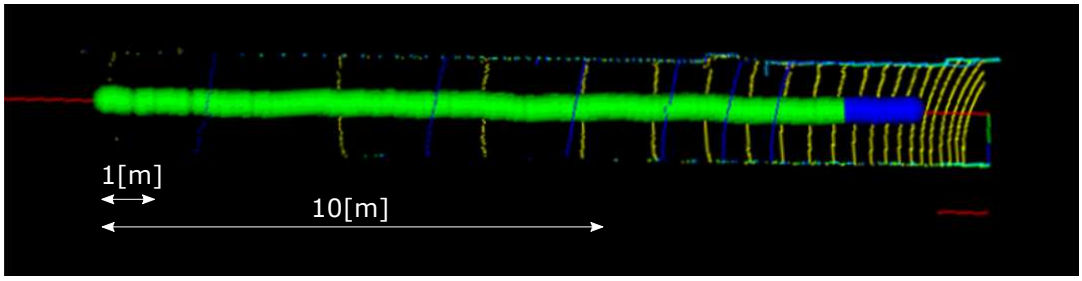
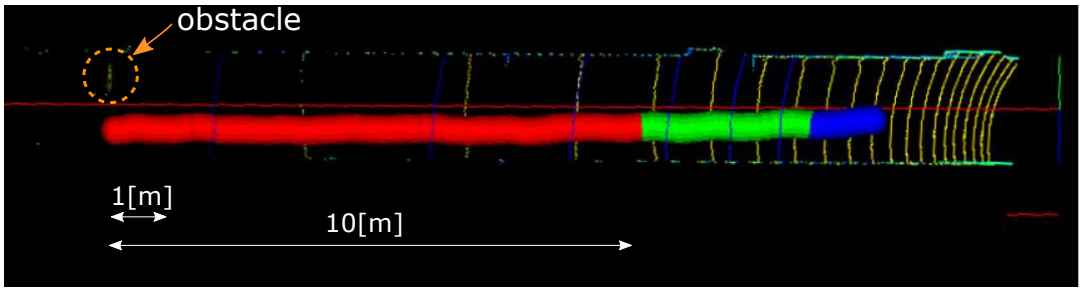


FIGURE 5.8: The relationship between the distance to the obstacle and estimation accuracy.



(A) Without obstacles.



(B) With obstacles.

FIGURE 5.9: Examples of estimation results. Thick lines indicate the trajectory of a person and the estimation results. Blue color indicates that the system is accumulating motion data and is not classifying the motion due to an insufficient amount of data. Green and red colors indicate that the person is unaware of the obstacle, and that the person is aware of the obstacle, respectively.

amount of data. Green and red colors indicate that the person is unaware of the obstacle, and that the person is aware of the obstacle, respectively. In the case of Fig. 5.9(a), the system started to accumulate the person's motion data when the person entered into the environment. After a sufficient amount of motion data is accumulated, the system successfully classified the person's motion as being unaware of the

TABLE 5.2: Statistics of the point where the classifier judged that the person is aware of the obstacle.

	mean	std. dev.	min	max
distance [m]	8.53	1.88	6.09	11.41

obstacle. In the case of Fig. 5.9(b), after the accumulation of data was finished, the system classified the person's motion as being unaware of the obstacle. However, as the person got closer to the obstacle, within about 10 [m], the system judged that he was aware of the obstacle. When a person is close to an obstacle, the system reliably estimates the person's awareness since the identification accuracy increases as a person gets closer to an obstacle as shown in Fig. 5.8.

In all of the cases without obstacles, the classifier did not judge that the person was aware of the obstacle, and in all of the cases with obstacles, the classifier successfully judged that the person was aware of the obstacle before the person reached to the obstacle. Table 5.2 shows the statistics of the point where the classifier judged. The classifier can realize that a person is aware of an obstacle at a point about 8.5 [m] from the obstacle on average, and about 6.1 [m] at least. If the person is walking at 1.2 [m/s], the time to bump into the obstacle is about 5.1 [s]. If the robot takes preventative action within this time, it can avoid the collision. We consider that the robot can interact with the person within this duration if the robot approaches the person in advance. At least the robot can call the person to make the obstacle attract their attention within this duration.

5.4 Deep Neural Network-based Awareness Estimation

In the previous section, we verified that we can estimate a person's awareness of an obstacle by observing his/her behavior. However, the proposed model was too simple, and it cannot be applied to complex environments. In order to apply the model to real complex environments, we extend it with a deep convolutional neural network. The network takes a person's behavior and environmental information, and outputs the person's awareness of the surrounding environment (see Fig. 5.10).

As the input of the network, we use a sequence of local maps (the person is located at the center of each map), and as the output, we consider a distribution map which represents the position of obstacles, which the person is going to bump into due to the lack of awareness. We also make the network predict the person trajectory in the following frames since it has a significant relation with the awareness estimation. To estimate a person's awareness, we need to know how he/she will move. On the other hand, the person's behavior would be influenced by the person's awareness of surrounding objects. It is a kind of chicken-and-egg problem, and thus, in this work, we propose the network which simultaneously estimates a person's awareness and trajectory.

As explained in Sec. 5.2, in order to avoid the unaware behavior collection problem, we assume that a person, who is not aware of an object, moves as if the object does not exist. It means the existence of an object does not influence the person's behavior if he/she is not aware of it. Thus, we can say that persons' unaware behavior is independent from obstacle properties. If we put a virtual obstacle at the person's position at time t, we can consider the person's trajectory until t as an imitation of a trajectory where the person bumps into the virtual obstacle (see Fig. 5.11).

We put virtual obstacles at random positions (obs_j) on the environmental map, and then generate awareness and trajectory maps to be estimated from the virtual obstacle positions and the person trajectory (p_t). Fig. 5.12 illustrates the awareness map generation. The awareness map y^{aware} is defined by:

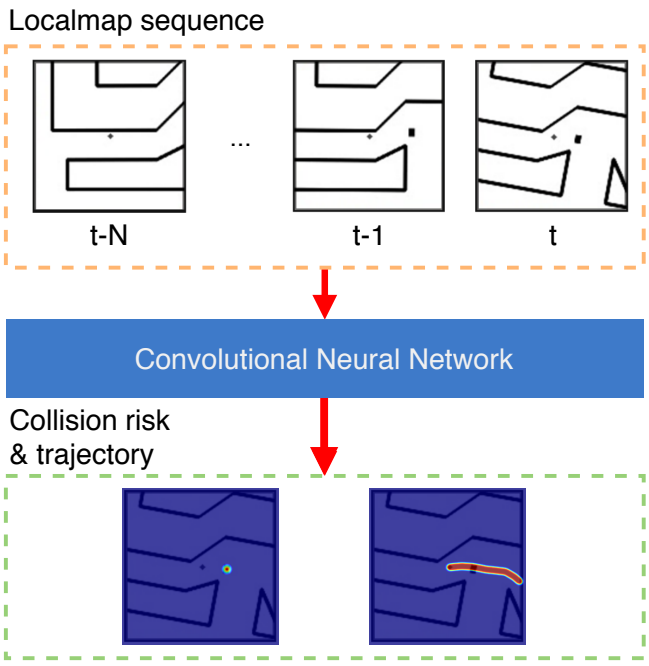
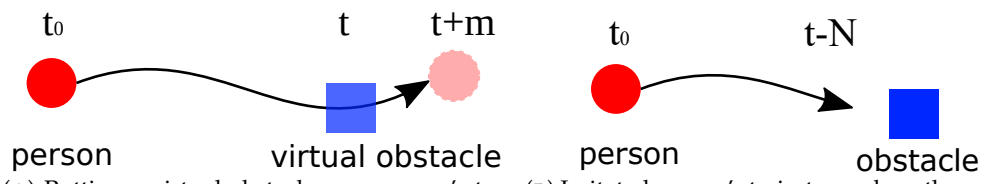


FIGURE 5.10: The proposed awareness estimation model.



(A) Putting a virtual obstacle on a person's trajectory.

(B) Imitated person's trajectory where the person bumps into the obstacle.

FIGURE 5.11: Generation of persons' unaware behavior data by putting a virtual obstacle.

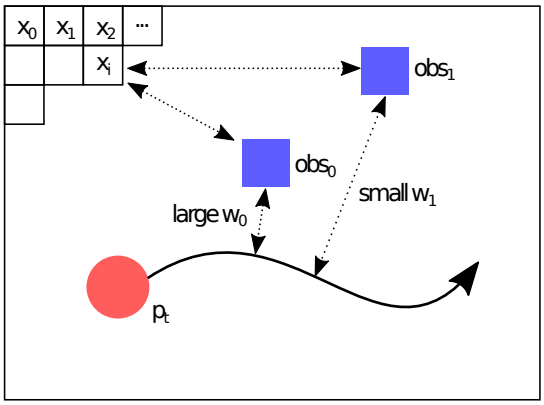


FIGURE 5.12: Awareness map generation.

$$w_j = \exp\left(\frac{-\min_t \|obs_j - p_t\|}{C_\alpha}\right), \tag{5.3}$$

$$y_i^{aware} = \sum_j w_j \cdot \exp\left(\frac{-\|obs_j - x_i\|}{C_d}\right), \tag{5.4}$$

where, w_j represents the weight of the obstacle j given by the minimum distance between the obstacle position obs_j and the person trajectory p_t. We give high weights to obstacles which the person gets close to, since there is a high risk that the person bumps into them. x_i is the position of the pixel i in the local map coordinate, y_i^{aware} is the calculated awareness distribution value, and C_α and C_d are constants.

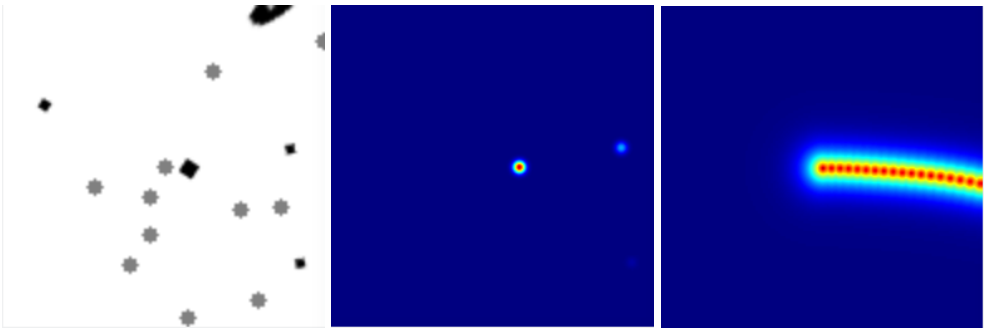
The trajectory map y^{traj} is calculated as:

$$y_i^{traj} = \exp\left(\frac{-\min_t \|p_t - x_i\|}{C_t}\right), \tag{5.5}$$

where and C_t is a constant.

Fig. 5.13 shows an example of input localmaps and corresponding awareness and trajectory maps. The black and gray pixels in the localmap represents obstacles and people, respectively. In the awareness map (Fig. 5.13 (B)), we can see a strong response on the obstacle which the person is going to bump into, while the response on the other obstacle is weak since the person will not get close to it.

To estimate awareness and trajectory maps from input localmaps, we use the *U-Net* architecture [137], which is similar to the usual convolutional *Encoder-Decoder* network model [138] except for skip connections. It first applies convolution layers to extract structured features from an input map, and then applies deconvolution layers to expand the extracted features to the output map with the same dimension as the input. The difference between the *U-Net* and the usual *Encoder-Decoder* model



(A) Input localmap (B) Awareness map (C) Trajectory map

FIGURE 5.13: The input localmap and the corresponding awareness and trajectory maps. The black and gray pixels in the localmap represents obstacles and people, respectively.

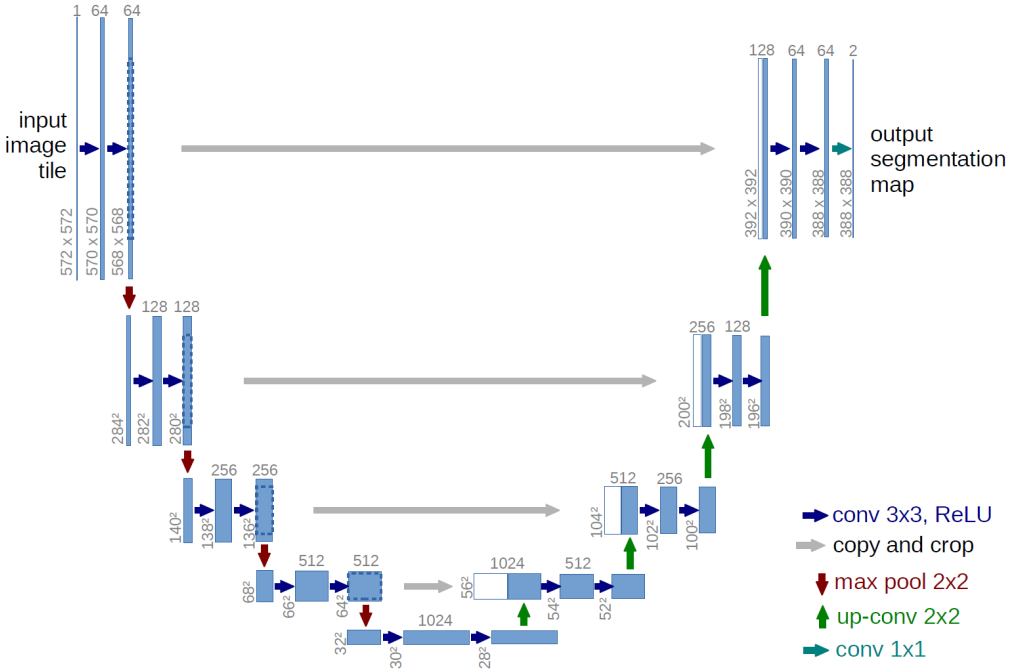


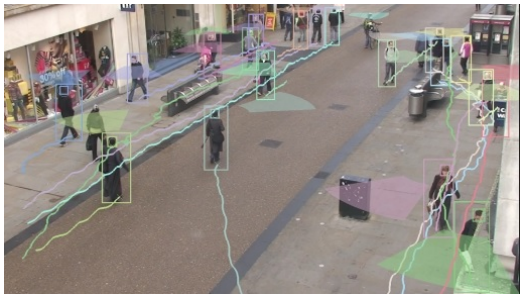
FIGURE 5.14: The *U-Net* architecture [137].

is that the *U-Net* has *skip connections* between the convolution layers and the corresponding deconvolution layers (see Fig.5.14). It allows us to effectively train the network since the skip connections tells the raw-level information, which can be lost by applying convolution filters, to the deconvolution layers and helps to avoid the vanishing gradient problem.

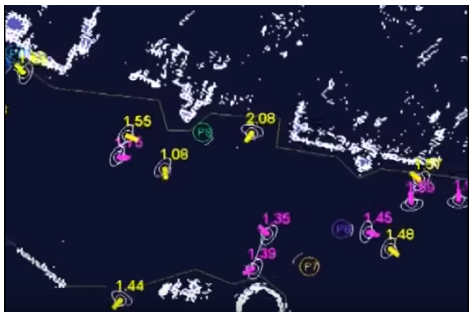
We extend the *U-Net* to be a recurrent network with LSTM (Long Short-Term Memory). We put an LSTM layer at the neck-part of the *U-Net*. Since recurrent neural networks can naturally exhibit dynamic temporal behavior of inputs, they would show better performance than feed forward networks when the input sequence contains complex time series behavior. With this recurrent *U-Net* architecture, we input localmaps to the network one by one, and the network outputs an awareness map and a trajectory map at each time step.

(A) PETS2009 [139].

(B) 3DPES [106].



(C) Towncenter [107].



(D) ATC shopping center [105].

FIGURE 5.15: The datasets used to create a real people behavior dataset.

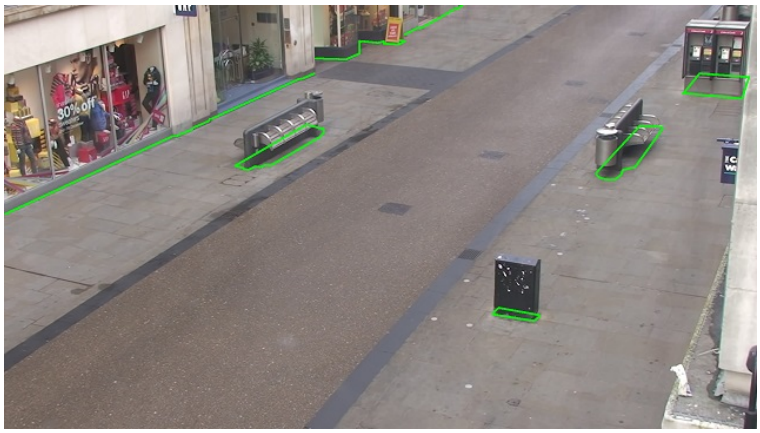
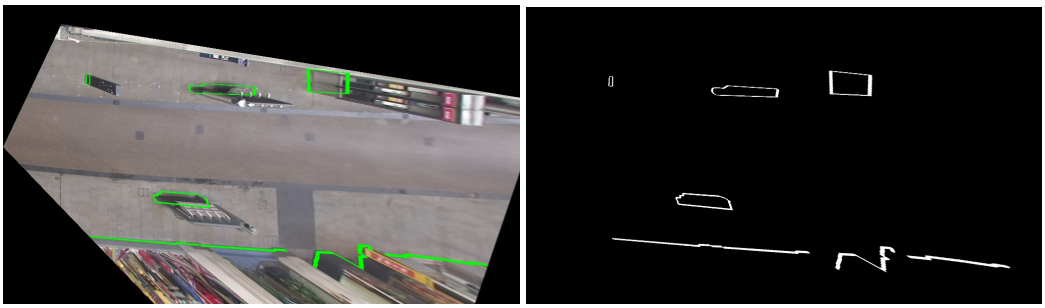


FIGURE 5.16: An obstacle footprint map created by hand.

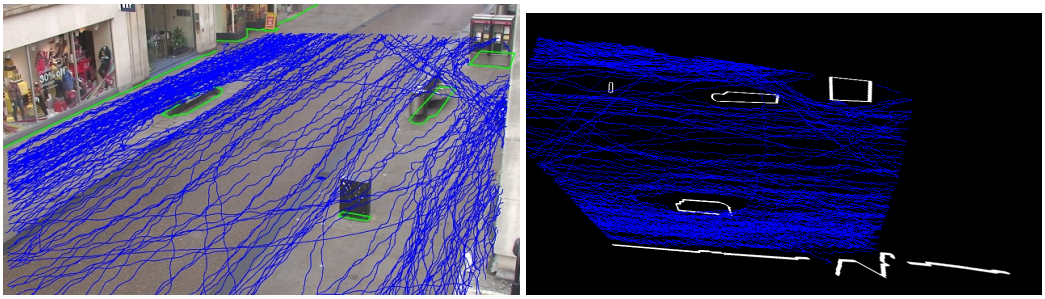
5.5 Training of the Awareness Estimation Model with Real People Behavior Data

In order to apply this framework to realistic situations, we train the model with real people behavior. We use *Towncenter* [107], *PETS2009* [139], *3DPES* [106], and *ATC shopping center* [105] datasets to create a real people behavior dataset (see Fig.5.15). *ATC shopping center* dataset consists of 3D people trajectories in a shopping center environment, and we can directly use the trajectories to create people behavior sequences. *Towncenter*, *PETS2009*, *3DPES* provide people trajectories in a camera optical space and intrinsic and extrinsic calibration parameters of each camera. To create people behavior sequences with the local map representation, we need to convert the trajectories in a camera space into a bird's eye view. We first create an obstacle



(A) Projected obstacle footprint map. (B) Environmental map.

FIGURE 5.17: Environmental map generation.



(A) The people trajectories in the camera space. (B) The people trajectories projected in the bird's eye view.

FIGURE 5.18: People trajectories provided by *Towncenter* dataset.

footprint map for each camera by hand (see Fig. 5.16), and then project it into the bird's eye view using the extrinsic parameters provided by the dataset (see Fig. 5.17) to obtain the map of the environment. Then, people trajectories are also projected into the map space, and we create sequences of local maps from the environmental map and the projected person trajectories (see Fig. 5.18).

Algorithm 2 Virtual obstacle placement

Input: $\mathcal{P} = \{\boldsymbol{p}_0, \ldots, \boldsymbol{p}_T\}$, person trajectory
Input: σ_o^2, variance of obstacle position
Input: $int_{min/max}$, minimum/maximum interval of obstacles
Output: $\mathcal{O} = \{\boldsymbol{obs}_0, \ldots, \boldsymbol{obs}_N\}$, virtual obstacles

```
O ⇐ {}
t ⇐ U(int_min, int_max)
while t < T do
    obs = p_t + N(μ = 0, σ² = σ_o²)
    O ⇐ O ∪ {obs}
    t ⇐ t + U(int_min, int_max)
end while
```

Algorithm 2 shows the algorithm to place virtual obstacles on the environmental map based on the person trajectory. $\mathcal{N}$ and $\mathcal{U}$ are sampling from normal and uniform distributions, respectively. In this algorithm, we place virtual obstacles at positions deviated from the trajectory. We first place obstacles which the person bumps into with $\sigma_o^2 = 0$[m], $int_{min} = 3.0$[s], $int_{max} = 10.0$[s], then place obstacles which the person does not bump into with $\sigma_o^2 = 5.0$[m], $int_{min} = 3.0$[s], $int_{max} = 5.0$[s].

The number of sequences is about 2000. We train the proposed network on the

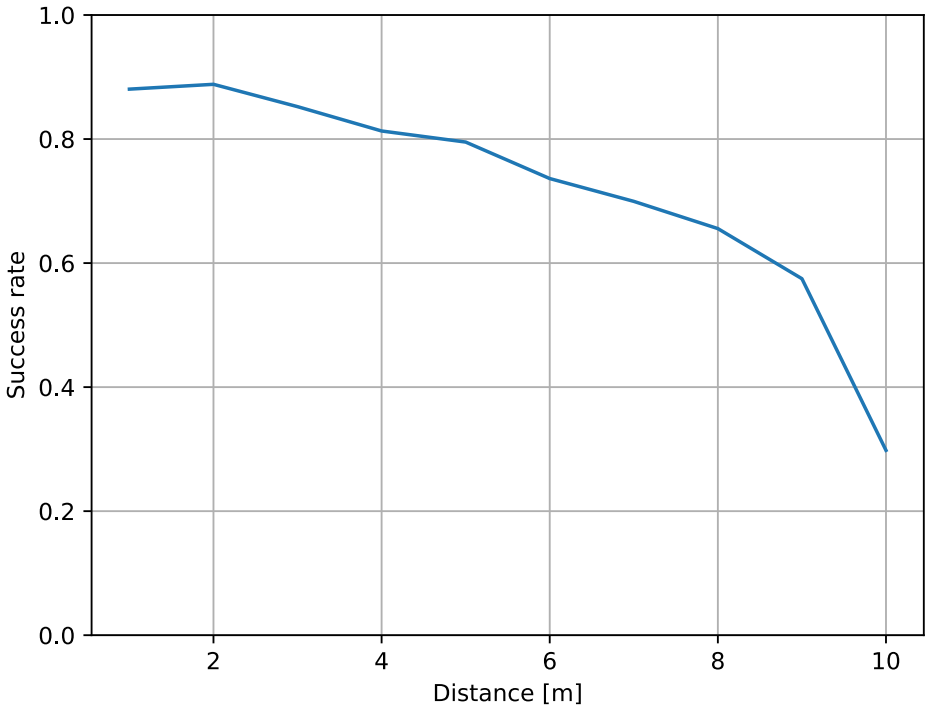


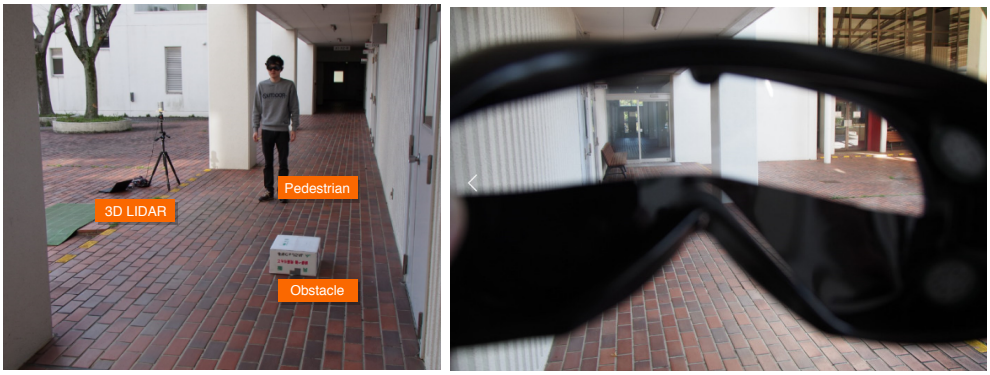
FIGURE 5.19: The plot of the awareness estimation accuracy versus the distance to the obstacle.

generated dataset. We use the L2 loss function to calculate the residual of the estimated awareness and trajectory maps. We give a small weight (10^{-3}) to the residual of the trajectory map in the L2 loss calculation, since the awareness map is more important than the trajectory map in our application, and the trajectory is sometimes unpredictable.

To test the trained model, we randomly sampled 100 sequences from the dataset, and tested if the model correctly detects obstacles which the person is not aware of. We replaced the obstacles in the sequences, thus, their positions are changed from the training set. With thresholding, we extract obstacles which the person gets closer than 0.5 [m] from the estimated awareness maps, and check if they are properly detected. Fig. 5.19 shows the plot of the success rate of the detection and the distance to the obstacle. We can see that, when the obstacle is very far from the person, the estimation accuracy is low (30% at 10 [m]). However, as the person gets close to the obstacle, the estimation accuracy gets increased. At the point of 4 [m], we achieve over 80 % of accuracy.

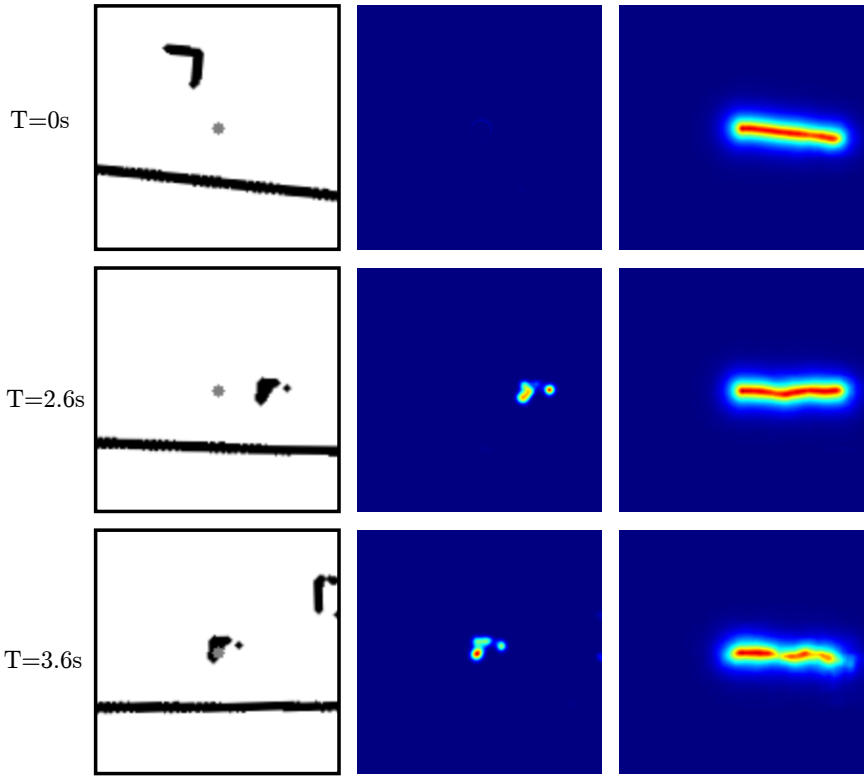
5.6 Model Validation on Real Data

To validate the trained awareness estimation model, we collected a set of people aware and unaware behavior data. Fig. 5.20 (A) shows the experimental environment. In this experiment, subjects wear a half-blind glasses (Fig. 5.20 (B)) so that he cannot see an obstacle on the ground and walks in the corridor. The obstacle position is changed at every trial, and the subject is not told where it is. To prevent any critical accidents, we chose a light cardboard box as the obstacle to be placed in



(A) The experimental environment. (B) The half-blind glasses.

FIGURE 5.20: The experimental setting. The subject wears a half-blind glasses so that he cannot see the obstacle on the ground and walk in the corridor. We predict if he bumps into the obstacle by using the trained awareness estimation model.

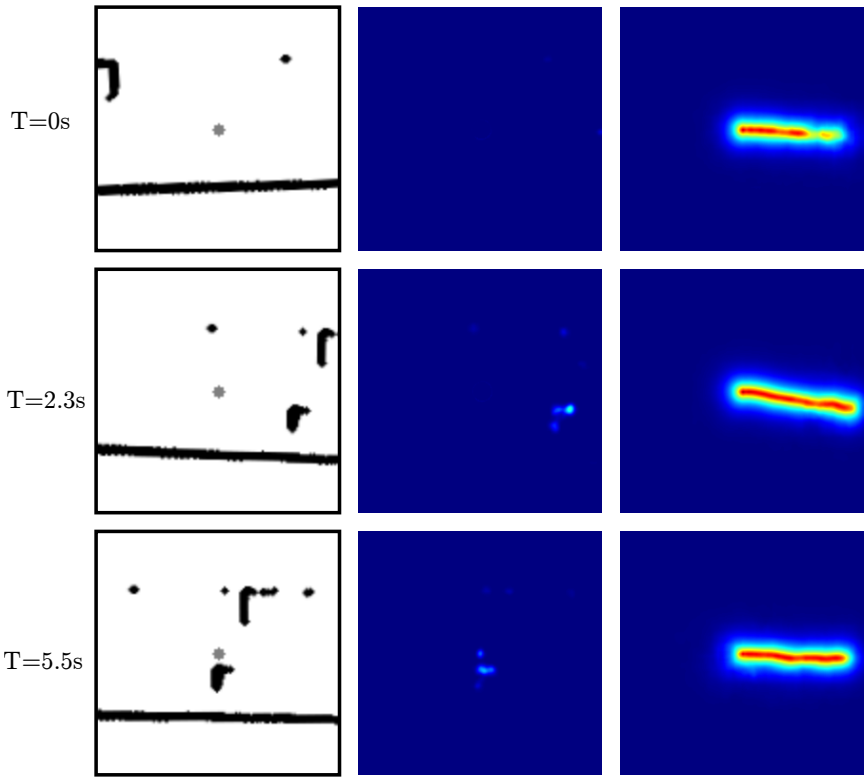


(A) Local map (B) Awareness map (C) Trajectory map

FIGURE 5.21: A trial where the person bumped into the obstacle. The left images show the input localmaps, and the center and right maps show estimated awareness and trajectory maps.

the corridor. In some trials, the subject bumped into the obstacle, and we consider the subject behavior in these sequences as unaware behavior data. We collected 15 sequences with four subjects in total. The subjects bumped into the obstacle in 10 out of the 15 sequences.

Fig. 5.21 shows a trial where the person bumped into the obstacle. The left figures show the input local maps, and the center and right images show estimated awareness and trajectory maps, respectively. At $T = 0$[s], we can see that, the network correctly estimates that the person will move along the corridor (Fig. 5.21(C)).



(A) Local map (B) Awareness map (C) Trajectory map

FIGURE 5.22: A trial where the person did not bump into the obstacle.

TABLE 5.3: Awareness Estimation Accuracy

	F1	precision	recall	TP	TN	FP	FN
Far (10 [m])	0.909	0.833	1.000	10	3	2	0
Middle (5 [m])	0.909	0.833	1.000	10	3	2	0
Near (1 [m])	0.952	0.909	1.000	10	4	1	0
Total	0.923	0.857	1.000	30	10	5	0

TP: True Positive, TN: True Negative,
FP: False Positive, FN: False Negative

As the subject approaches to the obstacle, the estimated awareness map shows high response at the position of the obstacle which he is going to bump into (Fig. 5.21(B)), and finally, he bumped into the obstacle (Fig. 5.21(B)). Fig. 5.22 shows another trial where the subject did not bump into the obstacle. We can see that, in this trial, the awareness map shows lower response on the obstacle position until the subject passes by the obstacle (Fig. 5.22(B) (C)).

To evaluate the proposed network quantitatively, we sample estimated awareness maps when the distance between the subject and the obstacle gets 10, 5, and 1 [m] for each sequence, and validate if the network correctly estimated the person's awareness of the obstacle. We apply thresholding to the awareness maps, and if the number of positive pixels is larger than a threshold, we consider that the network judged that the person is not aware of the obstacle.

Table 5.3 shows the evaluation result. Through the evaluation, the network correctly estimated the persons' awareness in the positive cases where the person is not aware of the obstacle and shows a good recall rate. In two sequences of negative

cases (the person is aware of the obstacle), the network wrongly judged that the person is not aware of the obstacle. However, as the person gets close to the obstacle, the awareness map response gets smaller, and the network correctly judged that he is aware of the obstacle when the distance between the person and the obstacle gets smaller than 5 [m]. In the other sequence, the network could not judge that he is aware of the obstacle until he passed by the obstacle. However, in practical situations, this kind of a few false positives are acceptable while false negatives are not acceptable since they would make the robot miss the chance to prevent accidents.

Chapter 6

Conclusions and Discussion

6.1 Conclusions

We have described a robotic attendant framework based on robust person identification and awareness estimation. We have presented robust person identification methods for two scenarios; identifying a person using only sensors on the robot, and identifying a person using signals obtained by a smartphone held by the person. Depending on the use case, we can choose one of them for robust person following. The proposed deep convolutional channel features and illumination independent gait and height features-based identification methods greatly improve the performance of the online person identification framework in severe illumination conditions while the foot strike timing-based method realizes a marker-based identification without any special equipment like antennas.

We have also described a system for people behavior measurement using a 3D LIDAR. It allows us to measure and analyze long-term and wide-area people behavior data. With this system, we collected professional caregivers' behavior in a hospital. The analysis of the caregivers' behavior reveals how human decides attending position while keeping the safeness and the comfortableness of attendance.

We have also proposed a deep convolutional network-based method to estimate a person's awareness of surrounding obstacles. In order to avoid the unaware behavior collection problem, we take a simple assumption of the influence of a person's awareness on his/her behavior. With this assumption, instead of training the network with persons' aware/unaware behavior, we train the network with persons' trajectories where an object exists/non-exists. The use of the deep neural network allows us to construct an awareness estimation model which is applicable to various environments and can estimate a person's trajectory and awareness simultaneously.

6.2 Proposal for a Robotic Attendant System Design

6.2.1 Robotic Attendant System

As a summary of the thesis, we present a proposal for a system design of an attendant robot based on robust person identification, awareness estimation, and the analysis of professional caregivers' attendant behavior.

Fig. 6.1 shows the system configuration of the proposed system. The system is built on top of fundamental person following functions, such as localization, person identification, and path planning. With the appearance and soft-biometric features-based person identification described in Sec. 3.1 and Sec. 3.2, the robot reliably follows the target person. In case it is allowed to let the person hold a smartphone, the identification method can be incorporated with the foot strike timing matching-based identification method presented in Sec. 3.3. The gait-based identification

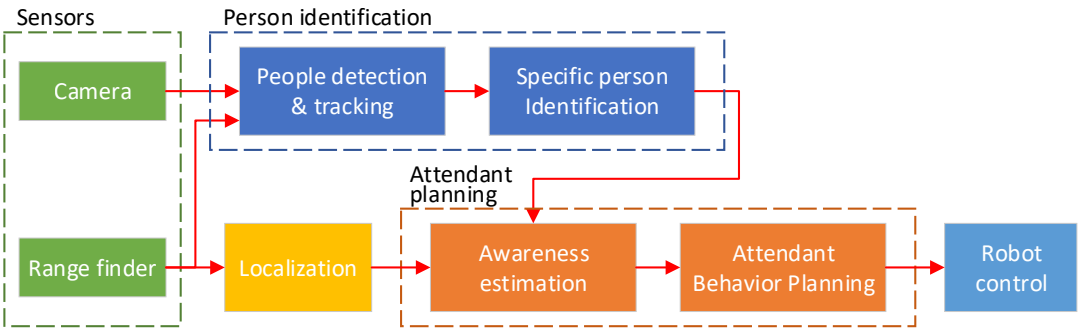


FIGURE 6.1: System configuration of the proposed attendant robot.

methods may not work well in case the person walks with difficulty due to stiff leg since the methods assume steady walking (the step length is constant). However, such a walking pattern could be a distinct feature to identify the person. By extending the gait feature so that it takes the unsteady walking gait into account (e.g., adding asymmetric left/right foot step length), the system could be more suitable for identification of elderly persons.

Then, the awareness estimation module takes the tracked target person's state (position and velocity) and the environmental information (the environmental map and the position of the robot estimated by the localization module) to estimate the person's awareness of surrounding obstacles, and the system assesses the risk of accidents from the estimated awareness. A possible concern here is that the awareness estimation model trained from behavior of people who are fine to walk may not be adaptable to elderly persons. Typically, elderly persons tend to walk slowly, and their walking is sometimes unsteady. However, we consider that the model can be applied to elderly persons by re-training the model with elderly persons' behavior data since the proposed deep convolutional neural network-based method does not take any human motion assumptions.

The attendant behavior planning module generates the robot behavior. In case an accident is anticipated, the robot takes an action to prevent the accident. On the other hand, as long as there is no risk of accidents, the robot lets the person walk freely. While the robot is following the person, it imitates the attending behavior of professional caregivers analyzed in Sec. 4.6.2 so that it does not disturb the person.

6.2.2 Basic Person Following Behavior

As long as there is no risk of accidents, the robot lets the person walk freely. Based on the analysis of the real caregivers' behavior presented in Sec. 4.6.2, we propose the design of the basic following behavior of the attendant robot as follows:

1. The robot attends the person while keeping the side-by-side positioning as long as it's possible. This positioning allows the robot to look ahead and check the safeness of the way. In particular, it should keep in the position 0.6m aside from the person.

2. Depending on the walking speed, the relative position would deviate along the front-back direction. However, even in such a case, the robot should keep the certain distance aside from the person.

3. At a corner, the robot should go on the outer-side of the corner so that it can check the safeness of the corridor while avoiding disturbing the person.

4. In case the robot cannot go on the outer-side due to positioning or obstacles, it should go on the inner-side before the person enters the corner and check if it's safe. It would slightly disturb the person from walking. However, the safety has a higher priority than the comfortableness in this case.

5. To attend a person who is fine to walk, the robot has to be able to run at about 1.4 m/s.

Note that the values in the rules, such as the distance to the person to be attended, should be adjusted depending on the robot configuration (e.g., size and shape).

6.2.3 Attendant Behavior Planning Strategy

Once the robot detects that the person is not aware of an obstacle and going to bump into it, the robot has to interact with the person to prevent the accident.

The timing to take an accident prevention is important. As a person gets close to an obstacle, the possibility that he/she notices it increases, and if the timing of the decision is too early, the robot would take an action before the person becomes aware of the obstacle by himself/herself. On the other hand, if the timing is too late, the robot cannot prevent the accident. We have to find the best timing for safeness and comfortableness.

We can estimate when a person will bump into an obstacle from the person's walking speed and the obstacle position, and it could be a good factor to decide the time to take a preventing action. The robot should start to take action by considering the estimated accident time, the time required to perform the action, and a time margin for safety.

As a baseline, we consider a simple prevention action; if the target person is not aware of an obstacle, the robot informs him/her of the obstacle by voice. The timing to take the action is given by $t_e - t_a - t_m$, where t_e is the expected accident time, t_a is the time required to perform the action (e.g., 1.5 sec), and t_m is a time margin (e.g., 1.0 sec). t_e is given by $\frac{d}{s}$, where d is the distance between the person and the obstacle, and s is the person's walking speed.

The choice of the prevention action has a big impact on the comfortableness of the system. Human caregivers prevent accidents in an unsure but less annoying way in low risk situations while they take a compelling way in high risk situations. For example, they try to change the elderly's trajectory by getting proximate to him/her when an obstacle is distant, and if the elderly does not change the trajectory before he/she gets close to the obstacle, they pull his/her hand or informing him/her of the obstacle by voice to let the elderly avoid the obstacle. This kind of adaptive prevention action would be a way to strike the balance between the risk of accidents and the comfortableness of the service, and it would be a future direction of research.

Bibliography

[1] Japanese Ministry of Health, Labour and Welfare, Welfare human resource securing meeting (2014).
URL http://www.mhlw.go.jp/stf/shingi/other-syakai.html?tid=198696

[2] V. Faucounau, Y.-H. Wu, M. Boulay, M. Maestrutti, A.-S. Rigaud, Caregivers' requirements for in-home robotic agent for supporting community-living elderly subjects with cognitive impairment, Technological Health Care 17 (1) (2009) 33–40.

[3] RT. Works, Rt. 2: Robot assist walker.
URL https://www.rtworks.co.jp/product/rt2.html

[4] Cyberdyne, Hal.
URL https://www.cyberdyne.jp/products/HAL/

[5] Daiwa House, Paro.
URL http://www.daiwahouse.co.jp/robot/paro/case/index.html

[6] Fuji Software, palro.
URL https://palro.jp/preventive-care/nursing-home.html

[7] M. T. Phan, V. Fremont, I. Thouvenin, M. Sallak, V. Cherfaoui, Recognizing driver awareness of pedestrian, in: IEEE Conference on Intelligent Transportation Systems, IEEE, 2014.

[8] T. Bar, D. Linke, D. Nienhuser, J. M. Zollner, Seen and missed traffic objects: A traffic object-specific awareness estimation, in: IEEE Intelligent Vehicles Symposium, IEEE, 2013.

[9] E. Murphy-Chutorian, M. M. Trivedi, Head pose estimation and augmented reality tracking: An integrated system and evaluation for monitoring driver awareness, IEEE Transactions on Intelligent Transportation Systems 11 (2) (2010) 300–311. doi:10.1109/TITS.2010.2044241.

[10] R. Stiefelhagen, J. Zhu, Head orientation and gaze direction in meetings, in: Extended Abstracts on Human Factors in Computing Systems, ACM, 2002.

[11] A. Doshi, M. M. Trivedi, Attention estimation by simultaneous observation of viewer and view, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops, IEEE, 2010.

[12] I. Leite, C. Martinho, A. Paiva, Social robots for long-term interaction: A survey, International Journal of Social Robotics 5 (2) (2013) 291–308. doi:10.1007/s12369-013-0178-y.

[13] B. S. B. Dewantara, J. Miura, Generation of a socially aware behavior of a guide robot using reinforcement learning, in: International Electronics Symposium, IEEE, 2016. doi:10.1109/ELECSYM.2016.7860984.

[14] 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. doi:10.1109/robio.2012.6491093.

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

[16] S. Thrun, D. Fox, W. Burgard, F. Dellaert, Robust monte carlo localization for mobile robots, Artificial Intelligence 128 (1-2) (2001) 99–141. doi:10.1016/S0004-3702(01)00069-8.

[17] I. Ardiyanto, J. Miura, Real-time navigation using randomized kinodynamic planning with arrival time field, Robotics and Autonomous Systems 60 (12) (2012) 1579–1591. doi:10.1016/j.robot.2012.09.011.

[18] Doog, Thouzer.
URL http://jp.doog-inc.com/product-thouzer.html

[19] Y. Morales, S. Satake, R. Huq, D. Glas, T. Kanda, N. Hagita, How do people walk side-by-side?; using a computational model of human behavior for a social robot, in: ACM/IEEE International Conference on Human-Robot Interaction, 2012, pp. 301–308.

[20] G. Ferrer, A. Garrell, A. Sanfeliu, Robot companion: A social-force based approach with human awareness-navigation in crowded environments, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2013, pp. 1688–1694. doi:10.1109/IROS.2013.6696576.

[21] S. Oishi, Y. Kohari, J. Miura, Toward a robotic attendant adaptively behaving according to human state, in: IEEE International Symposium on Robot and Human Interactive Communication, IEEE, 2016. doi:10.1109/ROMAN.2016.7745236.

[22] K. Koide, J. Miura, Person identification based on the matching of foot strike timings obtained by IMU and LRF, in: IROS Workshop on Assistance and Service Robotics in a Human Environment, IEEE, 2015.

[23] 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. doi:10.1016/j.robot.2016.07.004.

[24] 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. doi:10.1007/978-3-030-01370-7_15.

[25] D. Helbing, P. Molnar, Social force model for pedestrian dynamics, Physical review E 51 (5) (1995) 4282. doi:10.1103/PhysRevE.51.4282.

[26] E. Hall, The Hidden Dimension: Man's Use of Space in Public and Private, Doubleday anchor books, Bodley Head, 1969.

[27] E. Moussy, A. A. Mekonnen, G. Marion, F. Lerasle, A comparative view on exemplar tracking-by-detection approaches, in: IEEE International Conference on Advanced Video and Signal Based Surveillance, IEEE, 2015. doi:10.1109/avss.2015.7301774.

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

[29] 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. doi:10.1109/ROBOT.2007.363998.

[30] K. Kidono, T. Miyasaka, A. Watanabe, T. Naito, J. Miura, Pedestrian recognition using high-definition LIDAR, in: IEEE Intelligent Vehicles Symposium, IEEE, 2011, pp. 405–410. doi:10.1109/ivs.2011.5940433.

[31] L. Tamas, M. Popa, G. Lazea, I. Szoke, A. Majdik, Lidar and vision based people detection and tracking, Journal of Control Engineering and Applied Informatics 12 (2010) 30–35.

[32] K. Kidono, T. Naito, J. Miura, Reliable pedestrian recognition combining high-definition LIDAR and vision data, in: IEEE International Conference on Intelligent Transportation Systems, IEEE, 2012. doi:10.1109/itsc.2012.6338657.

[33] R. E. Kalman, A new approach to linear filtering and prediction problems, Journal of Basic Engineering 82 (1) (1960) 35. doi:10.1115/1.3662552.

[34] M. Luber, J. A. Stork, G. D. Tipaldi, K. O. Arras, People tracking with human motion predictions from social forces, in: IEEE International Conference on Robotics and Automation, IEEE, 2010. doi:10.1109/ROBOT.2010.5509779.

[35] P. Konstantinova, A. Udvarev, T. Semerdjiev, A study of a target tracking algorithm using global nearest neighbor approach, in: International Conference on Computer Systems and Technologies, 2003, pp. 290–295.

[36] S. H. Rezatofighi, A. Milan, Z. Zhang, Q. Shi, A. Dick, I. Reid, Joint probabilistic data association revisited, in: IEEE International Conference on Computer Vision, IEEE, 2015. doi:10.1109/iccv.2015.349.

[37] C. Kim, F. Li, A. Ciptadi, J. M. Rehg, Multiple hypothesis tracking revisited, in: IEEE International Conference on Computer Vision, IEEE, 2015. doi:10.1109/iccv.2015.533.

[38] Y. H. Kwon, N. da Vitoria Lobo, Face detection using templates, in: International Conference on Pattern Recognition, IEEE Comput. Soc. Press, 1994. doi:10.1109/icpr.1994.576435.

[39] Z. Jin, Z. Lou, J. Yang, Q. Sun, Face detection using template matching and skin-color information, Neurocomputing 70 (4-6) (2007) 794–800. doi:10.1016/j.neucom.2006.10.043.

[40] P. Viola, M. Jones, Rapid object detection using a boosted cascade of simple features, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE Comput. Soc, 2001. doi:10.1109/CVPR.2001.990517.

[41] T. Baltrusaitis, P. Robinson, L.-P. Morency, Constrained local neural fields for robust facial landmark detection in the wild, in: IEEE International Conference on Computer Vision Workshops, IEEE, 2013. doi:10.1109/iccvw.2013.54.

[42] S. Li, W. Deng, Deep facial expression recognition: A survey, arXiv preprint arXiv:1804.0834 abs/1804.08348.

[43] S. Zhang, R. Benenson, M. Omran, J. Hosang, B. Schiele, How far are we from solving pedestrian detection?, in: IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2016, pp. 1259–1267. doi:10.1109/CVPR.2016.141.

[44] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: IEEE Conference on Computer Vision and Pattern Recognition, Vol. 1, IEEE, 2005, pp. 886–893. doi:10.1109/CVPR.2005.177.

[45] P. Dollar, Z. Tu, P. Perona, S. Belongie, Integral channel features, in: British Machine Vision Conference, British Machine Vision Association, 2009. doi:10.5244/c.23.91.

[46] S. Zhang, R. Benenson, B. Schiele, Filtered channel features for pedestrian detection, in: IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2015, pp. 1751–1760. doi:10.1109/CVPR.2015.7298784.

[47] L. Bourdev, J. Brandt, Robust object detection via soft cascade, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE, 2005. doi:10.1109/cvpr.2005.310.

[48] Y. Tian, P. Luo, X. Wang, X. Tang, Pedestrian detection aided by deep learning semantic tasks, in: IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2015. doi:10.1109/cvpr.2015.7299143.

[49] J. Hosang, M. Omran, R. Benenson, B. Schiele, Taking a deeper look at pedestrians, in: IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2015. doi:10.1109/cvpr.2015.7299034.

[50] A. Angelova, A. Krizhevsky, V. Vanhoucke, A. Ogale, D. Ferguson, Real-time pedestrian detection with deep network cascades, in: British Machine Vision Conference, British Machine Vision Association, 2015. doi:10.5244/c.29.32.

[51] X. Ma, Z. Chen, J. Zhang, Fully convolutional network with cluster for semantic segmentation, in: AIP Conference Proceedings, 2018. doi:10.1063/1.5033713.

[52] L. Boominathan, S. S. S. Kruthiventi, R. V. Babu, CrowdNet, in: ACM on Multimedia Conference, ACM Press, 2016. doi:10.1145/2964284.2967300.

[53] M. Villamizar, A. Martinez-Gonzalez, O. Canevet, J.-M. Odobez, WatchNet: Efficient and depth-based network for people detection in video surveillance systems, in: IEEE International Conference on Advanced Video and Signal Based Surveillance, IEEE, 2018. doi:10.1109/AVSS.2018.8639165.

[54] Z. Cao, T. Simon, S.-E. Wei, Y. Sheikh, Realtime multi-person 2d pose estimation using part affinity fields, in: IEEE Conference on Computer Vision and Pattern Recognition, 2017.

[55] R. Stewart, M. Andriluka, A. Y. Ng, End-to-end people detection in crowded scenes, in: IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2016. doi:10.1109/cvpr.2016.255.

[56] Z. Zainudin, S. Kodagoda, G. Dissanayake, Torso detection and tracking using a 2d laser range finder, in: Australasian Conference on Robotics and Automation, ARAA, 2010.

[57] H. W. Kuhn, The hungarian method for the assignment problem, Naval Research Logistics 52 (1) (2005) 7–21. doi:10.1002/nav.20053.

[58] K. O. A. Timm Linder, Fabian Girrbach, Towards a robust people tracking framework for service robots in crowded , dynamic environments, in: IROS Workshop on Assistance and Service Robotics in a Human Environment, 2015.

[59] 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.

[60] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam, Mobilenets: Efficient convolutional neural networks for mobile vision applications, arXiv preprint arXiv:1704.04861.

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

[62] 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. doi:10.1016/j.imavis.2014.08.002.

[63] 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. doi:10.1109/asspcc.2000.882463.

[64] A. Carballo, A. Ohya, S. Yuta, Fusion of double layered multiple laser range finders for people detection from a mobile robot, in: IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems, IEEE, 2008, pp. 677–682. doi:10.1109/MFI.2008.4648023.

[65] D. Schulz, W. Burgard, D. Fox, A. B. Cremers, People tracking with mobile robots using sample-based joint probabilistic data association filters, The International Journal of Robotics Research 22 (2) (2003) 99–116. doi:10.1177/0278364903022002002.

[66] C. Cortes, V. Vapnik, Support-vector networks, Machine Learning 20 (3) (1995) 273–297. doi:10.1007/BF00994018.

[67] R. E. Schapire, Y. Singer, Improved boosting algorithms using confidence-rated predictions, Machine Learning 37 (3) (1999) 297–336. doi:10.1023/a:1007614523901.

[68] Q. Zhu, M.-C. Yeh, K.-T. Cheng, S. Avidan, Fast human detection using a cascade of histograms of oriented gradients, in: IEEE Conference on Computer Vision and Pattern Recognition, Vol. 2, IEEE, 2006, pp. 1491–1498. doi:10.1109/cvpr.2006.119.

[69] N. Bellotto, H. Hu, Multisensor data fusion for joint people tracking and identification with a service robot, in: IEEE International Conference on Robotics and Biomimetics, IEEE, 2007, pp. 1494–1499. doi:10.1109/ROBIO.2007.4522385.

[70] L. Nanni, M. Munaro, S. Ghidoni, E. Menegatti, S. Brahnam, Ensemble of different approaches for a reliable person re-identification system, Applied Computing and Informaticsdoi:10.1016/j.aci.2015.02.002.

[71] 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.

[72] 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.

[73] 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. doi:10.1109/cvpr.2015.7299016.

[74] 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. doi:10.1109/CVPRW.2017.186.

[75] 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.

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

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

[78] 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. doi:10.1109/cvpr.2006.215.

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

[80] 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.

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

[82] H. Grabner, M. Grabner, H. Bischof, Real-time tracking via on-line boosting, in: British Machine Vision Conference, British Machine Vision Association, 2006. doi:10.5244/c.20.6.

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

[84] 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: British Machine Vision Conference, British Machine Vision Association, 2015. doi:10.5244/C.29.145.

[85] S. D. Mowbray, M. S. Nixon, Automatic gait recognition via Fourier descriptors of deformable objects, in: Audio-and Video-Based Biometric Person Authentication, Springer, 2003, pp. 566–573. doi:10.1007/3-540-44887-x_67#.

[86] 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. doi:10.1007/978-3-642-19309-5{_}34.

[87] A. Hayder, J. Dargham, A. Chekima, G. M. Ervin, Person identification using gait, International Journal of Computer and Electrical Engineering 3 (4) (2011) 477–482. doi:10.7763/ijcee.2011.v3.364.

[88] C. A. Cifuentes, A. Frizera, R. Carelli, T. Bastos, Human robot interaction based on wearable IMU sensor and laser range finder, Robotics and Autonomous Systems 62 (10) (2014) 1425–1439. doi:10.1016/j.robot.2014.06.001.

[89] K. Nakamura, H. Zhao, X. Shao, R. Shibasaki, Human Sensing in Crowd Using Laser Scanners, InTech, 2012. doi:10.5772/33276.

[90] X. Song, J. Cui, H. Zhao, H. Zha, R. Shibasaki, Laser-based tracking of multiple interacting pedestrians via on-line learning, Neurocomputing 115 (2013) 92–105. doi:10.1016/j.neucom.2013.02.001.

[91] 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.

[92] D. A. Klein, D. Schulz, S. Frintrop, Boosting with a joint feature pool from different sensors, in: Computer Vision Systems, Springer, 2009, pp. 63–72. doi:10.1007/978-3-642-04667-4{_}7.

[93] A. Bedagkar-Gala, S. K. Shah, A survey of approaches and trends in person re-identification, Image and Vision Computing 32 (4) (2014) 270 – 286. doi:10.1016/j.imavis.2014.02.001.

[94] W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery, Numerical Recipes in FORTRAN: The Art of Scientific Computing, Cambridge University Press, 1992.

[95] C. BenAbdelkader, R. Cutler, L. Davis, Stride and cadence as a biometric in automatic person identification and verification, in: IEEE International Conference on Automatic Face Gesture Recognition, IEEE, 2002, pp. 372–377. doi:10.1109/afgr.2002.1004182.

[96] E. E. Stone, M. Skubic, Passive in-home measurement of stride-to-stride gait variability comparing vision and kinect sensing, in: Annual International Conference of the IEEE Engineering in Medicine and Biology Society, IEEE, 2011, pp. 6491–6494. doi:10.1109/iembs.2011.6091602.

[97] E. E. Stone, M. Skubic, Unobtrusive, continuous, in-home gait measurement using the microsoft kinect, IEEE Transactions on Biomedical Engineering 60 (10) (2013) 2925–2932. doi:10.1109/tbme.2013.2266341.

[98] B. R. Umberger, Stance and swing phase costs in human walking, Journal of The Royal Society Interface 7 (50) (2010) 1329–1340. doi:10.1098/rsif.2010.0084.

[99] Y. Cheng, Mean shift, mode seeking, and clustering, IEEE Transactions on Pattern Analysis and Machine Intelligence 17 (8) (1995) 790–799. doi:10.1109/34.400568.

[100] M. Shiomi, N. Hagita, Finding a person with a wearable acceleration sensor using a 3D position tracking system in daily environments, Advanced Robotics 29 (23) (2015) 1563–1574.

[101] T. Ikeda, H. Ishiguro, T. Miyashita, N. Hagita, Pedestrian identification by associating wearable and environmental sensors based on phase dependent correlation of human walking, Ambient Intell Human Computing 5 (5) (2013) 645–654.

[102] F. Li, C. Zhao, G. Ding, J. Gong, C. Liu, F. Zhao, A reliable and accurate indoor localization method using phone inertial sensors, in: ACM Conference on Ubiquitous Computing, ACM, 2012, pp. 421–430. doi:10.1145/2370216.2370280.

[103] A. Jimenez, F. Seco, J. Prieto, J. Guevara, Indoor pedestrian navigation using an INS/EKF framework for yaw drift reduction and a foot-mounted IMU, in: Workshop on Positioning, Navigation and Communication, IEEE, 2010, pp. 135–143. doi:10.1109/wpnc.2010.5649300.

[104] G. Ferrer, A. Sanfeliu, Proactive kinodynamic planning using the extended social force model and human motion prediction in urban environments, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2014, pp. 1730–1735. doi:10.1109/IROS.2014.6942788.

[105] T. I. T. M. D. Brscic, T. Kanda, Person position and body direction tracking in large public spaces using 3d range sensors, IEEE Transactions on Human-Machine Systems 43 (6) (2013) 522–534.

[106] D. Baltieri, R. Vezzani, R. Cucchiara, 3dpes: 3d people dataset for surveillance and forensics, in: ACM Workshop on Multimedia access to 3D Human Objects, Scottsdale, Arizona, USA, 2011, pp. 59–64.

[107] B. Benfold, I. Reid, Stable multi-target tracking in real-time surveillance video, in: IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2011, pp. 3457–3464.

[108] L. M. Fuentes, S. A. Velastin, People tracking in surveillance applications, Image and Vision Computing 24 (11) (2006) 1165 – 1171, performance Evaluation of Tracking and Surveillance. doi:10.1016/j.imavis.2005.06.006.

[109] M. Munaro, F. Basso, E. Menegatti, OpenPTrack: Open source multi-camera calibration and people tracking for RGB-d camera networks, Robotics and Autonomous Systems 75 (2016) 525–538. doi:10.1016/j.robot.2015.10.004.

[110] E. Ristani, C. Tomasi, Features for multi-target multi-camera tracking and re-identification, in: IEEE Conference on Computer Vision and Pattern Recognition, 2018.

[111] M. Munaro, A. Fossati, A. Basso, E. Menegatti, L. V. Gool, One-shot person re-identification with a consumer depth camera, in: Person Re-Identification, Springer, 2014, pp. 161–181. doi:10.1007/978-1-4471-6296-4_8.

[112] V. B. Semwal, M. Raj, G. Nandi, Biometric gait identification based on a multilayer perceptron, Robotics and Autonomous Systems 65 (2015) 65–75. doi:10.1016/j.robot.2014.11.010.

[113] T. Sabapathy, M. A. Mustapha, M. Jusoh, S. M. Salleh, P. J. Soh, Location tracking system using wearable on-body GPS antenna, in: Engineering Technology Int. Conf., Vol. 97, EDP, 2017. doi:10.1051/matecconf/20179701099.

[114] S. T. Doherty, C. J. Lemieux, C. Canally, Tracking human activity and well-being in natural environments using wearable sensors and experience sampling, Social Science & Medicine 106 (2014) 83 – 92. doi:10.1016/j.socscimed.2014.01.048.

[115] C. Escriba, J. Roux, B. Hajjine, J.-Y. Fourniols, Smart wearable active patch for elderly health prevention, in: Annual Conference on Computational Science & Computational Intelligence, Las Vegas, United States, 2018.

[116] A. Ramadhan, Wearable smart system for visually impaired people, Sensors 18 (3) (2018) 843. doi:10.3390/s18030843.

[117] X. Zhu, Q. Li, G. Chen, Apt accurate outdoor pedestrian tracking with smartphones, in: IEEE International Conference on Computer Communications, IEEE, 2013, pp. 2508–2516. doi:10.1109/INFCOM.2013.6567057.

[118] M. Kotaru, S. Katti, Position tracking for virtual reality using commodity wifi, in: IEEE Conference on Computer Vision and Pattern Recognition, 2017.

[119] E. Soltanaghaei, A. Kalyanaraman, K. Whitehouse, Multipath triangulation: Decimeter-level wifi localization and orientation with a single unaided receiver, Annual International Conference on Mobile Systemsdoi:10.1145/3210240.3210347.

[120] A. Edwards, B. Silva, R. dos Santos, G. Hancke, WiFi based indoor positioning using pattern recognition, in: IEEE International Symposium on Industrial Electronics, IEEE, 2018. doi:10.1109/isie.2018.8433869.

[121] W. Kang, Y. Han, Smartpdr: Smartphone-based pedestrian dead reckoning for indoor localization, IEEE Sensors Journal 15 (5) (2015) 2906–2916. doi:10.1109/JSEN.2014.2382568.

[122] G. Grisetti, R. Kummerle, C. Stachniss, W. Burgard, A tutorial on graph-based slam, IEEE Intelligent Transportation Systems Magazine 2 (4) (2010) 31–43. doi:10.1109/MITS.2010.939925.

[123] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, W. Burgard, G2o: A general framework for graph optimization, in: IEEE International Conference on Robotics and Automation, IEEE, 2011, pp. 3607–3613. doi:10.1109/ICRA.2011.5979949.

[124] M. Magnusson, A. Lilienthal, T. Duckett, Scan registration for autonomous mining vehicles using 3d-ndt, Journal of Field Robotics 24 (10) (2007) 803–827. doi:10.1.1.189.2393.

[125] P. J. Besl, N. D. McKay, A method for registration of 3-d shapes, IEEE Transactions on Pattern Analysis and Machine Intelligence 14 (2) (1992) 239–256. doi:10.1109/34.121791.

[126] M. Magnusson, A. Nuchter, C. Lorken, A. J. Lilienthal, J. Hertzberg, Evaluation of 3d registration reliability and speed - a comparison of icp and ndt, in: IEEE International Conference on Robotics and Automation, IEEE, 2009, pp. 3907–3912. doi:10.1109/ROBOT.2009.5152538.

[127] E. Nelson, Blam - berkeley localization and mapping.
URL https://github.com/erik-nelson/blam

[128] M. A. Fischler, R. C. Bolles, Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography, Communications 24 (6) (1981) 381–395. doi:10.1145/358669.358692.

[129] L. Ma, C. Kerl, J. Stückler, D. Cremers, Cpa-slam: Consistent plane-model alignment for direct rgb-d slam, in: IEEE International Conference on Robotics and Automation, IEEE, 2016, pp. 1285–1291. doi:10.1109/ICRA.2016.7487260.

[130] T. Shan, B. Englot, Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain, in: IEEE/RSJ International Conference on Intelligent Robots and Systems), IEEE, 2018, pp. 4758–4765. doi:10.1109/IROS.2018.8594299.

[131] M. Haselich, B. Jobgen, N. Wojke, J. Hedrich, D. Paulus, Confidence-based pedestrian tracking in unstructured environments using 3D laser distance measurements, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2014, pp. 4118–4123. doi:10.1109/iros.2014.6943142.

[132] B. Kulis, M. I. Jordan, Revisiting k-means: New algorithms via bayesian nonparametrics, International Conference on Machine Learning abs/1111.0352.

[133] WHO Media centre, Fact sheet: Dementia (2017).
URL http://www.who.int/mediacentre/factsheets/fs362/en/

[134] A. Gunawardana, M. Mahajan, A. Acero, J. C. Platt, Hidden conditional random fields for phone classification, in: International Conference on Speech Communication and Technology, International Speech Communication Association, 2005.

[135] F. Sha, F. Pereira, Shallow parsing with conditional random fields, in: Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology, ACL, 2003.

[136] A. Quattoni, M. Collins, T. Darrell, Conditional random fields for object recognition, in: Advances in neural information processing systems, 2004, pp. 1097–1104.

[137] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in: Lecture Notes in Computer Science, Springer International Publishing, 2015, pp. 234–241. doi:10.1007/978-3-319-24574-4_28.

[138] J. Yang, B. Price, S. Cohen, H. Lee, M.-H. Yang, Object contour detection with a fully convolutional encoder-decoder network, in: IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2016.

[139] IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Pets 2009 benchmark data.
URL http://www.cvg.reading.ac.uk/PETS2009/a.html

Acknowledgments

I would like to express my sincere gratitude to my supervisor Prof. Jun Miura for the continuous support of my BSc, MSc, and Ph.D study, for his expertise, understanding, and patience. I would like to thank him for his assistance in writing journals, papers, reports, and dissertation, also for giving me chances to attend prestigious conferences and take a short-stay at a foreign university. I appreciate all his advice on not only research activities but also on my career plan, which had a great influence on my life.

Besides my supervisor, I would also like to express my gratitude to Prof. Kuriyama, Prof. Kanazawa, and Prof. Kitazaki as the examiner members who kindly provide many fruitful suggestions on this thesis.

I am grateful to the former Assistant Professor Dr. Shuji Oishi and Dr. Junji Satake for supporting my research and the productive discussions. My sincere thanks also goes to Ms. Mikiko Kobayashi and Ms. Akiko Yamamoto who has helped me to prepare a lot of documents.

I thank all my lab-mates at the Advanced Intelligent Systems Laboratory: Yoshitaka Kohari, Seiichiro Une, Mitsuhiro Demura, Yutaro Chikada, Shota Tanaka, Tsubasa Kato, Motoki Kojima, Kazuho Morohashi, Masaki Hasegawa, Naoki Uzawa, Liliana Villamar Gomez, Hoai Luu Duc, Shigemichi Matsuzaki, Masanori Matsushita, Masataka Inouchi, Yasunori Kawamata, Hironori Fujimoto, Kazuki Mano for not only their cooperation on my work but also all the fun we have had in the lab. I also thank my former senior Dr. Igi Ardiyanto and Dr. Bima Sena Bayu Dewantara for having productive discussions on my research activities.

At last, I would like to express my gratitude to Prof. Emanuele Menegatti and the Leading Graduate School Program for giving me a chance to stay at the Intelligent Autonomous Systems Laboratory in the University of Padova in Italy during my Ph.D period.

List of Publications

Journals

[1] K. Koide and J. Miura, Identification of a Specific Person using Color, Height, and Gait Features for a Person Following Robot, Robotics and Autonomous Systems, Vol. 84, No. 10, pp. 76–87, Oct. 2016.

[2] K. Koide, J. Miura, and E. Menegatti, A Portable Three-dimentional LIDAR-based System for Long-term and Wide-area People Behavior Measurement, International Journal of Advanced Robotic Systems, Vol. 16, Issue 2, pp. 1–16, Apr. 2019.

Conferences

[1] K. Koide and J. Miura, Estimating Person's Awareness of an Obstacle using HCRF for an Attendant Robot, Proceedings of 4th International Conference on Human-Agent Interaction (iHAI2016), pp. 393–397, Singapore, Oct. 2016.

[2] K. Koide and J. Miura, Person Identification Based on the Matching of Foot Strike Timings Obtained by LRFs and Smartphone, Proceedings of 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS2016), Daejeon, Korea, pp. 4187–4192, Oct. 2016.

[3] K. Koide and J. Miura, Convolutional Channel Features-Based Person Identification for Person Following Robots, Proceedings of 15th International Conference on Intelligent Autonomous Systems (IAS-15), pp. 186–197, Baden-Baden, Germany, June 2018.