

A Portable 3D LIDAR-based System for Long-term and Wide-area People Behavior Measurement

Journal Title
XX(X):1–13
©The Author(s) 2018
Reprints and permission:
sagepub.co.uk/journalsPermissions.nav
DOI: 10.1177/ToBeAssigned
www.sagepub.com/

SAGE

Kenji Koide^{1,2}, Jun Miura¹, and Emanuele Menegatti²

Abstract

It is important to measure and analyze people behavior to design systems which interact with people. This paper describes a portable people behavior measurement system using a 3D-LIDAR. In this system, an observer carries the system equipped with a 3D LIDAR and follows persons to be measured while keeping them in the sensor view. The system estimates the sensor pose in a 3D environmental map and tracks the target persons. It enables long-term and wide-area people behavior measurements which are hard for existing people tracking systems. As a field test, we recorded the behavior of professional caregivers attending elderly persons with dementia in a hospital. The preliminary analysis of the behavior reveals how the caregivers decide the attending position while checking the surrounding people and environment. Based on the analysis result, empirical rules to design the behavior of attendant robots are proposed.

Keywords

3D LIDAR, People Detection and Tracking, Behavior Analysis.

Introduction

It is important to measure and analyze people behavior for designing systems which interact with people. We have to understand how people behave with respect to the surrounding people and environment to achieve systems with natural and rich interactions with people. In particular for service robots, by analyzing the behavior of a person who is helping another, we could model their behavior and create a robot with human-like behavior. This allows robots to have natural interaction with humans, and makes them more acceptable in daily service situations.

Several models which describe the social interaction between persons, such as social distance [1] and social force model [2], have been proposed, and a number of works have applied those models to service robots [3, 4, 5]. However, since those models are based on 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 [6] and outdoor environments [7, 8]. 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

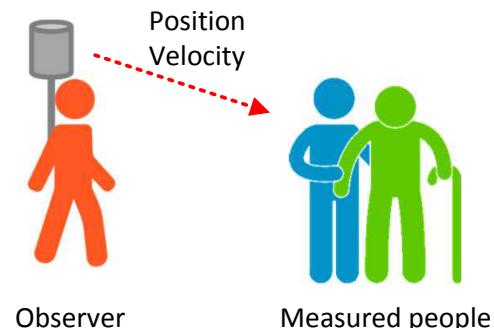


Figure 1. The proposed system to measure people behavior using a 3D LIDAR. The observer carries the backpack with a 3D LIDAR and follows the persons to be measured.

consists of real professional human's attendant behavior data.

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

¹The Department of Computer Science and Information Engineering, Toyohashi University of Technology, Japan

²The Department of Information Engineering, the University of Padova, Italy

Corresponding author:

Kenji Koide, Toyohashi University of Technology, Toyohashi, Aichi, Japan
Email: koide@aisl.cs.tut.ac.jp

The contributions of this paper are threefold. First, we propose a portable measurement system which enables long-term and wide-area people behavior measurements. We validated that the tracking accuracy of the proposed system is comparable to a static sensor-based people tracking system. Second, we provide a preliminary analysis of a field test of the proposed system in a hospital. We recorded the behavior of professional caregivers attending elderly persons with dementia. The results show that the proposed system can be applied to the measurement of real people behavior. In addition to that, based on the analysis results, we propose empirical rules to design the behavior of attendant robots. Thirdly, we provide the software of the system and the recorded people behavior as open-source and a public dataset*. They would be useful to measure and analyze people behavior in situations which are hard for existing people tracking systems.

The rest of the paper is organized as follows. The following section explains related work. The third section describes an overview of the proposed system. The fourth and fifth sections describe the offline SLAM method using a 3D LIDAR and the online people behavior measurement method which includes sensor localization and people tracking, respectively. The sixth section explains a field test in a hospital and provides a preliminary analysis of the field test. The last section concludes the paper and discusses future work.

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 [9] and tracking methods [10] using RGB cameras. Recent inexpensive consumer RGB-D cameras allow us to reliably detect and track people [11], and a camera network system for people tracking using RGB-D cameras has been proposed [12]. 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 [13, 14, 15, 16], and soft biometric features [17, 18] 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 [19, 20]. 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 [21, 22]. As an application, GPS-based wearable devices for helping elderly or visually impaired people have been proposed [23, 24]. The combination of GPS and INS improves tracking accuracy under low-level GPS radio power [25]. However, GPS signals are not available in places close to buildings and indoor environments.

Recently, WiFi signal-based localization has been widely studied [26, 27, 28]. Some of them are based on triangulation of WiFi signal strength and show decimeter or centimeter accuracy in ideal situations [26, 27]. 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 [28]. 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 [29, 30]. 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 and map-based localization [29]. 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.

System Overview

Fig. 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.

*http://github.com/koide3/hdl_graph_slam, and http://www.aisl.cs.tut.ac.jp/database_fukushima.html

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 [31]). 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 [32]. Simultaneously, the system detects and tracks the target persons.

Offline Environmental Mapping

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 [31, 33], let \mathbf{x}_k be the node k . Let \mathbf{z}_k and Ω_k be the mean and the information matrix of the constraints relating to \mathbf{x}_k . The objective function is defined as:

$$F(\mathbf{x}) = \sum e_k(\mathbf{x}_k, \mathbf{z}_k)^T \Omega_k e_k(\mathbf{x}_k, \mathbf{z}_k), \quad (1)$$

where, $e_k(\mathbf{x}_k, \mathbf{z}_k)$ is an error function between the parameters \mathbf{x}_k and the constraints \mathbf{z}_k . Typically, eq. (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 $\Delta \mathbf{x}$ on the manifold.

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

$$\mathbf{q}_k \boxplus \Delta \mathbf{q} = \left[\sqrt{1 - \|q'_x + q'_y + q'_z\|^2}, q'_x, q'_y, q'_z \right], \quad (2)$$

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

In the proposed system, we first estimate the sensor trajectory by iteratively applying NDT (Normal Distributions Transform) scan matching [34] between consecutive frames. For 3D LIDARs, NDT shows a better performance than other scan matching algorithms, such as Iterative Closest Points [35], in terms of both the reliability and the processing speed [36]. Let \mathbf{p}_t be the sensor pose at t , consisting of a translation vector \mathbf{t} and a quaternion \mathbf{q} , and $\mathbf{r}_{t, t+1}$ be the relative sensor pose between t and $t + 1$ estimated by the scan matching. We add them to the pose graph as nodes $[\mathbf{p}_0, \dots, \mathbf{p}_N]$ and edges $[\mathbf{r}_{0, 1}, \dots, \mathbf{r}_{N-1, N}]$. 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:  $\mathcal{P} = \{\mathbf{p}_0, \dots, \mathbf{p}_N\}$ , pose nodes
Input:  $\mathcal{R} = \{\mathbf{r}_{0,1}, \dots, \mathbf{r}_{N-1,N}\}$ , odometry edges
Output:  $\mathcal{L} = \{l_0, \dots, l_M\}$ , loop edges
1:  $\mathcal{L} \Leftarrow \{\}$ 
2: for  $i = 0 \dots N - 1$  do
3:    $\mathcal{C} \Leftarrow \{\}$                                  $\triangleright$  Loop candidates
4:    $accum\_d \Leftarrow 0$                           $\triangleright$  Accumulated distance
5:   for  $j = i + 1 \dots N$  do
6:      $d \Leftarrow \|\mathbf{p}_i.\mathbf{t} - \mathbf{p}_j.\mathbf{t}\|$ 
7:      $accum\_d \Leftarrow accum\_d + d$ 
8:     if  $d < th_d$  and  $accum\_d > th_a$  then
9:       Add loop candidate  $l = \{\mathbf{p}_i, \mathbf{p}_j\}$  to  $\mathcal{C}$ 
10:      end if
11:    end for
12:    for  $l = \{\mathbf{p}_i, \mathbf{p}_j\}$  in  $\mathcal{C}$  do
13:       $m \Leftarrow scan\_matching(\mathbf{p}_i, \mathbf{p}_j)$ 
14:      if  $m.score < th_s$  then
15:         $\mathcal{L} \Leftarrow \mathcal{L} \cup \{l\}$ 
16:      end if
17:    end for
18:  end for

```

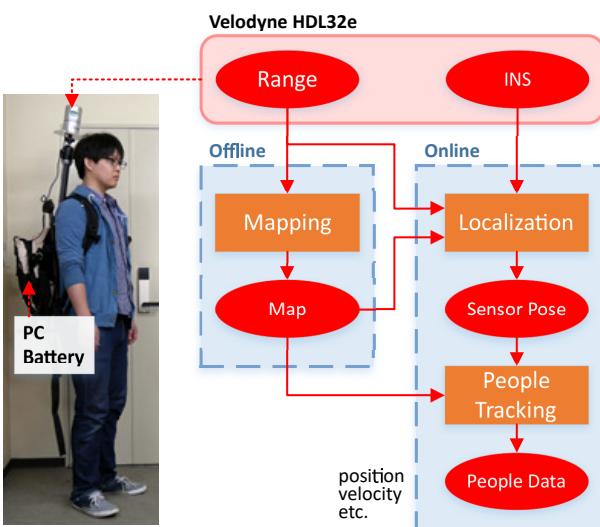


Figure 2. System overview.

The loop detection algorithm is similar to [37]. First, we detect loop candidates based on the translational distance and the length of the trajectory between nodes (Line 2 ~ 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 ~ 17). Every time a loop is found, the pose graph is updated such that eq. (1) is minimized. We utilize g2o, a general framework for hypergraph optimization [33], for the pose graph optimization.

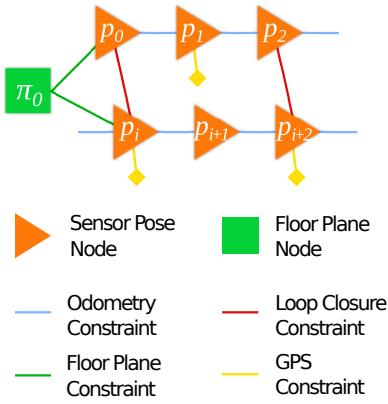


Figure 3. The proposed pose graph structure.

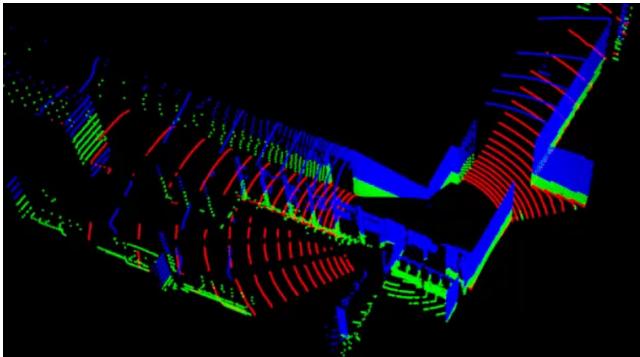


Figure 4. Ground plane detection. Points within a certain height range are extracted by height thresholding (green points), and then RANSAC is applied to them to detect the ground plane (red points). The horizontality of the ground plane is validated by checking the plane normal.

As a generated map gets larger, it tends to be bent due to the accumulated rotational error of the scan matching (see Fig. 7). In order to compensate the error, we introduce ground plane and GPS position constraints for indoor and outdoor environments, respectively. Fig. 3 shows an illustration of the graph structure of the proposed system.

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 [38] 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 shows an example of 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 p_i with the fixed ground plane node where the plane coefficients $\pi_0 = [n_x, n_y, n_z, d]^T = [0, 0, 1, 0]^T$.

To calculate the error between sensor pose p_t and the ground plane π_0 , we first transform the ground plane into the local coordinate of the sensor pose p_t :

$$[n'_x, n'_y, n'_z]^T = \mathbf{R}_t \cdot [n_x, n_y, n_z]^T, \quad (3)$$

$$d' = d - \mathbf{t}_t \cdot [n'_x, n'_y, n'_z]^T, \quad (4)$$

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

Following Ma's work [39], we employ the minimum parameterization $\tau(\pi) = (\phi, \psi, d)$, 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:

$$\tau(\pi) = \left[\arctan\left(\frac{n_y}{n_x}\right), \arctan\left(\frac{n_z}{|n|}\right), d \right], \quad (5)$$

$$e_{i,0} = \tau(\pi'_0) - \tau(\pi_t), \quad (6)$$

where π_t is the detected ground plane at t .

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 \mathbf{t}_t of a pose node p_t and a GPS position \mathbf{T}_t is simply given by:

$$e_i = \mathbf{t}_t - \mathbf{T}_t. \quad (7)$$

SLAM framework evaluation

In order to validate the proposed SLAM system, we recorded a 3D point cloud sequence in an indoor environment. Fig. 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 [37] and LeGO-LOAM [40], to this dataset.

Fig. 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



Figure 5. The experimental environment. The duration of the sequence is about 45 minutes, and the length of the trajectory is about 2400 m.

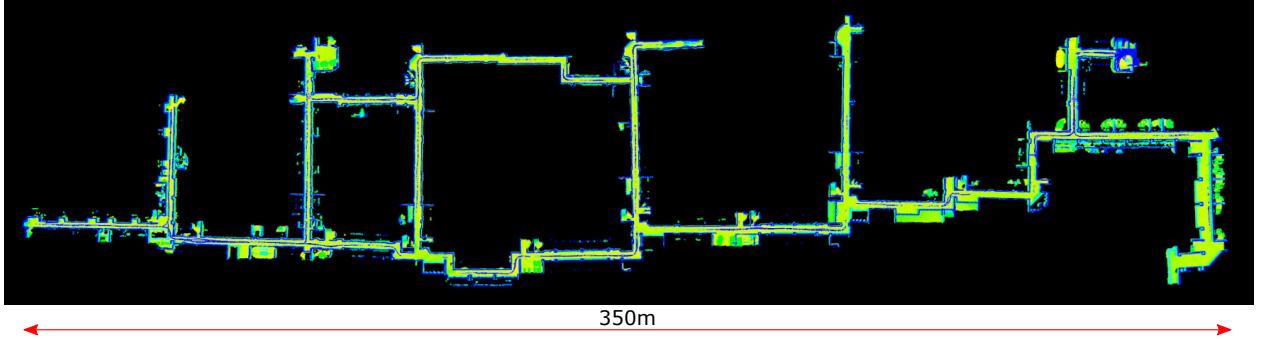


Figure 6. The created environmental map. The color indicates the height of each point. The height of the floor is consistent thanks to the plane constraint.

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. 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 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. 8 (a) shows the environment and the trajectory of the sequence. The duration of the sequence is about 42 minutes (2500 sec). Fig. 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 GPS constraint. Note that, without the GPS constraint,

Table 1. Processing time of BLAM and our SLAM system.

method	time [sec]
ours	scan matching
	floor detection
	loop closing
	total
BLAM	total

the system could not find the loop due to the scan matching error and failed to create the environmental map.

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. 9 shows an overview of the online sensor localization and people tracking system. 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.

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

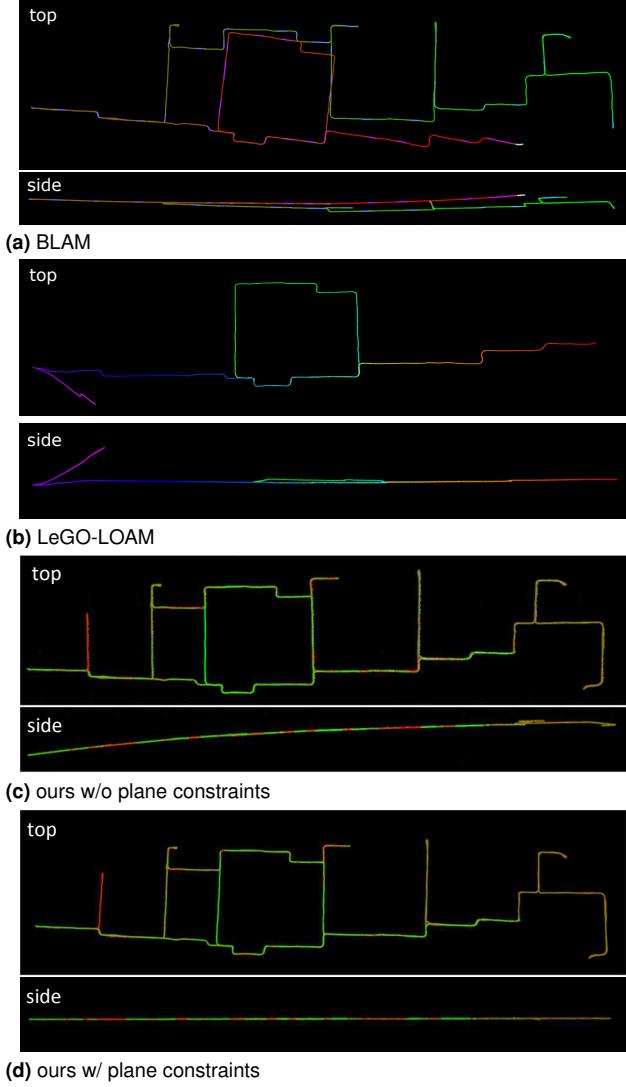


Figure 7. Comparison of the sensor trajectories estimated by the existing method and the proposed method.

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 [32].

We define the sensor state to be estimated as:

$$\mathbf{x}_t = [\mathbf{p}_t, \mathbf{q}_t, \mathbf{v}_t, \mathbf{b}_t^a]^T, \quad (8)$$

where, \mathbf{p}_t is the position, \mathbf{q}_t is the rotation quaternion, \mathbf{v}_t is the velocity, \mathbf{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:

$$\mathbf{x}_t = [\mathbf{p}_{t-1} + \Delta t \cdot \mathbf{v}_{t-1}, \mathbf{q}_{t-1} \cdot \Delta \mathbf{q}_t, \mathbf{v}_{t-1}, \mathbf{b}_{t-1}^a]^T, \quad (9)$$

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

$$\Delta \mathbf{q}_t = \left[1, \frac{\Delta t}{2} a_t^{x'}, \frac{\Delta t}{2} a_t^{y'}, \frac{\Delta t}{2} a_t^{z'} \right]^T. \quad (10)$$

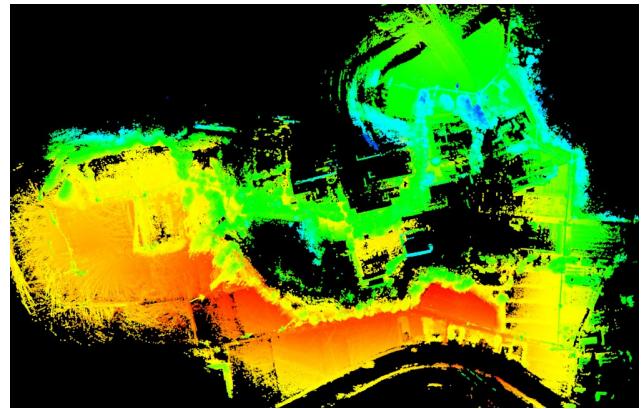


Figure 8. The SLAM system validation in an outdoor environment.

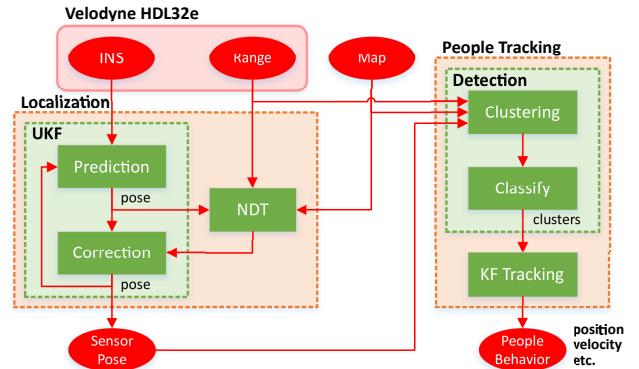
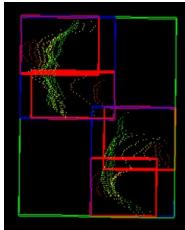


Figure 9. The online sensor pose estimation and people detection and tracking system.

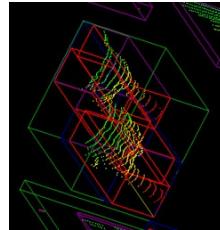
With eq. (9), the system predicts the sensor pose by using Unscented Kalman filter, and then applies NDT to match the observed point cloud with the global map with the estimated \mathbf{x}_t and \mathbf{q}_t as the initial guess of the sensor pose. Then, the system corrects the sensor state with the sensor pose estimated by the scan matching $\mathbf{z}_t = [\mathbf{p}_t', \mathbf{q}_t']^T$. The observation equation is defined as:

$$\mathbf{z}_t = [\mathbf{p}_t, \mathbf{q}_t]^T. \quad (11)$$

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



(a) Top view.



(b) Bird's eye view.

Figure 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.

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.

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 [41].

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 [42] 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. 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. [43] and Adaboost [44]. 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 [45] to track persons. The tracking scheme works well as long as the tracked persons are visible from the sensor and are correctly detected.

Sensor Localization Evaluation

To show how the pose prediction improves the sensor localization, we conducted a sensor localization experiment. Fig. 11 shows the experimental environment. An observer



Figure 11. The experimental environment of the sensor localization experiment.

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.

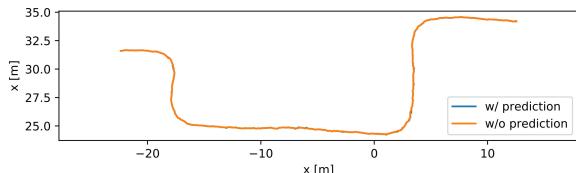
Fig. 12 shows the results of the first trial. Fig. 12 (a) shows the estimated trajectories with and without the pose prediction. Since the observer moved slowly during the first sequence, both the results show the same correct trajectory. To assess the effect of the sensor pose prediction, we assume 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. 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 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. 13. The system failed to estimate the sensor pose without the pose prediction (see. Fig. 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 2). The results show that the angular velocity-based pose prediction makes the pose estimation robust to quick motions and fast to converge.

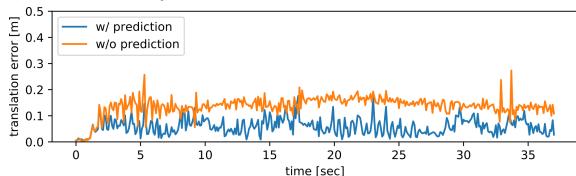
People Detection Evaluation

To analyze the effect of the split-merge clustering [41] and the human classifier [43], we recorded a 3D range data sequence, in which two persons are close together and 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.

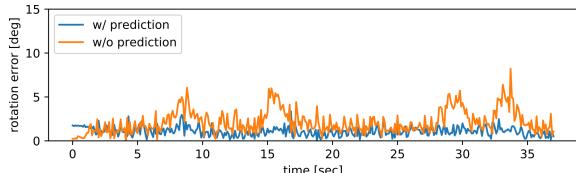
Table 3 shows the evaluation result. Without both the techniques, the recall value is low (0.834), since clusters of



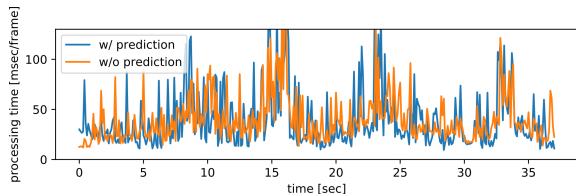
(a) Estimated trajectories.



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



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



(d) Processing time.

Figure 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.

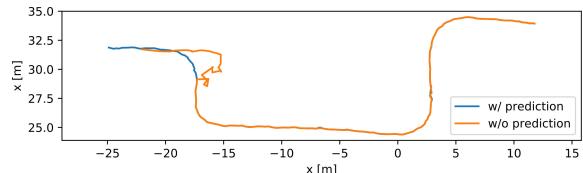
Table 2. The summary of the sensor localization experiment.

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

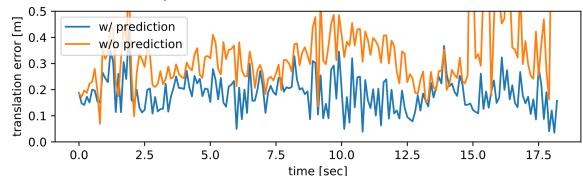
the persons are sometimes detected as a single cluster due to the Euclidean clustering. With the split-merge clustering, the wrongly merged clusters are split into sub-clusters, and the recall value gets higher (0.995). With both the split-merge clustering and the human classifier, over split sub-clusters 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 [41] effectively increases the recall of human detection, and by combining it with the human classifier [43], we can obtain reliable human detection results.

Comparison with a Static Sensor-based People Tracking System

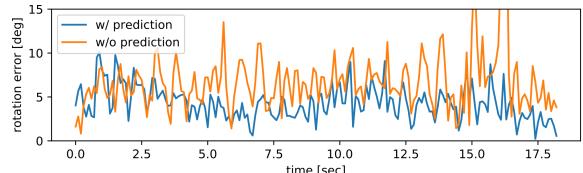
In order to reveal the pros and cons of the proposed system, we compared the proposed system with a publicly available static sensor-based people tracking framework, OpenPTrack [12]. The framework is designed for people tracking using



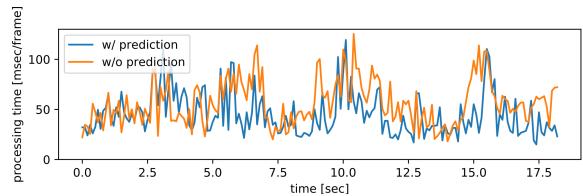
(a) Estimated trajectories.



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



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



(d) Processing time.

Figure 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.

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.

Fig. 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 $2\text{m} \times 20\text{m}$ 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 Kinetics 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 shows the summary of the differences between the people positions measured by the proposed system and OpenPTrack. The differences

Table 3. The people detection evaluation result.

Split-merge Clustering [41]	Human Classifier [43]	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

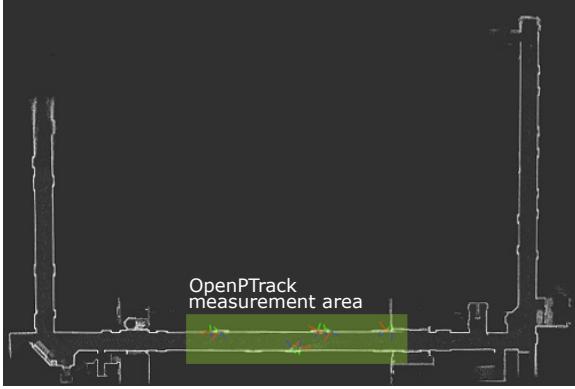


Figure 14. The experimental environment and the configuration of RGB-D cameras for OpenPTrack. Nine Kinect v2's are placed in the corridor. While OpenPTrack can measure only the limited area covered by cameras (about $2m \times 20m$ area), the proposed system can cover the whole of the floor.

Table 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

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. 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.

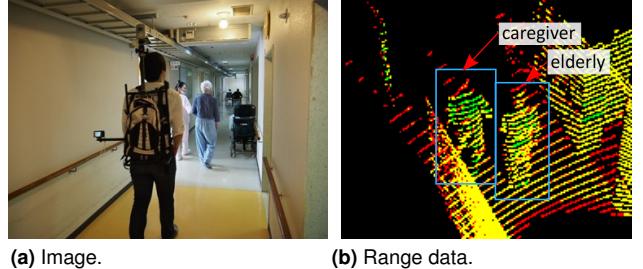


Figure 15. A snapshot of the field test. The behavior of the care giver attending an elderly is recorded by using the proposed system.

Field Test in a Hospital

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 Fukushima 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. 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. 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. 15 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. 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 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.



Figure 16. The environments of the field test.

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.

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. 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 [1] 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. 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.

Fig. 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.

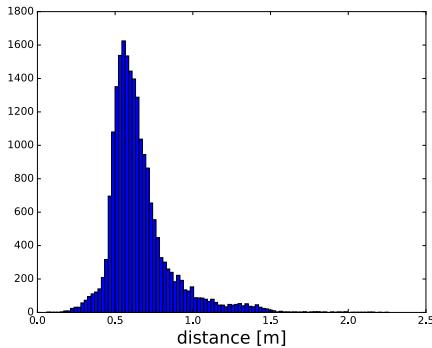
Fig. 19 (a) shows the recorded trajectories in the outdoor environment. In this sequence, the elderly was fine to walk, and the caregiver did let him walk relatively freely while navigating him to return back to the hospital. Fig. 19 (b) shows the caregiver’s walking speed and the elevation of her position in the global map. When the caregiver (and the elderly) was going up a slope, they got slow down to 1.0 ~ 1.2 m/sec while they walked at 1.2 ~ 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 of the caregiver’s relative position with respect to the elderly (see Fig. 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. 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.

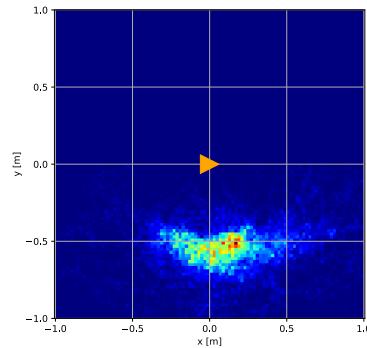
Person Following Behavior Rules

Based on the analysis of the real caregivers’ behavior, we propose empirical rules to design the behavior of attendant robots. It would be helpful to design a robot which attends a person while keeping him/her away from dangerous situations.

- 1) The robot attends the person while keeping the side-by-side positioning as long as it’s possible. In particular, it should keep in the position 0.6m aside from the person.

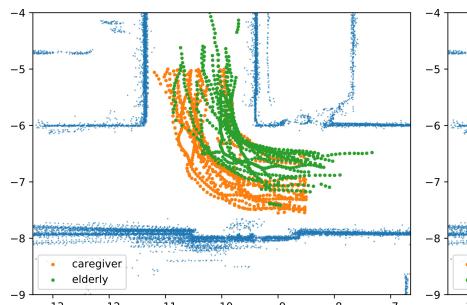


(a) The distribution of the distance between the elderly person and the care giver.

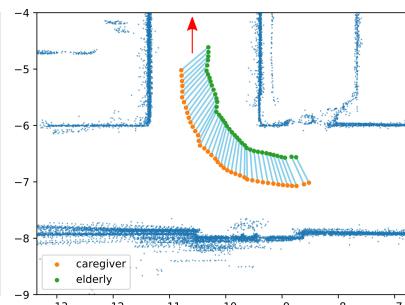


(b) The distribution of the relative position of the care giver with respect to the elderly person.

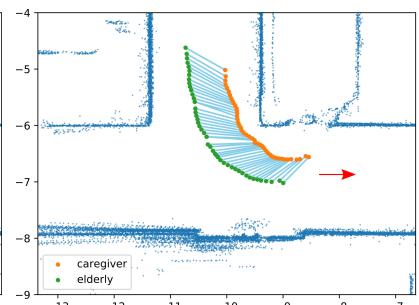
Figure 17. An analysis of the people attending behavior during the field test in an indoor environment.



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

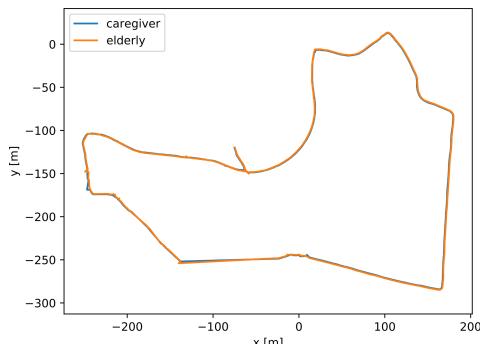


(b) An example of the cases where the caregiver walks on the outer side of the corner.

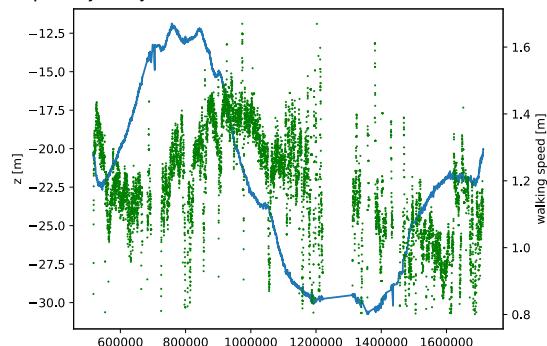


(c) The case where the caregiver walks on the inner side of the corner.

Figure 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 corridor (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).



(a) People trajectory.



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

Figure 19. The recorded attendant behavior in the outdoor environment.

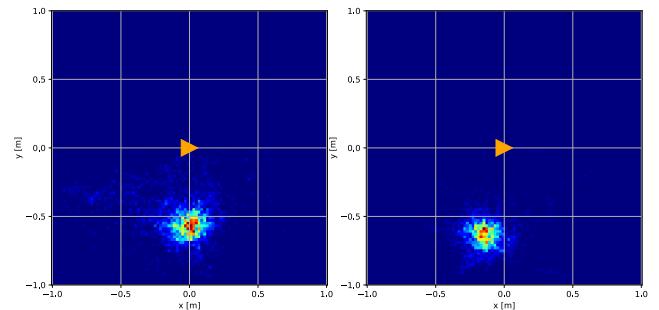


Figure 20. The distribution of the relative position of the care giver with respect to the elderly person in an outdoor environment.

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 to disturb the person.

4) In case the robot cannot go on the outer-side due to positioning and 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.

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). However, we believe that the rules would be a good initial guide to designing a comfortable attendant robot which is socially acceptable.

Conclusions and Discussion

This paper has described a portable people behavior measurement system using a 3D LIDAR. The proposed system enables long-term and wide-area behavior measurement. The system first creates a 3D map of the environment using the graph slam approach in advance to measurements. Then, it estimates its pose, detects and tracks people simultaneously. The tracking accuracy of the system is comparable to a static sensor-based people tracking system. As a field test, we demonstrated the effectiveness of the proposed system in measuring the behavior of professional caregivers' attending elderly persons. Based on the analysis of the measured behavior, empirical rules to design the behavior of attendant robots are proposed. The measurement system and the professional caregivers' behavior dataset have been public so that they can be used for to measurement and analysis of people attendant behavior.

The current system requires a human observer who carries the backpack with the 3D LIDAR, thus manual effort to observe people is necessary. The human observer would be replaced with a mobile robot so that a large attendant behavior dataset is automatically created for improving the robot attendant behavior.

Acknowledgements

This work is in part supported by JSPS Kakenhi No.25280093, and the Leading Graduate School Program R03 of MEXT.

We would like to thank O. Kohashi, S. Yamamoto, and T. Gomyo for allowing us to conduct the field test in Sawarabikai Fukushima hospital and their excellent cooperation during the test.

References

- [1] Hall E. *The Hidden Dimension: Man's Use of Space in Public and Private*. Doubleday anchor books, Bodley Head, 1969. ISBN 9780370013084.
- [2] Helbing D and Molnar P. Social force model for pedestrian dynamics. *Physical review E* 1995; 51(5): 4282. DOI: 10.1103/PhysRevE.51.4282.
- [3] Ferrer G, Garrell A and Sanfeliu A. 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, pp. 1688–1694. DOI:10.1109/IROS.2013.6696576.
- [4] Ferrer G and Sanfeliu A. 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*. pp. 1730–1735. DOI:10.1109/IROS.2014.6942788.
- [5] Oishi S, Kohari Y and Miura J. Toward a robotic attendant adaptively behaving according to human state. In *IEEE International Symposium on Robot and Human Interactive Communication*. IEEE, pp. 1038–1043. DOI:10.1109/ROMAN.2016.7745236.
- [6] Brscic D, Kanda T, Ikeda T et al. Person position and body direction tracking in large public spaces using 3d range sensors. *IEEE Transactions on Human-Machine Systems* 2013; 43(6): 522–534.
- [7] Baltieri D, Vezzani R and Cucchiara R. 3dpes: 3d people dataset for surveillance and forensics. In *ACM Workshop on Multimedia access to 3D Human Objects*. Scottsdale, Arizona, USA, pp. 59–64.
- [8] Benfold B and Reid I. Stable multi-target tracking in real-time surveillance video. In *IEEE Conference on Computer Vision and Pattern Recognition*. pp. 3457–3464.
- [9] Zhang S, Benenson R, Omran M et al. How far are we from solving pedestrian detection? In *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 1259–1267. DOI:10.1109/CVPR.2016.141.
- [10] Fuentes LM and Velastin SA. People tracking in surveillance applications. *Image and Vision Computing* 2006; 24(11): 1165 – 1171. DOI:10.1016/j.imavis.2005.06.006. Performance Evaluation of Tracking and Surveillance.
- [11] Luber M, Spinello L and Arras KO. People tracking in RGB-D data with on-line boosted target models. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, pp. 3844–3849. DOI:10.1109/IROS.2011.6095075.
- [12] Munaro M, Basso F and Menegatti E. OpenPTTrack: Open source multi-camera calibration and people tracking for RGB-d camera networks. *Robotics and Autonomous Systems* 2016; 75: 525–538. DOI:10.1016/j.robot.2015.10.004.
- [13] Bedagkar-Gala A and Shah SK. A survey of approaches and trends in person re-identification. *Image and Vision Computing* 2014; 32(4): 270 – 286. DOI:10.1016/j.imavis.2014.02.001.
- [14] Satake J, Chiba M and Miura J. 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, pp. 962–967. DOI:10.1109/ROBIO.2012.6491093.
- [15] Koide K and Miura J. Identification of a specific person using color, height, and gait features for a person following robot. *Robotics and Autonomous Systems* 2016; 84: 76–87. DOI: 10.1016/j.robot.2016.07.004.
- [16] Ristani E and Tomasi C. Features for multi-target multi-camera tracking and re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition*.
- [17] Munaro M, Fossati A, Basso A et al. 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.
- [18] Semwal VB, Raj M and Nandi G. Biometric gait identification based on a multilayer perceptron. *Robotics and Autonomous Systems* 2014; DOI:10.1016/j.robot.2014.11.010.
- [19] Song X, Cui J, Zhao H et al. Laser-based tracking of multiple interacting pedestrians via on-line learning. *Neurocomputing* 2013; 115: 92–105. DOI:10.1016/j.neucom.2013.02.001.
- [20] Nakamura K, Zhao H, Shibasaki R et al. *Human sensing in crowd using laser scanners*. INTECH Open Access Publisher, 2012. DOI:10.5772/33276.

- [21] Sabapathy T, Mustapha MA, Jusoh M et al. Location tracking system using wearable on-body GPS antenna. In *Engineering Technology Int. Conf.*, volume 97. EDP. DOI: 10.1051/matecconf/20179701099.
- [22] Doherty ST, Lemieux CJ and Canally C. Tracking human activity and well-being in natural environments using wearable sensors and experience sampling. *Social Science & Medicine* 2014; 106: 83 – 92. DOI:10.1016/j.socscimed.2014.01.048.
- [23] Escriba C, Roux J, Hajjine B et al. Smart wearable active patch for elderly health prevention. In *5th Annual Conference on Computational Science & Computational Intelligence*. Las Vegas, United States.
- [24] Ramadhan A. Wearable smart system for visually impaired people. *Sensors* 2018; 18(3): 843. DOI:10.3390/s18030843.
- [25] Zhu X, Li Q and Chen G. Apt accurate outdoor pedestrian tracking with smartphones. In *Proceedings IEEE INFOCOM*. IEEE, pp. 2508–2516. DOI:10.1109/INFCOM.2013.6567057.
- [26] Kotaru M and Katti S. Position tracking for virtual reality using commodity wifi. In *IEEE Conference on Computer Vision and Pattern Recognition*.
- [27] Soltanaghaei E, Kalyanaraman A and Whitehouse K. Multipath triangulation: Decimeter-level wifi localization and orientation with a single unaided receiver 2018; DOI:10.1145/3210240.3210347.
- [28] Edwards A, Silva B, dos Santos R et al. WiFi based indoor positioning using pattern recognition. In *IEEE 27th International Symposium on Industrial Electronics*. IEEE. DOI:10.1109/isie.2018.8433869.
- [29] Li F, Zhao C, Ding G et al. A reliable and accurate indoor localization method using phone inertial sensors. In *ACM Conference on Ubiquitous Computing*. ACM, pp. 421–430. DOI:10.1145/2370216.2370280.
- [30] Kang W and Han Y. Smartpdr: Smartphone-based pedestrian dead reckoning for indoor localization. *IEEE Sensors Journal* 2015; 15(5): 2906–2916. DOI:10.1109/JSEN.2014.2382568.
- [31] Grisetti G, Kummerle R, Stachniss C et al. A tutorial on graph-based slam. *IEEE Intelligent Transportation Systems Magazine* 2010; 2(4): 31–43. DOI:10.1109/MITS.2010.939925.
- [32] Wan E and Merwe RVD. The unscented Kalman filter for nonlinear estimation. In *Adaptive Systems for Signal Processing, Communications, and Control Symposium*. IEEE. DOI:10.1109/asspcc.2000.882463.
- [33] Kummerle R, Grisetti G, Strasdat H et al. G2o: A general framework for graph optimization. In *IEEE International Conference on Robotics and Automation*. IEEE, pp. 3607–3613. DOI:10.1109/ICRA.2011.5979949.
- [34] Magnusson M, Lilienthal A and Duckett T. Scan registration for autonomous mining vehicles using 3d-ndt. *Journal of Field Robotics* 2007; 24(10): 803–827. DOI:10.1.1.189.2393.
- [35] Besl PJ and McKay ND. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 1992; 14(2): 239–256. DOI:10.1109/34.121791.
- [36] Magnusson M, Nuchter A, Lorken C et al. Evaluation of 3d registration reliability and speed - a comparison of icp and ndt. In *IEEE International Conference on Robotics and Automation*. IEEE, pp. 3907–3912. DOI:10.1109/ROBOT.2009.5152538.
- [37] Nelson E. Blam - berkeley localization and mapping, 2016. URL <https://github.com/erik-nelson/blam>.
- [38] Fischler MA and Bolles RC. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications* 1981; 24(6): 381–395. DOI:10.1145/358669.358692.
- [39] Ma L, Kerl C, Stckler J et al. Cpa-slam: Consistent plane-model alignment for direct rgb-d slam. In *IEEE International Conference on Robotics and Automation*. IEEE, pp. 1285–1291. DOI:10.1109/ICRA.2016.7487260.
- [40] Shan T and Englot B. Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, pp. 4758–4765. DOI:10.1109/IROS.2018.8594299.
- [41] Haselich M, Jobgen B, Wojke N et al. Confidence-based pedestrian tracking in unstructured environments using 3D laser distance measurements. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, pp. 4118–4123. DOI:10.1109/iros.2014.6943142.
- [42] Kulis B and Jordan MI. Revisiting k-means: New algorithms via bayesian nonparametrics. *CoRR* 2011; abs/1111.0352.
- [43] Kidono K, Miyasaka T, Watanabe A et al. Pedestrian recognition using high-definition LIDAR. In *IEEE Intelligent Vehicles Symp. (IV)*. IEEE, pp. 405–410. DOI:10.1109/ivs.2011.5940433.
- [44] Schapire RE and Singer Y. Improved boosting algorithms using confidence-rated predictions. In *Annual Conference on Computational learning theory*, volume 37. ACM, pp. 297–336. DOI:10.1145/279943.279960.
- [45] Radosavljevic Z. A study of a target tracking method using global nearest neighbor algorithm. *Vojnotehnicki glasnik* 2006; (2): 160–167. DOI:10.5937/vojtehg0602160r.