Person Tracking in Three-Dimensional Laser Range Data with Explicit Occlusion Adaption

Florian Schöler, Jens Behley, Volker Steinhage, Dirk Schulz and Armin B. Cremers

***Abstract*— This paper presents an approach to exploit the richer information of sensor data provided by 3d laser rangefinders for the purpose of person tracking. Introduced is a method to adapt the observation model of a particle filter, to identify partial and full occlusions of a person, to determine the amount of occlusion behind an obstacle, and the occluding obstacle itself. This is done by tracing rays from positions near the person to the sensor and determining whether the ray hits an obstacle. The laser range data is represented using a voxel grid, which facilitates efficient retrieval and data reduction. As our experiments show, our proposed tracking approach is able to reliably keep track of a person in real-time, even when only partially visible, when moving in uneven terrain, or when the person passes closely another person of different size.**

I. Introduction

Person tracking is a key technology not only for autonomous systems operating in populated environments, but it can also be used in various other fields of application. For instance, driver assistance systems could detect and estimate future positions of pedestrians and issue a warning or even actively perform maneuver corrections. Also, for service robots it is indispensable to detect, track and react to humans in their vicinity. Furthermore, surveillance systems for public places, museums, etc. can be thought of, based on a tracking system. In order to avoid collisions with humans in populated environments simple collision avoidance algorithms may be sufficient to trigger emergency breaks as humans are detected in the immediate vicinity of the robot. However, a robust tracking system provides the whole path of people and robots can incorporate this additional information in their global motion planning. This enables a dynamic and more efficient path replanning without stopping.

In this paper, we present a robust, particle filter based tracking system that uses 3d range data from a 3d laser rangefinder instead of 2d range data. Two-dimensional laser range data represent only a slice of the environment. This reduced perspective on things has the effect that relevant objects, esp. the tracked target, and irrelevant objects can be erroneously confused. The used 3d laser range data in our approach offers far more comprehensive information so that the whole body of a person can be detected, which leads to a better discrimination between relevant and irrelevant objects and a better data association based thereon.

F. Schöler, J. Behley, V. Steinhage, A. B. Cremers are with the Department of Computer Science III, University of Bonn, 53117 Bonn, Germany.
D. Schulz is with the Unmanned Systems Department, Fraunhofer FKIE, 53343 Wachtberg, Germany.
{schoele,behley,steinhag,abc}@iai.uni-bonn.de, dirk.schulz@fkie.fraunhofer.de

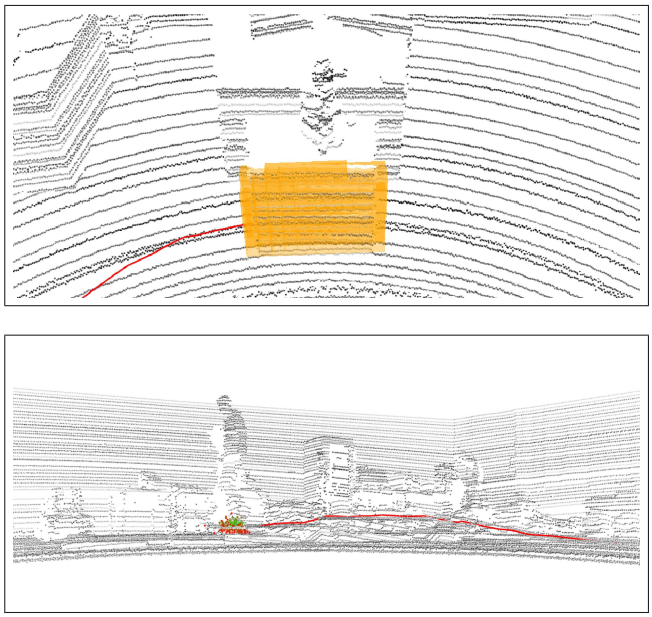


Fig. 1. The tracking approach using 3d laser range scans in different scenarios. The upper figure shows a scene with partial occlusion by a detected obstacle (orange). The observation model is adapted accordingly. The lower figure depicts the estimated trajectory in an uneven environment.

This work focuses on three aspects of the observation model of the particle filter: (1) By means of a novel occlusion detection based on ray tracing we can explicitly detect when a person is occluded, to what extent it is occluded, and detect the occluding obstacles. This knowledge is used to adapt the observation model in situations of partial or full occlusions. (2) By the use of a penalizing term we can distinguish the tracked person from another person of different size and continue the track without much confusion, even if both persons occlude each other. (3) Based on a three-dimensional motion model in combination with the observation model we can track people in uneven terrain.

The paper is organized as follows. In section II we briefly summarize related work. Then, in section III, a general overview outlines the basic ideas of our tracking approach. The upfollowing sections describe the main parts in more detail and present our adaptive observation model. Starting with section IV, we introduce the underlying data structure which is the foundation of our adaptive observation model. In section V the adaptive observation model for handling partial and full occlusions is thoroughly explained. Next, in section VI we experimentally evaluate our approach in different scenarios. Finally, in section VII we conclude and discuss future work.

II. Related Work

Over the years several tracking approaches were proposed using particle filters and 2d laser range sensors. However, recently more and more approaches also use 3d laser range data to improve person detection and object tracking. The main disadvantages of using a 2d laserscanner are the fixed height of the sensor and that only a slice of the environment is observable. A common practice is to mount the laserscanner in the height of a person's tibia, such that the movements of the legs can be detected. But this may lead to a loss of the tracked person in uneven terrain or the confusion of different persons.

3d laser rangefinders can be applied to overcome these shortcomings, as shown by the following approaches. Navarro-Serment *et al.* [1], [2] used 3d laser range data to improve the robustness of a 2d tracker by a 3d detection of humans, and extracting a so-called virtual scan slice from the 3d laser range data using an elevation map. A similar approach was applied by Petrovskaya and Thrun [3] to track vehicles. They also extracted a virtual 2d scan from the 3d data which allows them to use an existing 2d particle filter-based tracker to track vehicles. But these approaches still cannot handle occlusions properly.

Recently some approaches to detect people in 3d laser scans were proposed. Mozos *et al.* [4] used multiple 2d laser scanners mounted in different fixed heights to detect the legs, the upper body and the head of people in cluttered environments using an AdaBoost-based classification, which leads to better tracking results even during occlusions. On the other hand one is constrained to persons of specific measures because of the fixed mounting heights. Spinello *et al.* [5] also used AdaBoost to learn a model of humans, but they used actual 3d laser range scans from a Velodyne HDL 64-E laser range scanner.

Several approaches were proposed to handle occlusions explicitly. In the approach of Arras *et al.* [6], a multi-hypothesis leg tracker was presented, which explicitly models occlusions. Besides the usual labels of a track (*detection* and *deletion*) a third label for *occlusion* of tracks was introduced. Due to the tracking of separate legs this also includes self-occlusions by moving legs. Using this approach a significant reduction of false-positive initializations of tracks was achieved. Taylor and Kleeman [7] handled self-occlusions by a geometric model of the person. The scan shadow in respect to the front and rear leg indicates when it is more likely to detect both legs. In contrast to these approaches, we do not need to rely only on moving legs in the sensor data stream or have to analyze the self occlusion of the legs. We can take the whole body of the person as an observation and modify our observation model to account for partial and full occlusions. To the best of our knowledge, this is the first work on explicit occlusion handling in 3d laser range data.

III. The Overall Tracking Approach

The use of 3d laser range data for person tracking has the huge advantage that the full information about the 3d geometric structure of the robot's environment is perceived and can be utilized during tracking. However, this advantage comes at the expense of a vastly increased number of individual range measurements per second that needs to be processed online. For this purpose our approach maintains a regular grid of voxels to store the 3d points corresponding to the measurements and also to estimate the occupancy state of regions in the environment. In comparison to kd-trees [8] or octrees [9], which are frequently used in 3d mapping for this purpose, the grid-based representation allows to insert and query laser points in constant time.

Based on the voxel grid, our approach maintains a particle filter to track individual objects. In our current implementation we assume that the person's dynamics is governed by a simple three dimensional constant velocity model, i.e.. the person's state at time t is described by the 6-dimensional vector $\mathbf{x}_t = (x, y, z, v_x, v_y, v_z)_t$, where $\mathbf{p}_t = (x, y, z)_t$ denotes the position and $\mathbf{v}_t = (v_x, v_y, v_z)_t$ the velocity of the person at time t. The particle filter represents the person's state density by a set of weighted samples $S_t = \{s^{(i)}\}_{1 \leq i \leq N}$, $s_i = (\mathbf{x}_t^{(i)}, w_t^{(i)})$. Based on the constant velocity model, the prediction step of the particle filter is implemented as

$$(x, y, z)_t^{(i)} = (x, y, z)_{t-1}^{(i)} + (v_x, v_y, v_z)_{t-1}^{(i)}. \tag{1}$$

The predicted velocity of a position hypothesis is updated according to

$$(v_x, v_y, v_z)_t^{(i)} = (v_x, v_y, v_z)_{t-1}^{(i)} + (\epsilon_x, \epsilon_y, \epsilon_z) \tag{2}$$

where $\epsilon_x \sim \mathcal{N}(0, \sigma_x^2)$, $\epsilon_y \sim \mathcal{N}(0, \sigma_y^2)$, and $\epsilon_z \sim \mathcal{N}(0, \sigma_z^2)$ are normally distributed noise terms that allow to adapt to velocity changes of the person.

The position estimate of the particle filter is the weighted average position of the particles,

$$\hat{\mathbf{p}}_t = \sum_{i=1}^{N} w_t^{(i)} \cdot \mathbf{p}_t^{(i)} \tag{3}$$

The main innovation of our tracking approach lies in the design and the evaluation of the probabilistic observation model $P(z_t \,|\, x_t^{(i)})$ used to update the particle filter based on the voxel grid computed from the 3d laser range scans. In the subsequent sections, we will describe this observation model in detail; we will explain how occupied voxels originating from a moving person can be extracted fast, how partial and full occlusions can be taken into account, and also how size information about the person being tracked can be utilized during tracking.

As the focus of this work is on this observation model, we restricted our current implementation to tracking single objects from a stationary platform. However, in conjunction with an appropriate data association algorithm it should be easily generalizable to tracking multiple objects, e.g., to a 3d version of the SJPDAF approach [10].

IV. Representation of the Environment

As mentioned before, we represent the environment by a regular grid of voxels, which is centered at the sensor's

position. A voxel is denoted by $v_{i,j,k}$, where i, j, and k are the indices of that voxel in the three-dimensional grid. From the Cartesian coordinates (x, y, z) of a laser point in the local frame of the laser range sensor, the indices of the surrounding voxel are given by $i = \lfloor x\rho^{-1} \rfloor$, $j = \lfloor y\rho^{-1} \rfloor$, and $k = \lfloor z\rho^{-1} \rfloor$, respectively, where ρ is the side length of a voxel.

The grid is furthermore used as a basis for a three-dimensional extension of the well-known 2d occupancy grid map [11]. For the occupancy update, every voxel is visited and its occupancy value $P(\text{occupied}_{i,j,k})$ is increased as laser beams end in that voxel, i.e., the end points of the beams are inside the voxel. On the other hand it is decreased if no laser beam ends in that voxel. So we are using an "end point occupancy" instead of a "beam occupancy", which would also increase or decrease the occupancy of voxels along the laser beam [11]. The occupied voxels are later used to locally determine occlusions explicitly, as needed. As we will see, locally calculating the occlusions is far more efficient in 3d laser range scans than implicitly determining them as in the "beam occupancy" model usually employed when 2d laser range scans are used.

Instead of creating a new grid for every scan, we maintain only a single grid and update it in every iteration. Furthermore we store lists of voxels containing laser points that are occupied, or are occluded by an obstacle, respectively. Using these lists only a fraction of all voxels must be traversed to re-initialize and update the grid as a new scan is available.

As we will see later, by using these lists as well as the efficient Digital Differential Algorithm (DDA) ray tracing algorithm [12], [13], a complete iteration – insertion of laser points, occupancy update, occlusion calculations and the particle filter update – can be done in real-time.

The particle filter is initialized at positions, where dynamics are detected, since we currently assume that only persons move in the environment. The basis for this person detection is the aforementioned occupancy voxel grid. Its underlying idea is as follows: The occupancy grid map captures the static structures in the environment. If there is a moving entity the subset of the 3d point cloud that represents the person does not cause any voxel to be occupied. Thus, if we take all voxels that contain points but are not marked as occupied, we get an initial hypothesis for the position of a person. Due to the noise in the laser range measurements, we have to post process the results of this approach. Currently, we analyze the number of connected hypothesis voxels and apply a threshold on that number to filter false positive detections.

V. Adaptive Observation Model

The observation model $P(z_t \mid x_t^{(i)})$ expects at any time step a certain number of points $\overline{z}$ that fall on the person. To weight the particles, we assume this number is normally distributed, hence,

$$P(z_t \,|\, x_t^{(i)}) = \mathcal{N}\left(z_t;\ \overline{z},\ \sigma_{\overline{z}}^2\right), \tag{4}$$

where z_t is the number of points near a particle, $\overline{z}$ is the expected number of points and $\sigma_{\overline{z}}^2$ the variance. This way, a particle gets the highest weight, if there are as many points near it as expected and lower weights otherwise. This number is determined by counting the points in a bounding box around the particles position which is defined as

$$\mathcal{B}_r((x, y, z)_t^{(i)},\, h) = \{ v_{i,j,k} \mid \left| i - \left\lfloor x\rho^{-1} \right\rfloor \right| \leq r \quad \wedge \tag{5}$$

$$\left| j - \left\lfloor y\rho^{-1} \right\rfloor \right| \leq r \quad \wedge \tag{6}$$

$$0 \leq k - \left\lfloor z\rho^{-1} \right\rfloor \leq h \,\}, \tag{7}$$

where h is the height of the box and r its radius. Figure 2 sketches the bounding box in the voxel grid. As the person moves behind obstacles, some of the voxel can be occupied, and some are also in front of the obstacle. These voxels are not included in the bounding box, since these voxels are very likely to not contain laser end points of the person. Especially when calculating the occlusion these voxels would cause an overestimated amount of non-occluded voxels.

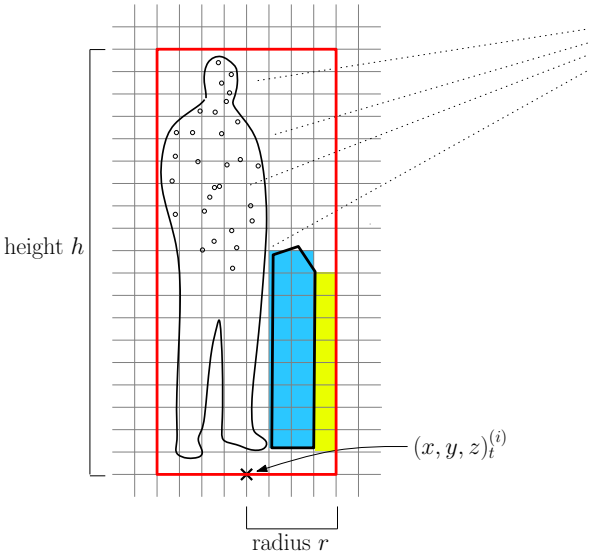
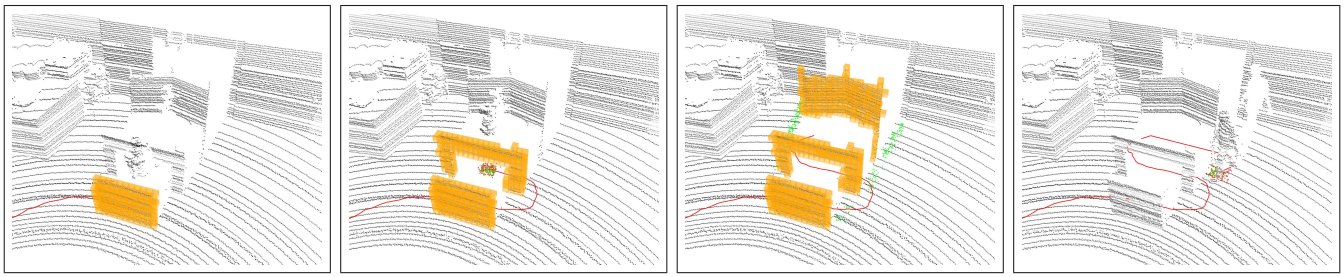


Fig. 2. Bounding box (red) at a given particle position $(x, y, z)_t^{(i)}$ with radius r and height h. As the person is partially occluded by an obstacle only some laser beams are reflected by the person. Occupied voxels (blue) by obstacles and also voxels in front of obstacles (yellow) are not included in the bounding box and are therefore not taken into account for the occlusion calculation or the point count. (Best viewed in color.)

Since we want to track people in natural environments using a 3d laser range scanner, we have to handle different types of occlusions: partial and full occlusions. In the following, we will use $\text{occluded}_{i,j,k}$ and $\neg\text{occluded}_{i,j,k}$ to distinguish voxels $v_{i,j,k}$ that are occluded or non-occluded, respectively. Figure 3 shows the results in a scene with partial and full occlusions.

A. Partial Occlusions

As a person enters some space of the environment that is partially occluded by an obstacle, the number of points that fall on the person decreases. Thus, the aforementioned observation model assigns similar weights to all particles and a divergence of the particles occurs. To avoid this behavior and to get an accurate position estimate in partially occluded spaces, we explicitly account for this and adapt $\overline{z}$ accordingly.



(a) (b) (c) (d)

Fig. 3. Results with partial and full occlusions. The person walks around obstacles of different heights. The sensor model of the particle filter is adapted in the case of partial occlusions and the occluding obstacles are identified. As the person moves behind the first obstacle in (a) the expected amount of laser points is decreased according to the amount of occluded voxels behind the obstacle. The same happens behind the second obstacle in (b), where the particles are visible and show a concentrated shape around the person. As the person steps behind the largest obstacle in (c), the full occlusion is detected and the particles are uniformly distributed at the occlusion boundaries of the obstacles. When the person leaves the full occlusion the particles are concentrated again on the person, as depicted in (d) and the usual tracking is continued.

First, we determine the occlusion ratio ω for a particle position $(x, y, z)_t^{(i)}$ using the bounding box $\mathcal{B}_r((x, y, z)_t^{(i)}, h)$, which is defined as follows:

$$\omega = \frac{\left|\left\{\text{occluded}_{i,j,k} \;\middle|\; v_{i,j,k} \in \mathcal{B}_r((x, y, z)_t^{(i)}, h)\right\}\right|}{\left|\mathcal{B}_r((x, y, z)_t^{(i)}, h)\right|} \tag{8}$$

Hence, ω is the amount of occluded voxels divided by the number of voxels in a bounding box at the particle position that are not occupied and not in front of an obstacle. Intuitively, as the amount of occluded voxels increases, the expected number of laser points on the person decreases. This principle is now integrated in the observation model, which changes the observation model $P(z_t \,|\, x_t^{(i)})$ to

$$P(z_t \,|\, x_t^{(i)}) \sim \mathcal{N}\left(z_t;\ (1-\omega)\overline{z},\ (1-\omega)\sigma_{\overline{z}}^2\right). \tag{9}$$

To determine the occluded voxels inside $\mathcal{B}_r((x, y, z)_t^{(i)}, h)$ for a given particle position, we trace for each voxel in the bounding box a ray to the sensor. As the ray hits a voxel that is occupied, i.e., $P\left(\text{occupied}_{i,j,k}\right) \geq 0.5$, the voxel and the traversed voxels are marked as occluded. Only locally evaluating the occlusion and reusing the results from previous calculations turned out to be far more efficient than calculating the occlusion at every time step for the complete grid. Even though we have to calculate this for all particles individually. However, in the common case, the particles are concentrated around the position of the tracked person, the overlap of the bounding boxes of the particles is obviously significant and can be exploited by re-using already computed voxel occlusions.

B. Full Occlusions

If full occlusions were not modelled explicitly, the particles would spread out over the entire environment (in every iteration no correction using an observation could be made) and the person, after leaving the occlusion, could only be redetected by chance. A common and trivial solution is to end the track when all particles get the same weights due to missing detections. However, using the above defined occlusion ratio ω, we can deduce full occlusions – if ω is greater than a certain threshold, then we are apparently in a state of full occlusion.

As the tracked target enters full occlusion, indicated by $\omega \geq 0.9$, we uniformly distribute the particles at the boundary of the shadowed area behind the obstacle(s) which occlude(s) the person. The set of boundary voxels, in the same height k as the voxel $v_{i,j,k}$ at the estimated position $\hat{\mathbf{p}}_t$, is determined by region growing. As before, we calculate the occlusion for every voxel by tracing the ray starting at the current voxel, which ends at the sensor. Now two cases can occur:

1) The voxel is occluded by an obstacle, then we proceed to extend the region by growing to neighboring voxels that were not visited before.
2) The voxel is not occluded by any obstacle, then we add the voxel to the boundary voxels.

Now we can uniformly distribute the particles among the boundary voxels, and evaluate for every particle the observation model in subsequent iterations without applying the motion model (cf. fig. 3 (c)). Particles at the boundary get velocity vectors pointing away from the occlusion, since we can expect that the person will walk away from the obstacle. The magnitude of a velocity vector is sampled from the last velocity magnitudes from the particles. As we detect, that the person leaves the occlusion, i.e., there are at least 300 laser points in the bounding box around the particle, the particle filter is re-initialized at the position with maximal confidence, i.e., the particle that "sees" the most laser points. Altogether, the track is continued using the motion model again after the person has left the occluded space.

C. Person Height

In order to distinguish a tracked person from persons of different size and to enable a tracking in uneven terrain, we update a size value for the tracked person. In every scan, in which the person is partially or fully visible, we take the current estimated position $\hat{\mathbf{p}}_t$, and determine the height of the person. Starting from the voxel at $\hat{\mathbf{p}}_t$ we search for the last voxel in k-direction in the open bounding box that contains points and take the z-coordinate of that point as the top of

the person. The size is then determined by the difference between that z-coordinate and the z-coordinate of $\hat{\mathbf{p}}_t$. The overall size value is then the average size over all scans, in which the person has been tracked,

$$h_{t_s:t_c} = \frac{1}{t_s - t_c + 1} \sum_{t=t_s}^{t_c} h_t \tag{10}$$

where t_s is the number of the first tracked scan, t_c is the current scan and h_t is the guessed size in scan t. To weight the particles according to the mismatch in height h_t to the average height $h_{t_s:t_c}$, we use the following observation model

$$P(z_t \,|\, x_t^{(i)}) \sim \mathcal{N}\left(z_t; (1-\omega)\overline{z}, (1-\omega)\sigma_{\overline{z}}^2\right) \tag{11}$$

$$\cdot\, \Delta\left(h_{t_s:t_c} - h_t, \tau\right), \tag{12}$$

where $\Delta(a, b)$ denotes the triangle function, defined as follows:

$$\Delta(a,b) = \begin{cases} 1 - |ab^{-1}| & , |a| < |b| \\ 0 & , \text{ otherwise.} \end{cases} \tag{13}$$

With this additional term non-matching heights are penalized. As we will see, this stabilizes the tracking results, if persons of different sizes are observable in the environment. Furthermore, tracking in uneven terrain is possible, since particle positions beneath the real position are also penalized, which leads to an adaption of the z-coordinate of the position vector.

VI. Experimental Evaluation

We implemented the proposed tracking system in C++ and conducted several experiments for the tracking of a single person in static indoor environments and with a stationary sensor. We used a Velodyne HDL-64E laser scanner mounted on a tripod in the center of a hall. The Velodyne laser scanner was rotating with a speed of 9 Hz, so that approx. 144.000 points per turn were recorded. We used a computer with an Intel Xeon x5550 CPU with 2.67 GHz using a single core and 12 GB main memory.

We set the voxel size to $\rho = 0.125\,\text{m}$, since this gave the best tradeoff between visual results of the estimated trajectory and a reference trajectory. The grid had a length and width of 19 m and a height of 2.5 m. In all experiments, we used $M = 500$ particles and the expected number of points $\overline{z} = 950$ with standard deviation $\sigma_{\overline{z}} = 140$ was experimentally evaluated on a hold-out scene using the criteria minimal, average, and maximal error to a reference trajectory, and the number of lost tracks.

A. Runtime

Figure 4 shows the runtime of the approach and the average accuracy of the estimated trajectory using different voxel sizes ρ. We tested the approach in 10 runs for every voxel resolution.

As we argued in section IV, the update of the occupancy, where we have to traverse all voxels to update their occupancy, consumed the most time per iteration (approx. 40%). After this follows the computation of the occlusion

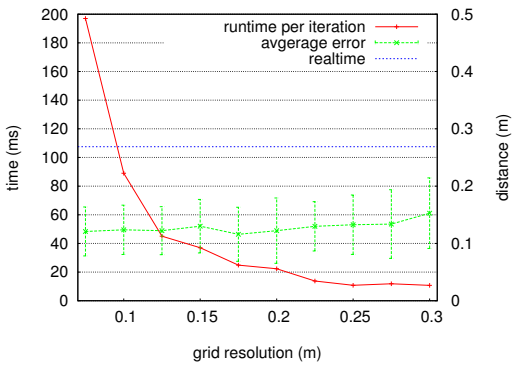


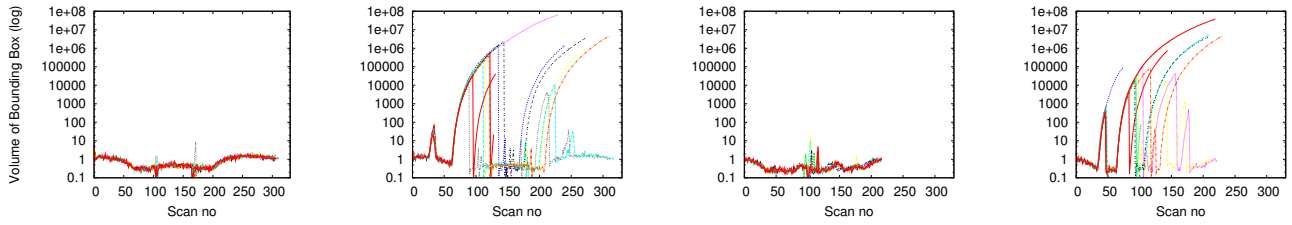
Fig. 4. Runtime of an iteration in milliseconds and average accuracy of the estimated trajectory using different grid resolutions. Shown are the needed times for occupancy update, occlusion calculations and other calculations with less impact on the run time. (Best viewed in color.)

(approx. 30%), which needs to trace a ray for every voxel in the bounding box. The influence of the occlusion update decreases with greater voxel sizes. The blue horizontal line in figure 4 depicts the time between two consecutive 3d laser range scans, thus we can process and update the particle filter using a complete turn in real-time with $\rho = 0.1\,\text{m}$. Also visible is a decrease in the accuracy as the grid resolution increases. In turn, the depicted standard deviation also increases, which is also visually observable in the estimated trajectories. Hence, we decided to use $\rho = 0.125\,\text{m}$ for all experiments in this rest of this section.

B. Partial and full occlusions

To evaluate the robustness of our approach against partial and full occlusions, we placed three different obstacles in the hall with heights of 1 m, 1.2 m, and 2 m, respectively. Two obstacle configurations were tested. In the first scene the obstacles are placed side by side with small gaps between them and the person walks behind every obstacle (cf. figure 6 middle). As introduced in section V-B our approach uniformly distributes the particles at the visibility boundaries of the obstacle, when a full occlusion occurs. The other scene is shown in figure 3, where the tracked person subsequently walks behind the three obstacles. The person is partially occluded behind the two smaller obstacles and fully occluded behind the largest obstacle. After leaving the full occlusion the person walks the same way back.

The effect of the adapted sensor model is evaluated by calculating in every time step the volume of the axis-aligned bounding box around the current particle set and compare the volume sizes of the adaptive and the non-adaptive observation model. The bounding box volume has advantages over considering the average distance to a ground truth trajectory. The latter one can be quite small, but the particles can still be spread over a large space, whereas the bounding box is only small, when the particles are concentrated around a position. As depicted in figure 5 a clearly visible effect can be seen. With an adaption of the expected number points (images (a) and (c)), the particle set stays concentrated on the person, whereas the non-adaptive observation model diverges in almost every pass, when entering a partial occlusion. This



(a) (b) (c) (d)

Fig. 5. Robustness against partial occlusions: Depicted is the log of the volume of an axis-aligned bounding box containing the particle set in two different scenes. Images (a) and (b) show the results of the first scene with (a) and without (b) occlusion adaption. As one can see, in case (a) the bounding box volume stays small even when the person is partially occluded, whereas in case (b) the particles spread and the person is lost, resulting in large volumes of the bounding box. The second scene shows similar results, depicted in images (c) with occlusion adaption and (d) without occlusion adaption.

	step1 (11 cm)	step2 (22 cm)	step3 (33 cm)	slant
avg (cm)	3.0	3.7	2.4	0.4
min (cm)	−2.1	0.7	0.7	−3.5
max (cm)	13.8	9.0	4.6	6.8
delay	1	2	4	/

TABLE I

AVERAGE, MINIMAL, AND MAXIMAL HEIGHT DEVIATIONS AND DELAYS AT DIFFERENT HEIGHTS IN THE LAST SCENE DEPICTED IN FIGURE 6.

can be explained by the effect that most of the particles receive similar weights using the non-adapting model, even those around the person's position, because there are fewer laser returns from the person than expected. When the person leaves the partial occlusion, the particles can concentrate again on the person in some passes, so that the bounding box volume decreases as shown in figure 5 (b) and (d).

Full occlusions are evaluated by comparing the scan number when a full occlusion is detected, with the scan number when the person actually is fully occluded. The latter one is the scan in which for the first time no more laser beams fall on the person. The end of a full occlusion analogously is the scan in which for the first time laser beams fall on the person again. A full occlusion is canceled if a particle has at least 300 points in its bounding box. If more than one particle does so, then the one with the highest number is selected. This results in a detection offset of ± 4 scans, meaning that the full occlusion is detected four scans too early up to four scans too late and a constant cancellation offset between 1 and 4, depending on the actual scene.

C. Tracking in uneven environments

In this experiment, we tested our approach for height adaption. As our observation model integrates the height of the person, the particle filter can track the height differences with a minor delay and a slight difference to the real height of the ground truth trajectory.

Table I shows the average of the differences in height between the trajectory and the actual object over 10 runs. Also shown is the delay of the trajectory when following the person up or down the steps. E.g., the 4 means it took 4 complete turns before the trajectory reached the height of step3 ± 5 cm after the person was already on that step.

D. Interfering Person

As stated in section V-C, the observation model is modified by a triangular function to penalize particles that are close to a person of different size. For an evaluation of this, we let two persons (1.8 m and 2 m) walk through the hall, and tracked the smaller person. Their paths crossed twice. The first time during a partial occlusion, where the tracked person is farther away from the sensor than the taller person. The second crossing occurs in non-occluded space. Two configurations were tested: (1) Full observation model as stated before, (2) observation model as before but without the penalizing term. In each configuration two cases can occur: (a) No particles or only a few shortly change to the wrong person and the track is continued without much disturbance, (b) the originally tracked person is lost and the other person gets tracked.

Our results show that the height adaption has a great advantage over the non-adaptive model. In particular, we let the scene run 25 times and determined in every run for both crossings if the particles switched to the taller person or not. In configuration (1) in only four cases the particles switched to the taller person during the first crossing. In the other 21 cases the particles stayed with the originally tracked person. During the second crossing the particles stayed on the previously tracked person in all 25 cases. During the first crossing for a short time the smaller person is nearly fully occluded by the taller person, which is not modeled and therefore leads to a spreading of the particles. Then, some particles are close to the taller person and receive a similar weight as the particles close to the smaller person, but due to the penalization this weight is reduced for particles near the taller person and so, during resampling, the particles get concentrated around the smaller person again. In the four exceptional cases too many particles were moved to the taller person. However, in configuration (2) the particles switched to the taller person in all 25 cases during the first crossing, while during the second crossing the particles remained on the taller person in every case.

VII. CONCLUSION AND FUTURE WORK

We presented an approach for tracking people using 3d laser range data with an adaptive observation model. Using this observation model, our approach is able to track a

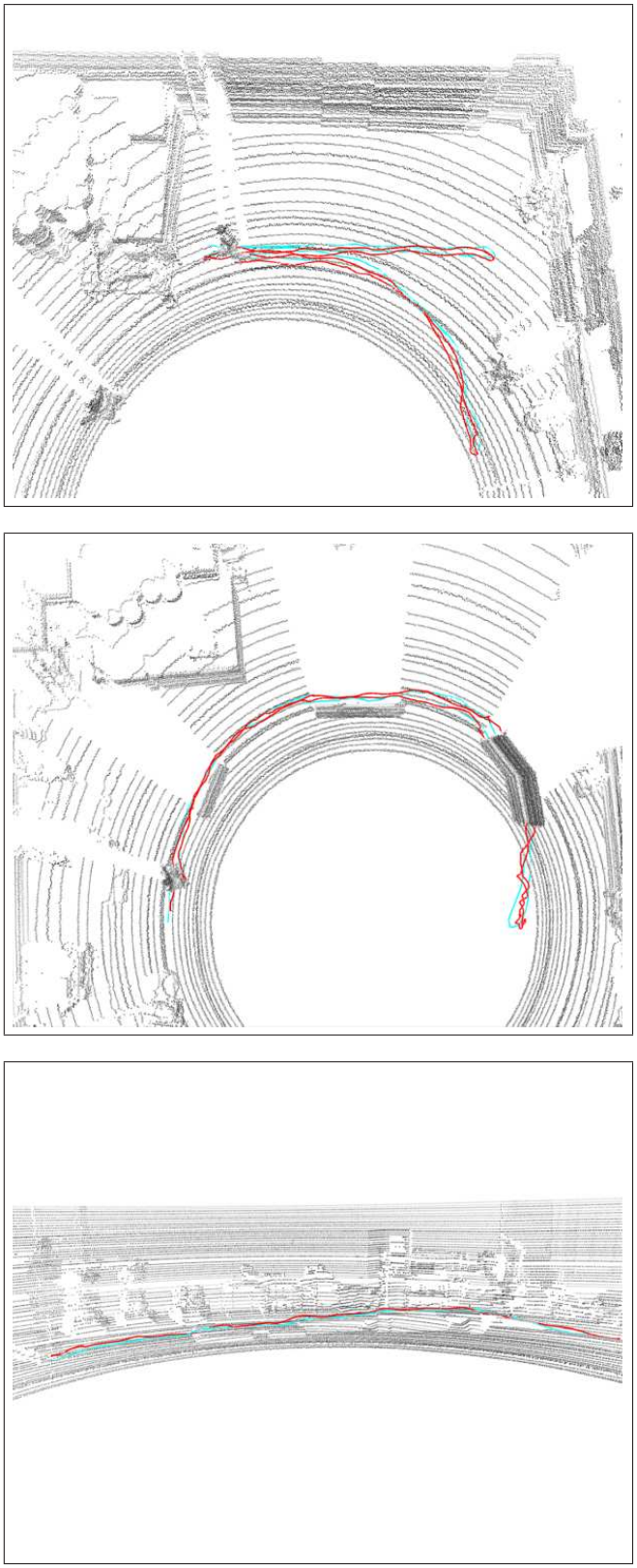


Fig. 6. Experimental setups. In top figure a setup without any occlusion is shown, the middle image depicts a scene with two partial occlusions and a full occlusion, and the bottom shows a trajectory in a non-flat environment. The reference trajectory is shown in light blue and the estimated trajectory of our approach is shown in red.

person even in situations with significant partial occlusions. Furthermore, we proposed a robust and efficient way of resolving full occlusions, when the target person is completely occluded by an obstacle. In those situations, our approach uniformly distributes the particles at the occlusion boundaries of the occluding obstacles, and thus the track can be continued in a controlled way. To accelerate the local evaluation of the occlusion we used a regular grid of voxels and an efficient ray tracing algorithm.

A challenging aim for future work is the reliable tracking of multiple targets with mutual occlusions using a moving platform. Another interesting avenue is the extension of the observation model to capture more characteristics of humans, and further exploit the appearance of people in 3d laser range data. This could be used to distinguish persons from one another and in turn enhance the data association in the multi-person tracking scenario.

VIII. ACKNOWLEDGMENTS

We want to thank Achim Königs, Frank Höller, Timo Röhling, Ansgar Tessmer and Michael Brunner at the Fraunhofer FKIE for helping us acquiring the data and providing technical support. Furthermore we want to thank Marcell Missura, who gave new insights to our works.

REFERENCES

[1] L. E. Navarro-Serment, C. Mertz, N. Vandapel, and M. Hebert, "LADAR-based Pedestrian Detection and Tracking," in *Proc. of the ICRA Workshop on Human Detection from Mobile Robot Platforms*, 2008.

[2] L. E. Navarro-Serment, C. Mertz, and M. Hebert, "Pedestrian Detection and Tracking Using Three-dimensional LADAR Data," *The International Journal of Robotics Research (IJRR)*, 2010, in press.

[3] A. Petrovskaya and S. Thrun, "Model based vehicle detection and tracking for autonomous urban driving," *Autonomous Robots*, vol. 26, no. 2-3, pp. 123–139, 2009.

[4] O. M. Mozos, R. Kurazume, and T. Hasegawa, "Multi-Part People Detection Using 2D Range Data," *International Journal of Social Robotics*, vol. 2, no. 1, pp. 31–40, 2010.

[5] L. Spinello, K. O. Arras, R. Triebel, and R. Siegwart, "A Layered Approach to People Detection in 3D Range Data," in *Proc. of AAAI Conference on Artificial Intelligence (AAAI)*, 2010, pp. 1625–1630.

[6] K. O. Arras, S. Grzonka, M. Luber, and W. Burgard, "Efficient people tracking in laser range data using a multi-hypothesis leg-tracker with adaptive occlusion probabilities," in *Proc. of IEEE International Conference on Robotics and Automation (ICRA)*, 2008, pp. 1710–1715.

[7] G. Taylor and L. Kleeman, "A Multiple Hypothesis Walking Person Tracker with Switched Dynamic Model," in *Proc. of the Australasian Conference on Robotics and Automation (ACRA)*, 2004.

[8] J. L. Bentley, "Multidimensional Binary Search Trees Used for Associative Searching," *Communications of the ACM*, vol. 18, no. 2, pp. 509–517, 1975.

[9] D. Meagher, "Geometric Modeling Using Octree Encoding," *Computer Graphics and Image Processing*, vol. 19, no. 2, pp. 129–147, 1982.

[10] D. Schulz, W. Burgard, and D. Fox, "People Tracking with a Mobile Robot Using Samble-Based Joint Probabilistic Data Association Filters," *International Journal of Robotics Research (IJRR)*, no. 2, pp. 99–116, 2003.

[11] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. The MIT Press, 2005.

[12] J. G. Cleary and G. Wyvill, "Analysis of an algorithm for fast ray tracing using uniform space subdivision," *The Visual Computer*, vol. 4, no. 2, pp. 65–83, 1988.

[13] M. Pharr and G. Humphreys, *Physically Based Rendering: From Theory to Implementation*. Morgan Kaufmann, 2010.