Activity-based Estimation of Human Trajectories

Slawomir Grzonka   Andreas Karwath   Frederic Dijoux   Wolfram Burgard

***Abstract*—We present a novel approach to incrementally determine the trajectory of a person in 3D based on its motions and activities in real-time. In our algorithm, we estimate the motions and activities of the user given the data obtained from a motion capture suit equipped with several inertial measurement units (IMUs). These activities include walking up and down staircases as well as opening and closing doors. We interpret the first two types of activities as motion constraints and door handling events as landmark detections in a graph-based simultaneous localization and mapping (SLAM) framework. Since we cannot distinguish between individual doors, we employ a multi-hypothesis tracking approach on top of the SLAM procedure to deal with the high data-association uncertainty. As a result, we are able to accurately and robustly recover the trajectory of the person. Additionally we present an algorithm to build approximate geometrical and the topological maps based on the estimated trajectory and detected activities. We evaluate our approach in practical experiments carried out with different subjects and in various environments.**

***Index Terms*—Activity Recognition, SLAM, Motion Capture**

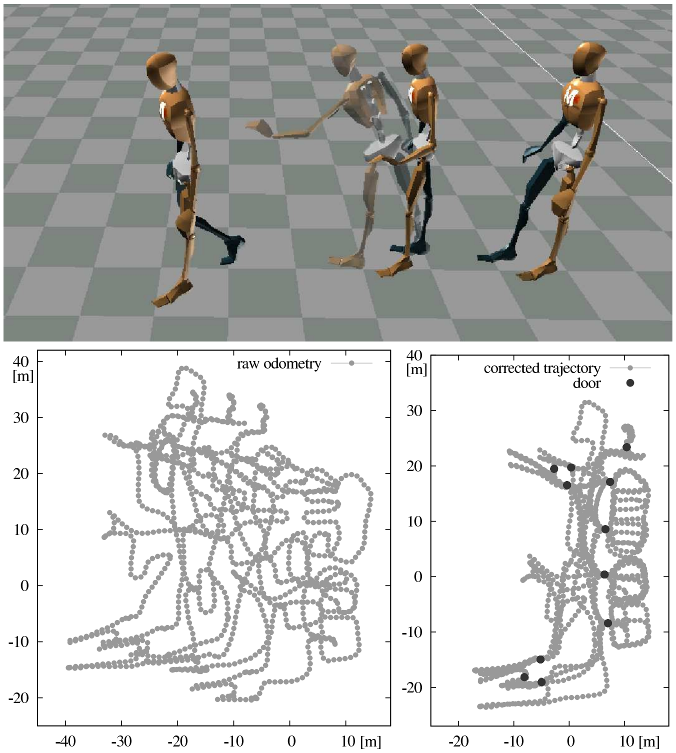


Fig. 1. Top: Typical data obtained from the Xsens MVN data suit when a subject opens a door. Our approach uses such motions to detect door handling events that are then utilized as landmarks in a graph-based formulation of the SLAM problem for recovering the full trajectory of the person. Bottom: The raw odometry data provided by the data suit (left) and the corrected trajectory after applying our approach (right).

I. Introduction

The problem of localizing and tracking people has recently received substantial attention in the robotics community as knowledge about the current position of its users can help a robot to improve its services. Especially in emergency situations, like after earthquakes or during fire fighting, the knowledge about the location of people can greatly support search and rescue missions. Consider, for example, firefighters in a building enclosed by smoke and fire. If a map of the environment can be constructed while the firefighters are within the building, an operator or automated system can re-route the people to the exit in case of an emergency. Alternatively, one can use the map of the environment to more intelligently coordinate the actions of the rescue workers to more effectively search the environment for potential victims and at the same time reduce the time the rescue workers are exposed to potential threats and hazards.

In this paper, we present an approach to recover human trajectories from data obtained with an XSens MVN data suit [1] by treating activities as landmarks. We employ this information in a graph-based SLAM approach to calculate the most likely trajectory of the human. The MVN data suit records full body postures of a subject, by using a set of inertial measurement units (IMUs) and a biomechanical human model. Figure 1(top) depicts typical data obtained from the data suit when a person opens a door, whereas the bottom left plot depicts the raw odometry estimated by the suit. The outcome of our proposed approach is depicted in the bottom right plot of Figure 1. To correct for odometry errors our approach applies supervised classification for different types of activities such as stair climbing and door handling. It then utilizes the learned classifiers to detect doors and stairs and applies a graph-based formulation of the SLAM problem to recover the full 3D trajectory of the person. In this formulation, the odometry estimated by the data suit and the estimated heights of the stairs are regarded as links between the landmarks which are constituted of the detected doors. To deal with the high data association uncertainty in the landmark detection, our algorithm applies a multi-hypothesis tracking scheme. After calculating the path of the person, our algorithm renders a map containing the individual stairs, the estimated doors, and approximate locations of walls. The work presented here extends our previous work [15] by detecting additional activities and extending our approach from 2D towards 3D trajectory reconstruction evaluated by a new set of large indoor experiments carried out in different environments.

The remainder of this paper is structured as follows. After discussing related work in the next section we present our approaches for learning door handling events and detecting stairs in Section III. Section IV introduces the multi-hypothesis tracking algorithm for sensors providing only positive feedback and especially the expressions needed to calculate the probabilities of individual hypotheses. Subsequently, Section V describes how we detect potential loop closure candidates. This is followed by the description of our overall system in Section VI. In Section VII we present our experimental results based on real data recorded with subjects walking inside of various buildings and covering multiple floor levels. We furthermore present our results on approximate mapping and compare the estimated maps with floor plans of the same building.

II. Related Work

The problem of tracking the correct data association [16] as well as human indoor navigation and localization has recently become an active research field [5], [17], [18], [24]. A number of different sensors have been employed as well as different kinds of localization techniques have been used. One of the first approaches in this area has been proposed by Lee and Mase [17], who employ wearable accelerometers and other sensors, like a digital compass and a velocity sensor, to recognize when humans perform specific activities and change their locations in indoor environments. They integrate the accelerometer data over time and estimate the position of humans in a known environment based on higher level descriptors such as *standing*, *2 steps north*, or *40 steps east* etc. The field of human indoor navigation and localization is therefore closely related to activity recognition using accelerometer data. Bao and Intille, [3] as well as Ravi *et al.* [21] have presented approaches to predict certain low level activities like *walking*, *standing*, *running*, *sit-ups*, and others using features from raw accelerometer data and a variety of different learning algorithms. However, they do not employ this information for indoor positioning. Schindler *et al.* [24] utilize an accelerometer together with an infrared proximity sensor mounted on a pair of headphones to detect when a human is passing through a doorway. In this work, the authors are able to construct topological maps, where rooms are represented by single nodes and edges represent the path in steps between doorways. For building these maps and for detecting loop closures, the user has to indicate by gesture which door was passed, i.e., giving each door a unique identifier via the infrared proximity sensor. They furthermore apply a Bayesian filtering scheme to localize the person within the resulting map.

In recent years, low-cost inertial measurements units (IMU) based on MEMS have become available and many researchers use such sensors for pedestrian localization, either alone or in combination with other sensors. Foxlin *et al.* [11] incorporate a zero velocity update allowing to estimate the users trajectory using an extended Kalman filter. Borenstein *et al.* [4] use a highly precise IMU also combined with zero velocity updates and obtain an accurate dead reckoning odometry. Woodman *et al.* [27] as well as Wang *et al.* [26] include additional information using WiFi. Both research groups employ a particle filter to track possible trajectories and calculate the weights of the particles based on the WiFi signal strength. Fischer *et al.* [10] discuss the possibility of using ultrasound sensors to reduce the error introduced by the MEMS sensors and present simulation results. Feliz *et al.* [8] utilize a neural network to estimate the step size using a singe IMU and thus estimate the odometry. Coley *et al.* [6] use wavelets to detect steps using gyroscopes only. In the work of Toth *et al.* [25], a prototype for pedestrian dead-reckoning and their general concept of sensor fusion is discussed. The HeadSLAM approach by Cinaz and Kenn [5] employs a laser scanner together with an IMU mounted on a helmet. They use the IMU sensor to project the laser scans into a horizontal plane in a global coordinate system and employ a variant of GMapping [14] for mapping. In particular, they incorporate a simplified motion model with two modes. Whereas the first mode corresponds to the activity walking and assumes constant velocity, the second mode represents the situation that the person is standing still and assumes zero speed. An overview over existing techniques can also be found in [9].

III. Feature Detection

The MVN software filters the raw data of the IMU's in the data suit and estimates an odometry of the body segments consisting of the (filtered) pose, velocity, and acceleration. However, we need to keep track of other specific events or features. Without this additional information we cannot detect loop closures and thus cannot correct the raw odometry from the data suit. A dead reckoning estimate of the trajectory, however, leads to an inconsistent map due to the accumulation of small errors over time as shown in Figure 1(bottom left).

In this work presented here, we restrict ourselves to structured environments such as office buildings. To allow us to correct the odometry within such buildings, we propose to use information about human activities as landmarks. We extract two different types of activities: *opening or closing of a door* and *walking up or going down a stair*. We use motion templates [15] to detect door opening or closing events and a neural network to detect steps. In the next sections, we will briefly describe both approaches.

A. Motion Templates

To learn the typical motion used for handling a door we use motion templates (*MT*) as proposed by Müller *et al.* [19]. The key idea of this work is to use simple Boolean features like *right hand is above head* and to create more expressive features (motion templates) by combining the simple ones. Given f of those features and a motion sequence of length K this leads to a matrix of size $f \times K$. Note that each entry of this matrix is either 1 or 0 indicating this feature being active or not at the specific time and that the sequence length K can in general be different for each motion sequence. Consider for example the two features $f_l,\ f_r$ with f_l indicating the left foot being in front of the body and f_r being 1 if and only if the right foot is in front of the body. Given this set of features, a typical walking template for two different sequences of the same length could look like Figure 2(a) and (b). The learned template given these two examples is depicted in Figure 2(c). Here, black and white correspond to 0 and 1 respectively. The gray-shaded boxes account for the fact 0.5 meaning *don't care*.

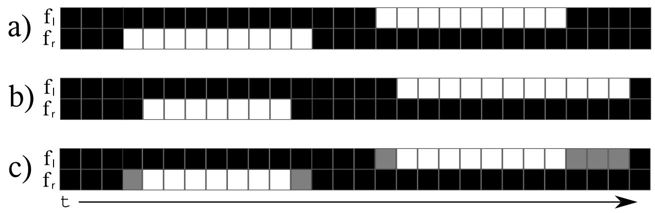


Fig. 2. A synthetic example: Given two examples (a) and (b) of the same motion *walking*. The features f_l, f_r are 1 (white) iff the left/right foot is in front of the body and 0 otherwise. The resulting merged template is depicted in (c). Here, gray areas indicate the value 0.5, meaning *don't care*. Intuitively, the matrix can be interpreted through the following sections: *feet parallel*, *right foot in front*, *feet parallel*, *left foot in front*, *feet parallel*.

The algorithm for learning a motion template for a single activity $\mathcal{A}$ can be briefly summarized as follows:

1) calculate the motion templates for all examples of $\mathcal{A}$
2) take one of the motion templates, called the reference template, and align all remaining to this one using dynamic time warping [20]. This procedure ensures that all other templates have now the same length as the reference template.
3) compute a new template as the average of all.
4) repeat the previous two steps for each motion template being once the reference template
5) replace the training data by the outcome of the computed templates
6) repeat the whole process until no major difference between the templates exists.

Note that the averaging of the templates include more complicated steps, but we refer to the original work of Müller *et al.* [19] for more details about learning a motion template.

Given the learned template for each activity (which we call a class template) and a new motion sequence, we can calculate a similarity between both. To do so, we compute a motion template of the actual sequence and align it to each class template using dynamic time warping. This allow us to compute a distance for each pair of templates. If this distance is below a threshold $\tau \in [0, 1]$ the actual motion sequence is said to belong to the same motion class as the class template. Intuitively, this value reflects the percentage of features which do not match the learned template.

Since we are only interested in the motion used for handling (i.e., opening or closing) a door with either the left or the right hand we use features based on the pose and velocity of the hands only. More precisely, we use a set of features describing whether the hand is at the level of the door handle, whether it is raising, hold still or lowered, and finally whether the hand is moving towards the body or away from it. We learned the template for the activity *handling a door*, which consists of the four subclasses *open left*, *close left*, *open right*, *close right*, using 10 examples from a training data set for each subclass. Based on a second validation data set, we selected the threshold $\tau = 0.25$ for detecting the motion. Using this threshold, we did not encounter any *false positives* on the validation data set. Within this process, we used data recorded by three subjects. The motion of two subjects was used for training, whereas the motion of the third one was used for validation. Although the features used for detecting a door are not very sophisticated, we can reliably detect the point in time when the door handle was touched within 1.5 seconds of the true event (we evaluated this using manually labeled ground truth). Therefore, we can use the pose of the hand as an approximation of the location of the door.

B. *Stair detection*

To be able to reconstruct 3D trajectories within buildings, it is inevitable to detect vertical movements of the user. Due to the high uncertainty in the height estimate of IMUs, the manufacturer's software assumes an environment consisting of a single floor. When walking up or down a staircase, the software "snaps" the human to the ground. Therefore, one needs additional means for determining changes in the z coordinate. In this paper, we achieve this by identifying stair stepping motions carried out whenever the user walks up or down staircases. In principle, we could have employed the same motion template approach as for the door handling events. However, in practical experiments we found that during typical stair-climbing people need approximately 0.5 seconds for each stair so that the motion templates described above, which detect doors with an accuracy of 1.5 seconds, were not accurate enough to exactly determine the point in time when the foot is placed onto a stair. However, increasing the time resolution of the MT accordingly leads to a high computational complexity due to the dynamic time warping. We therefore developed an efficient and temporally substantially more accurate classifier for detecting the individual stairs based on neural networks.

The goal of the following approach is to detect *stair* events, consisting of two subclasses namely *stair up* and *stair down*. To achieve this, our method employs a sliding window consisting of 5 frames that correspond to 40.7 milliseconds. Within this window, we extract features from the suit's data. In more detail, we use the relative position of the feet and the toes as well as the minimum and maximum acceleration. We trained the neural network using manually labeled training data employing SNNS [28] and RProp [23] as learning functions. The training data was recorded by a person walking up and down two different staircases twice and contains a total of 56 stair events, covering slightly more than two minutes. Once our predictor has detected a stair event, we estimate the height of each stair, by calculating the difference between the two feet along the z-axis given the pose estimates obtained from the data suit. Using this approach, we are able to detect step events with an error up to 1.5 frames (≈ 12ms) with respect to a manually measured ground truth.

Up to now, we are able to detect when the user climbed up or down a staircase and employing the motion templates, we are able to detect when the user touched *a door*. However, we do not possess any information of which door was handled. We therefore have to take care about possible data associations, which we deal with by employing a multi-hypothesis-tracker as described in the next section.

IV. Multi Hypothesis Tracking

In this section we review the Multi Hypothesis Tracker (MHT) as described by Reid [22] for sensors providing only positive feedback. If the user handles a door, we gain information about this door only and not about any other door in the users neighborhood, which is different from tracking multiple targets with a laser scanner for example. In the original paper by Reid, sensors providing only this kind of positive feedback are called type 2 sensors. There, any measurement can be either detected (assigned to an existing track), marked as a false alarm, or be a new track. Since in our particular case the tracks are static doors, we will call them doors in the remainder of this section, rather than tracks. As described in Section III-A we select a threshold for detection in such a way, that we do not have to model false positives. Therefore, a measurement can only be interpreted as *detected* (when matched to an existing door) or as a *new door*. In the remainder of this section we derive the probabilities of individual measurement assignments.

Let Ω_j^k be the $j-$th hypothesis at time k and $\Omega_{p(j)}^{k-1}$ the parent hypothesis from which Ω_j^k was derived. Let further $\Psi_j(k)$ denote an assignment, that based on the parent hypothesis $\Omega_{p(j)}^{k-1}$ and the current measurement z_k gives rise to Ω_j^k. The assignment set $\Psi_j(k)$ associates the current measurement either to an existing door or a new door. Given the probability of an assignment and the probability of the parent hypothesis $\Omega_{p(k)}^{k-1}$, we can calculate the probability of each child that has been created through $\Psi_j(k)$. This calculation is carried out recursively [22]:

$$\begin{aligned} p(\Omega_j^k|z_k) &= p(\Psi_j(k), \Omega_{p(j)}^{k-1}|z_k) \\ &\underset{\text{Markov}}{\overset{\text{Bayes+}}{=}} \eta\, p(z_k|\Psi_j(k), \Omega_{p(j)}^{k-1}) p(\Psi_j(k)|\Omega_{p(j)}^{k-1}) \cdot \\ &\quad\; p(\Omega_{p(j)}^{k-1}), \end{aligned} \tag{1}$$

with $p(\Omega_{p(j)}^{k-1})$ being the recursive term, i.e., the probability of its parent. Here, the factor η is a normalizer. The leftmost term on the right-hand side after the normalizer is the measurement likelihood. We assume that a measurement z_k associated with a door j has a Gaussian pdf centered around the measurement prediction $\hat{z}_k^j$ with innovation covariance matrix $\mathcal{S}_k^j$, $\mathcal{N}(z_k) := \mathcal{N}(z_k\,; \hat{z}_k^j, \mathcal{S}_k^j)$. Here, the innovation covariance matrix is the uncertainty of the door with respect to the current trajectory and is described in Section V. We further assume the pdf of a measurement belonging to a new door to be uniform in the observation volume V with probability V^{-1}. Hence, we have

$$p(z_k|\Psi_j(k), \Omega_{p(j)}^{k-1}) \;=\; \mathcal{N}(z_k)^{\delta} V^{\delta-1}\,, \tag{2}$$

with δ being 1 if and only if the measurement has been associated with an existing door and 0 otherwise. The central term on the right-hand side of Equation (1) is the probability of an assignment set, $p(\Psi_j(k)|\Omega_{p(j)}^{k-1})$, which is composed of the following two terms: the probability of detection $p_{det_j^k}$ and the probability of a new door. In our case the probability of a detection is equal to choosing one of the current candidate doors, i.e., all doors within an uncertainty ellipsoid. Therefore,

$$p_{det_j^k} \;:=\; \mathrm{NC}(\mathbf{x}_{1:k}, \Omega_{p(j)}^{k-1})^{-1}, \quad \text{with} \tag{3}$$

$\mathrm{NC}(\mathbf{x}_{1:k}, \Omega_{p(j)}^{k-1})$ being the number of door candidates, assuming the trajectory $\mathbf{x}_{1:k}$ within the world $\Omega_{p(j)}^{k-1}$. Assuming the number of new doors following a Poisson distribution with expected number of doors λ_{new} in the observation Volume V we obtain

$$p(\Psi_j(k)|\Omega_{p(j)}^{k-1}) \;=\; p_{det_j^k}^{\delta} \cdot \mu(1-\delta; \lambda_{new} V) \tag{4}$$

where $\mu(n; \lambda V) := \dfrac{(\lambda V)^n \exp(-\lambda V)}{n!}$ is the Poisson distribution for n events given the average rate of events is λ in the volume V. Therefore, Equation (1) can be reformulated as

$$\begin{aligned} p(\Omega_j^k|z_k) &= p(\Psi_j(k), \Omega_{p(j)}^{k-1}|z_k) \\ &\underset{Markov}{\overset{Bayes+}{=}} \eta\, p(z_k|\Psi_j(k), \Omega_{p(j)}^{k-1}) p(\Psi_j(k)|\Omega_{p(j)}^{k-1}) \cdot \\ &\quad\; p(\Omega_{p(j)}^{k-1}) \\ &= \eta\, \mathcal{N}(z_k)^{\delta} V^{\delta-1} p_{det_j^k}^{\delta} (\lambda_{new} V)^{1-\delta} \cdot \\ &\quad\; \exp(-\lambda_{new} V)(1-\delta)!^{-1} p(\Omega_{p(j)}^{k-1}). \end{aligned} \tag{5}$$

Observing that $(1-\delta)!$ is always 1 (since δ is $\in \{0, 1\}$) and noting that $\exp(-\lambda_{new} V)$ can be taken into the normalizer η, we can finally rewrite Equation (5) as

$$p(\Omega_j^k|z_k) \;=\; \eta \left(\mathcal{N}(z_k)\, p_{det_j^k}\right)^{\delta} \cdot \lambda_{new}^{1-\delta} \cdot p(\Omega_{p(j)}^{k-1}). \tag{6}$$

Up to now, we can detect doors and stair steps and calculate the probability of a data association. In the next section we address the remaining questions during our SLAM procedure, namely the detection of possible door candidates (i.e., loop closures), the calculation of the innovation covariance, and the algorithms which are utilized to correct the trajectory.

V. Simultaneous Localization And Mapping

We address the simultaneous localization and mapping problem by its graph based formulation. A node in the graph represents either a pose of the human (center of the hip) or a location of a door (pose of the hand which was handling the door) whereas an edge between two nodes models a spatial constraint between them. These spatial constraints arise either from incremental odometry, potentially adjusted according to the stair heights estimated from stair climbing events, or by closing a loop which corresponds to establish a data association between two doors. Thus, the edges are labeled with the relative motion between two nodes. To compute the spatial configuration of the nodes best satisfying the constraints encoded in the edges of the graph, we utilize a variant of stochastic gradient descent optimization [12], [13]. Since the door handling activities give us no information about roll and pitch, we restrict our optimization problem to (x, y, z, θ), with θ being the yaw. This allow us to adapt the fast 2D ((x, y, θ)) version of the tree-based network optimizer (Toro [2]) towards (x, y, z, θ) optimization and still maintain its computational properties. By repeatedly performing this optimization whenever a new door has been detected and a new data association has been established we can incrementally reduce the uncertainty in the current pose estimate while processing the data.

Since we are only able to detect the fact that there is a door, we have to track different possibilities of data association, namely whether the current detected door is one

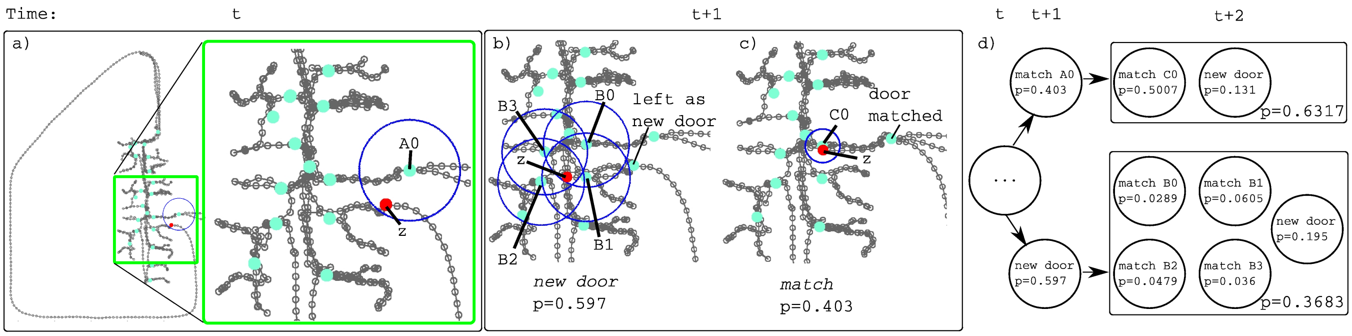


Fig. 3. A snapshot from one of our experiments. (a) The human re-enters the building through door A0. Based on the MHT decision *new door* and *match with A0* different hypothesis are generated (b) and (c). The probabilities of the hypothesis are depicted in (d).

of the already mapped doors or whether the door has not been perceived before. As already mentioned above, we utilize multi-hypothesis tracking for keeping track of all possible outcomes. To detect a potential loop closure (i.e., recognize a previously seen door), we identify all formerly detected doors within the uncertainty ellipsoid of the current pose by a Dijkstra projection of the node covariances starting from the current position. The innovation covariance is directly used for calculating the likelihood of the door as described in Equation (6). All doors being within the 99% confidence region of the current pose are considered as potential loop closure candidates, and together with the possibility of the current detected door being a *new door*, lead to $n+1$ different outcomes, given the number of loop closure candidates is n.

For each of these association possibilities we create a separate graph, encode the selected constraint and optimize it. The multi-hypothesis tree therefore grows exponentially in time and pruning of this tree is mandatory to keep computational costs reasonable. In our case, we utilize *N-scan-back* pruning as proposed by Cox and Hingorani [7], which works as follows: it considers an ancestor hypothesis at time $k-N$ and looks ahead in time to all its children at the current time k (the leaf nodes). The probabilities of the children are summed up and propagated to the parent node at time $k-N$. Given the probabilities of the possible outcomes at time $k-N$, the branch with the highest probability at time k is maintained whereas all others are discarded. Since in our case, a step in the MHT only arises when a door has been detected, this is identical to localize N steps ahead in time (at door level). In our implementation, we do not count a data association (step in time) if the only child of each hypothesis is the association with a *new door* or if the trajectory between two subsequent handling events was smaller than 1 m, reflecting the immediate closing of the same door after passing it. Thus we ensure that at least one combination of N data associations in time reflect an N step localization among different and already mapped doors.

An example of the N-scan-back MHT algorithm is visualized in Figure 3. This example is a snapshot from one of our experiments which is described in detail in our previous work [15]. At the specific time t, the human walked around the building leaving at the top exit and entered the building through the main entry labeled A0 in 3(a). Starting from the pose $\mathbf{z}$, where the current door was detected, the uncertainty of the pose was back-propagated utilizing Dijkstra expansion. Since we used the same uncertainty for x and y, the resulting ellipsoid is a circle. Note that due to the back-propagation of the uncertainty the current pose is in the uncertainty region of the door A0. For better visibility, only the doors being considered as candidates are shown with their uncertainty regions. Therefore, only two data associations are possible in this case, namely matching the current door with A0, which in this case is the correct association, or marking it as a new door. Calculating the posterior probability of each association leads to $p = 0.597$ for the case *new door* and $p = 0.403$ for the correct association. Note that in this situation, a maximum likelihood approach selects the wrong association. However, as the human enters the building and opens another door, given the previous association, different possible outcomes are possible. Figure 3(b) depicts the situation for the case that the previous decision was *new door* and Figure 3(c) shows the situation for the decision *match with A0*. Given this sequence of doors, the full posterior of the branch *match with A0* at time t sums up to 0.6317 while the probability for the branch for *new door* sum up to 0.3683 (see Figure 3(d)). Here, a N-scan-back of 2 would be sufficient to keep track of the correct data association, since the MHT can decide to keep *match with A0* at time t and discard the other branch.

The output of this approach can be used to generate an approximate geometrical as well as a topological map of the environment. In short, we build a modified Voronoi diagram based on the trajectory segments belonging to the same room. By assuming that doors separate rooms, we cut the trajectory based on the locations of individual doors. Thus, even when a door was not always detected or the user moved through an open door, the trajectory is segmented into different rooms, given the specific door was detected at least once. Given the orientation of the door, we merge subsequent segments which are both connected to the same door and on the same side (i.e., we cluster the segments according to which room they belong to). In order to seek for walls, we incrementally enlarge each cluster's trajectory until it touches a trajectory belonging to another room or up to a distance d, which was set to 1.5 m in all experiments. Since we segmented the trajectory with respect to different rooms, we also obtain a topological map of the environment at the same time. Typical outcomes of this process are shown in Section VII-B.

VI. Overall System

Our approach is summarized by the pseudo-code in Algorithm 1. Given the odometry up to the current point in time t, $\mathbf{x}_{1:t}$, the N-scan-back size n and the current multi-hypothesis tree $\Omega^{1:k} = \{\Omega^1, \ldots, \Omega^k\}$, with $\Omega^j = \{\Omega^j_{j_1}, \ldots, \Omega^j_{j_n}\}$, the algorithm works as follows. Note that k is the current depth of the hypothesis tree and is increased only if there is ambiguity in the data association of a door. First, we add a node (current pose of the hip) and an edge into each graph of the current hypothesis at the current depth k and detect the current activities in line 1-3. This is performed by using motion templates for detecting door handling events and neural networks for detecting step activities as described in Section III. If an activity is detected and this activity is a stair step, we augment the odometry information of the current added nodes with our height estimate (lines 4-8). If a currently detected activity is a door handling event, we calculate for each hypothesis Ω^k_j at depth k potential loop closure candidates C^k_j using a Dijkstra expansion starting from the corresponding current pose. If for all hypothesis no potential loop closure candidate exists, each of the current hypothesis can only include a *new door* as described by lines 17-20. Note that in this case it is obsolete to adjust the hypothesis probabilities since all probabilities are multiplied by the same factor λ_{new} which would be normalized out later on. In the case that at least one hypothesis at depth k has one potential loop closure candidate we create a new set of children for all hypothesis (lines 22-23). A *new door* is added to one child of each hypothesis whereas the graphs of the remaining children are augmented with the loop closure edges and the probabilities of the individual hypothesis are calculated according to Equation 6 (lines 24-31). Subsequently, we normalize the probabilities and perform the N-scan-back pruning as described in the previous section. Finally, we optimize the remaining hypotheses at depth $k+1$ and calculate the approximate map of the environment as specified by lines 32-36.

Algorithm 1 Human Indoor Mapping

```
Require: measurements up to current time $t$: $\mathbf{x}_{1:t}$
Require: N-scan-back size: $n$
Require: hypothesis tree: $\Omega^{1:k}$
 1: addNodeToEachHypothesis($\mathbf{x}_t$)
 2: addEdgeToEachHypothesis($\mathbf{x}_{t-1}$,$\mathbf{x}_t$)
 3: $\mathcal{A}$ = detectCurrentActivities($\mathbf{x}_{1:t}$)
 4: if stepActivity $\in \mathcal{A}$ then
 5:   $\mathbf{x}_t$ = estimateHeight($\mathbf{x}_t$)
 6:   updateLastAddedNodeInEachHypothesis($\mathbf{x}_t$)
 7:   updateLastAddedEdgeInEachHypothesis($\mathbf{x}_{t-1}$,$\mathbf{x}_t$)
 8: end if
 9: if doorActivity $\in \mathcal{A}$ then
10:   da = doorActivity // for better readability
11:   $k_n = |\Omega^k|$ // number of hypothesis at depth $k$
12:   $v = 0$ // number of all loop closure candidates
13:   for $j = 1, \ldots, k_n$ do
14:     $C^k_j$ = calculateLoopClosureCandidates($\Omega^k_j$)
15:     $v = v + |C^k_j|$
16:   end for
      // no candidates $\rightarrow$ new door for all hypothesis
17:   if $v == 0$ then
18:     addDoorNodeToEachHypothesis(da.hand($\mathbf{x}_t$))
19:     addEdgeToEachHypothesis($\mathbf{x}_{t-1}$, da.hand($\mathbf{x}_t$))
20:   else
21:     for $j = 1, \ldots, k_n$ do
22:       $v_j = |C^k_j|$ // current number of candidates
23:       $\{\Omega^{k+1}_1, \ldots, \Omega^{k+1}_{v_j+1}\}$ = createChildren($\Omega^k_j$, $v_j + 1$)
          // new door
24:       $\Omega^{k+1}_{v_j+1}$.addDoorNode(da.hand($\mathbf{x}_t$))
25:       $\Omega^{k+1}_{v_j+1}$.addEdge($\mathbf{x}_{t-1}$, da.hand($\mathbf{x}_t$))
26:       calculateProbability($\Omega^{k+1}_{v_j+1}$)
          // loop closures
27:       for $i = 1, \ldots, v_j$ do
28:         $\Omega^{k+1}_i$.addLoopClosureEdges($C^k_j(i)$)
29:         calculateProbability($\Omega^{k+1}_i$)
30:       end for
31:     end for
32:     $k = k + 1$
33:     normalizeProbabilities($\Omega^{k+1}$)
34:     nScanBackPruning($\Omega^{k+1-n:k+1}, n$)
35:     optimizeEachHypothesis($\Omega^{k+1}$, numIterations)
36:     calculateApproximateMapForEachHypothesis($\Omega^{k+1}$)
37:   end if
38: end if
```

VII. Experiments

The following sections show the results obtained with our currently implemented system. First, we will present our results on 3D and 2D trajectory reconstruction based on human motion and activity and evaluate the error of our estimated door locations wrt. a manually measured ground truth. We calculate the error by first estimating the best transformation between the estimated map and the ground truth throughout all floors. This transformation is then used to calculate the error (mean and standard deviation) between the estimated door locations and the ground truth map. In Section VII-B, we finally present our results on approximate mapping. Videos of the experiments can be found on the Web (*http://ais.informatik.uni-freiburg.de/projects/mvn*) showing the incremental update of the final best hypothesis. Our current system, though not optimized, is able to perform an incremental update at a rate of 10Hz on an Intel i7 1.7 GHz laptop.

A. 3D Trajectory Estimation

We evaluated the approach described above on different data sets in which the user walked through buildings containing several floor levels. All experiments were performed using an N-scan-back of 3 and $\lambda_{new} = 0.03$, which approximately is the number of doors relative to the area covered by the building. In general, λ_{new} depends on the type of building. For example, in a hotel λ_{new} should be significantly higher than in a warehouse. However, we found that small changes to this parameter do not lead to substantially different results. Thus, the remaining free parameter is the covariance matrix for the Dijkstra expansion. Recall that we have no information about the current magnetic field. The covariance matrix, therefore, also reflects the magnetic disturbances present in the building,

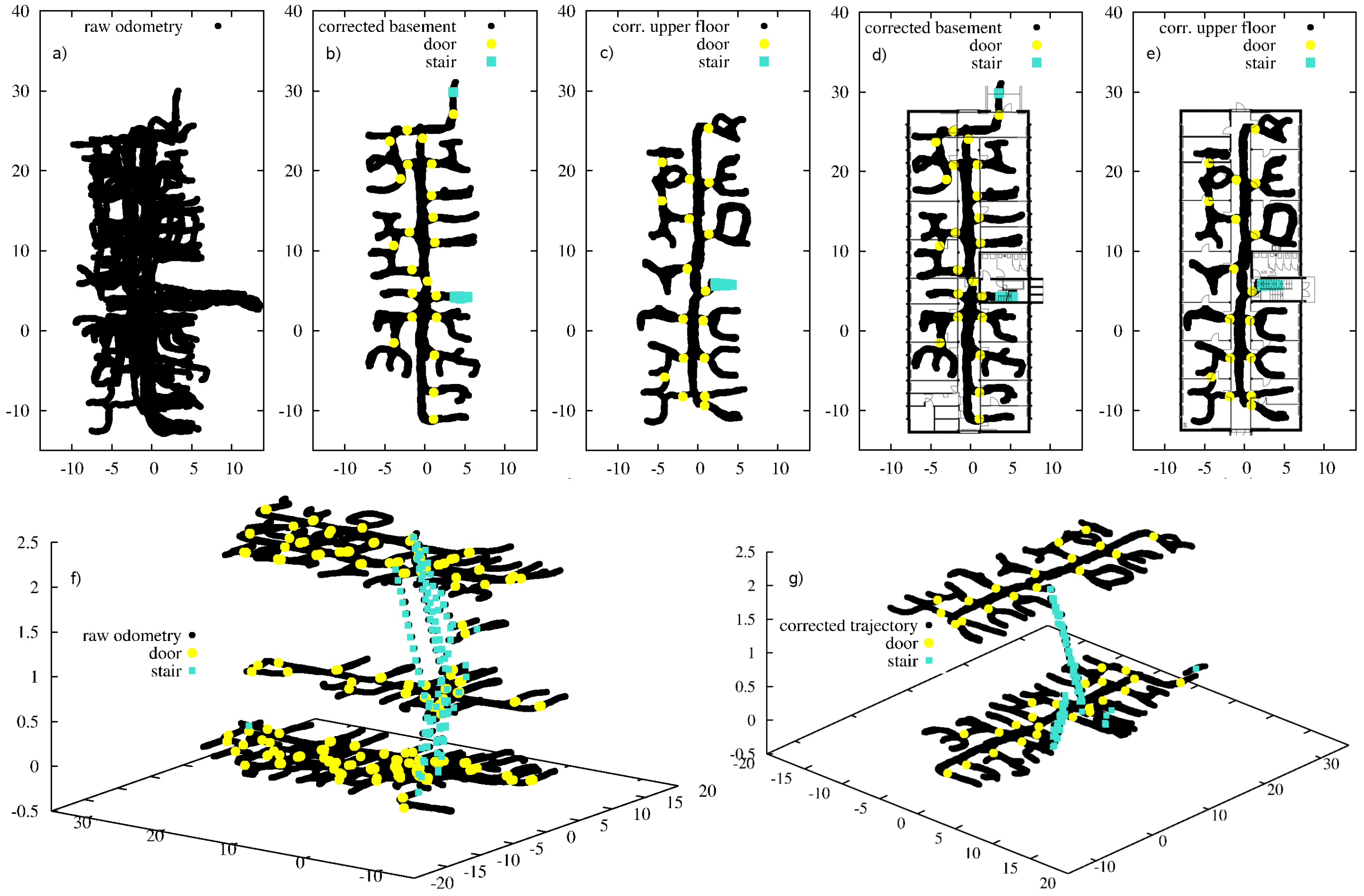


Fig. 4. Outcome of the first experiment: (a) the raw trajectory estimated by the data suit. The estimation of the ground level and the first floor are shown in (b) and (c), and aligned to a floor plan in (d) and (e) respectively. The raw odometry combined with the raw detection of stairs and doors is shown in (f). A 3d plot of the building estimated by our approach is shown in (g).

since high magnetic field errors result in a high pose error estimation from the data suit. Note, that all plots of single levels of the buildings given in this section also contain all points up to the middle of the next and the previous floor respectively. Furthermore, all distances are given in meters. Please also note that the raw data (without the step detection) contains no information along the z-axis wrt. different floors, i.e., only a single floor level is present.

The first experiment contains a trajectory of approximately 2.2 km including 222 door handling actions and is depicted in Figure 4. The building has three floor levels, namely the basement, an intermediate floor level containing the main entrance, and the first floor. Since the intermediate level contains only the main entrance door, we omitted to plot this floor separately for better readability. We used a variance of $0.03\,\text{m}$ per meter in x as well as in y and a variance of $0.1\,\text{m}$ per meter along the z axis. Our approach reliably detected 215 out of the 222 door handling events with one false alarm. The average error of the estimated door locations is $0.31\,\text{m}\pm0.17\,\text{m}$ wrt. a manually measured ground truth. We detected 106 out of 116 stairs, missing 7 stairs down and 3 stair up and had one false alarm. The difference in the calculated stair size between up and down is approximately 3.5 cm. The raw odometry is depicted in Figure 4(a). Although no floor level information is present in the raw data, the raw odometry trajectory is already quite accurate. This results from the fact that the building contains less metal structure compared to modern buildings so that we obtained only small magnetic disturbances. As can be seen in the next experiments, larger disturbances typically lead to a high pose error. The raw odometry including our step and door detection is plotted in Figure 4(f). The maximum-likelihood map estimated by our approach is depicted in Figure 4(g). For better comparison, we also segmented the trajectory for different floor levels and compare them to floor plans generated by the architect of the same building as shown in Figure 4(b-e).

The data for the second experiment was recorded in a typical university building containing several floors and including small seminar rooms as well as big lecture halls and a small library. The trajectory is approximately 2.85 km long covering three floor levels. This experiment is challenging due to two reasons. First, the metal disturbances rising from the metal structure of the building itself and from walking closely to chairs and tables lead to a high pose error as can be seen in the raw data depicted in Figure 5(a). Second, the first and the second floor are nearly identical on one side of the building which results in many potential loop closure candidates. Compared to the first experiment, this building contains in total five different staircases. Two staircases are present in each of the two lecture halls (see Figure 5(f) left part) connecting the first and the second floor, whereas the main staircase connects all three floors. In this experiment,

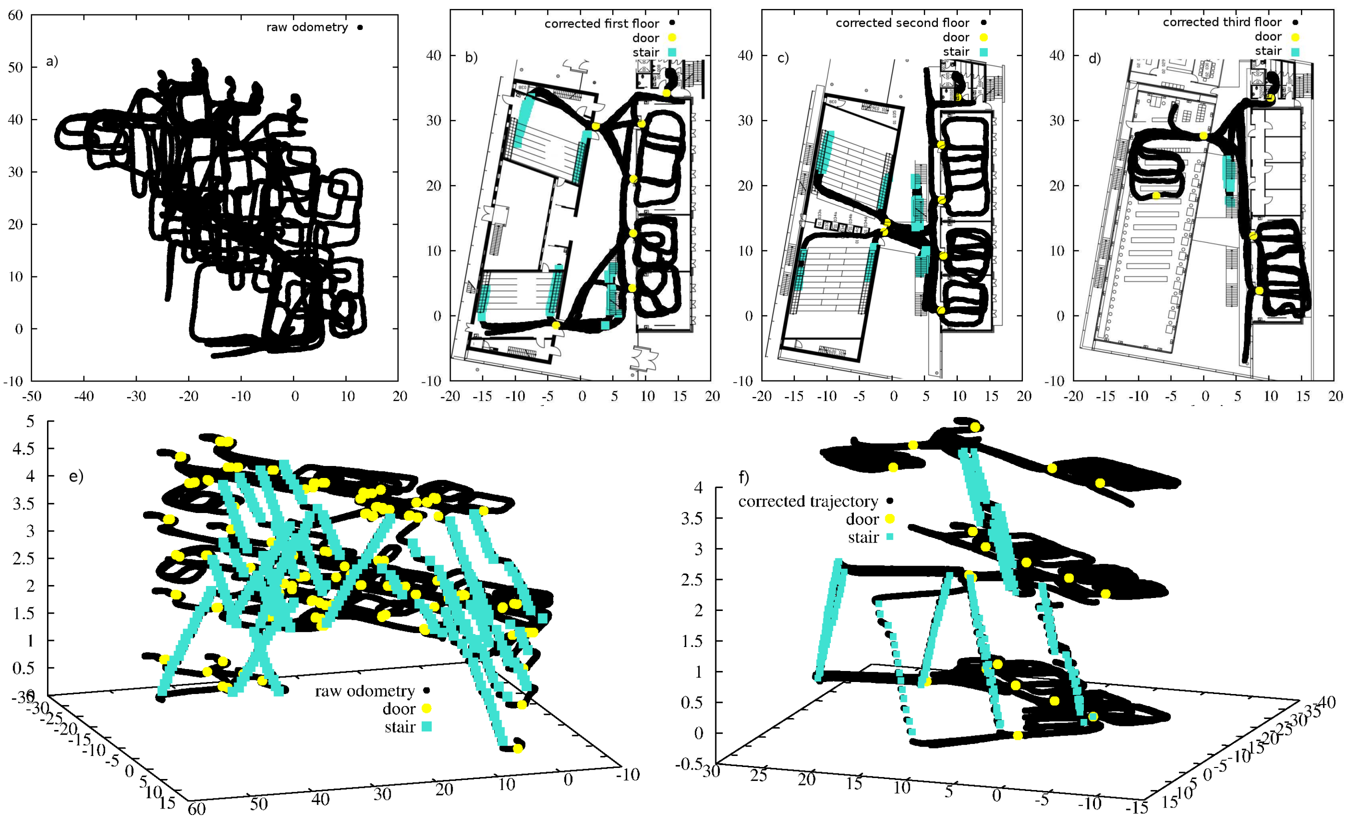


Fig. 5. The second experiment was performed in a university building. The raw data is depicted in (a), whereas the different floor levels plotted on top of the floor-plans of the building are shown in (b)-(d). The raw trajectory including the uncorrected stairs and doors is depicted in (e). The maximum likelihood estimate of the whole building using our approach is shown in (f).

we used a variance of $0.1\,\mathrm{m}$ per meter in all directions, i.e., x, y, and z. The raw odometry trajectory including the steps detected by our algorithm is plotted in Figure 5(e). The maximum-likelihood result of our approach compared to the floor plans of this building are shown in Figure 5(b)-(d). Finally, the maximum likelihood estimation of the whole building is depicted in Figure 5(f). In this experiment we detected 175 out of 178 door handling events with an average error of $1\,\mathrm{m}{\pm}0.41\,\mathrm{m}$. We also had one false alarm at the third floor level which originates from moving a chair away in the library which was blocking the user's path. Regarding the stair detection we missed 62 out of 473 stairs (42 stairs up and 20 stairs down). The average difference between the calculated stair heights is $1.3\,\mathrm{cm}$.

The third experiment was recorded in a university building consisting of five floors and containing a substantial amount of metal structures. Here, the magnetic disturbances did not even allow for a proper initial calibration of the data suit. This had a severe influence on the estimated raw odometry trajectory. We intentionally included this experiment to show the robustness of the current approach even in the context of substantial disturbances. Although our assumption about a Gaussian error on all degrees of freedom is highly violated (for example, one staircase is rotated by 45 degrees in the raw odometry data) we still were able to approximately recover the true trajectory but with one misaligned door (see Figure 6(b)). This door, which is marked by an arrow in the figure, is wrongly labeled as a new door. As in the previous experiment, we used a innovation of $0.1\,\mathrm{m}$ per meter along all axis. The total distance traveled in this building is approximately $1.46\,\mathrm{km}$ and contains 135 door handling events from which our approach detected 126. It furthermore resulted in one false alarm in the lower left corner of the first floor. The least mean square error of our estimated door locations is $0.67\,\mathrm{m}{\pm}0.40\,\mathrm{m}$. Regarding the step detection, we were able to detect 271 out of 280 stairs, missing 7 stairs up and 2 stairs down. The calculated stair size for the class *stair down* was in average 4cm higher than for the class *stair up*. The raw trajectory is depicted in Figure 6(a) and (g) together with the raw steps and doors detected by our algorithm. The resulting map estimated by our approach is depicted in Figure 6 (h). The individual floors plotted on top of the floor plan are shown in Figure 6(b)-(f). Note that the estimate of the first floor is slightly suboptimal due to the severe error in the raw data. Since some of the doors were locked, we were not able to enter all rooms. The corresponding doors appear to be not connected to the trajectory in Figure 6(c)-(f). This effect originates from the fact, that the user was not able to pass through the corresponding doorways, i.e., door positions are obtained by the hand pose handling the door whereas the trajectory is given by the position of the user's hip.

We also performed several experiments covering a single floor level using the motion of different subjects (see our previous work [15] for more details). The trajectory of the fourth experiment is about $1.6\,\mathrm{km}$ long. Our approach reliably detected 125 out of 133 door handling events. The corrected trajectory including the approximate location of walls is shown in Figure 7(a). This experiment also contains several loops around the building but we show only the inner part for better

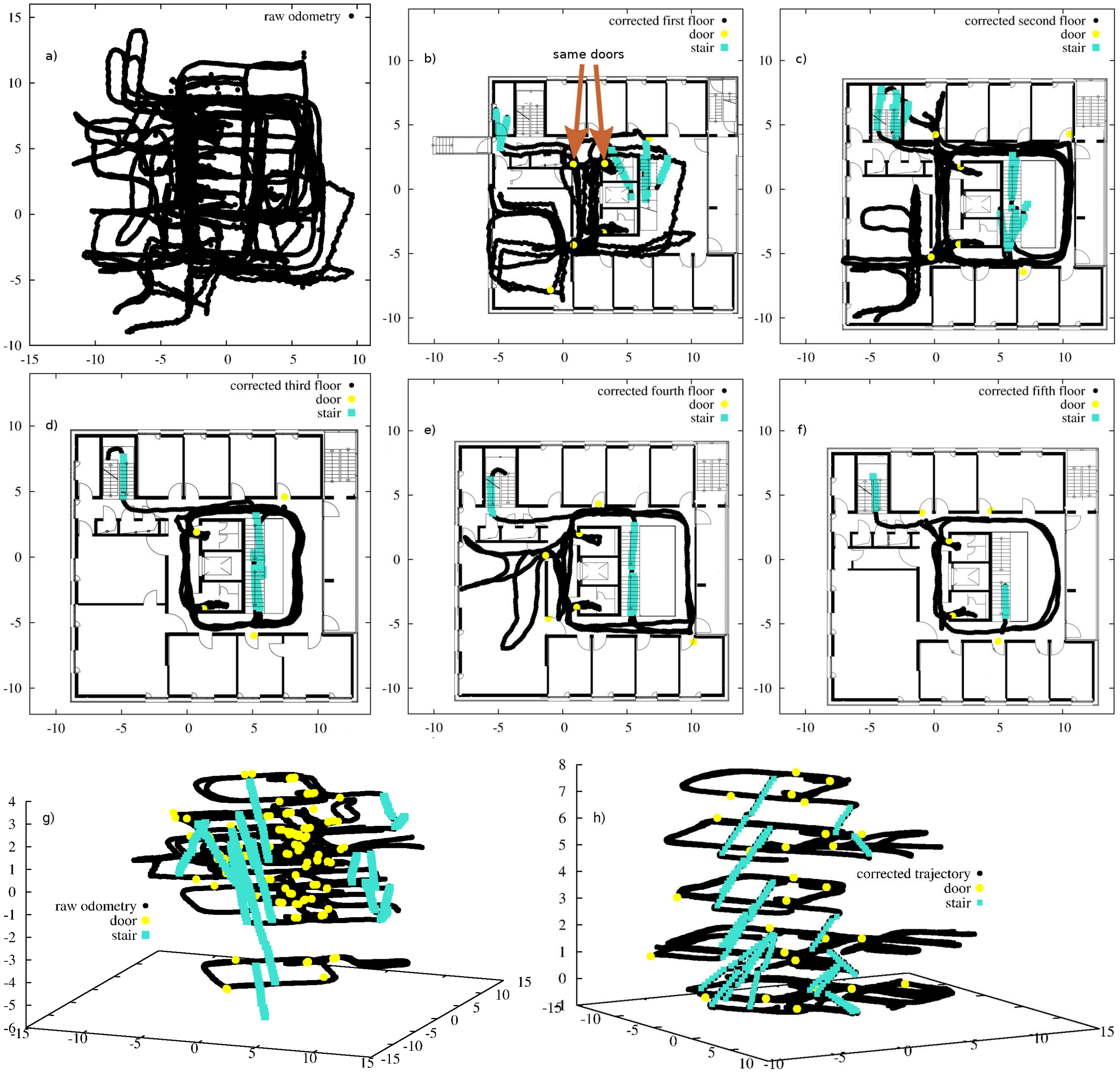


Fig. 6. The third experiment was performed in a building containing a lot of metal structure. This introduced severe errors in the trajectory provided by the data suit, especially when walking up and down the staircase between the first and the second floor. Note the two instances of the staircase are rotated by approximately 45 and -40 degrees in the left part of image (g). The raw trajectory is depicted in (a) and (g) together with the outcome of our step and door detection algorithms. The maximum likelihood map of the whole building is shown in (h), whereas the individual floors compared to its floor plans are shown in (b) through (f). Note that the high pose errors lead to a wrong data association in the ground level (a), where the left door marked with the arrow was wrongly labeled as a *new door*.

readability. Note that this experiment was recorded in the same building as the first one and that we used the same parameters. The error of the estimated door locations is $0.5\,\text{m}\pm0.24\,\text{m}$.

The fifth experiment covers a trajectory of approximately 1.3 km and our approach reliably detected all 63 door handling events with an error of $0.61\,\text{m}\pm0.17\,\text{m}$. The corrected trajectory is shown in Figure 7(d). Here, we used the same parameters as in the second experiment since it was performed in the same building.

The last experiment was recorded in a typical office environment. For this experiment we asked people from a company to record data while walking in their building. The raw odometry estimate is shown in Figure 7(g) and the corrected trajectory is shown in (h). The trajectory is approximately 0.4 km long. In this experiment, we detected 24 out of 27 door handling events and used the same parameters as in the previous one. However, since this experiment was recorded by a different team, we do not have ground truth data of the locations of the doors but only a floor plan of the building.

The outcome of all experiments together with the parameters used is also summarized in Table I.

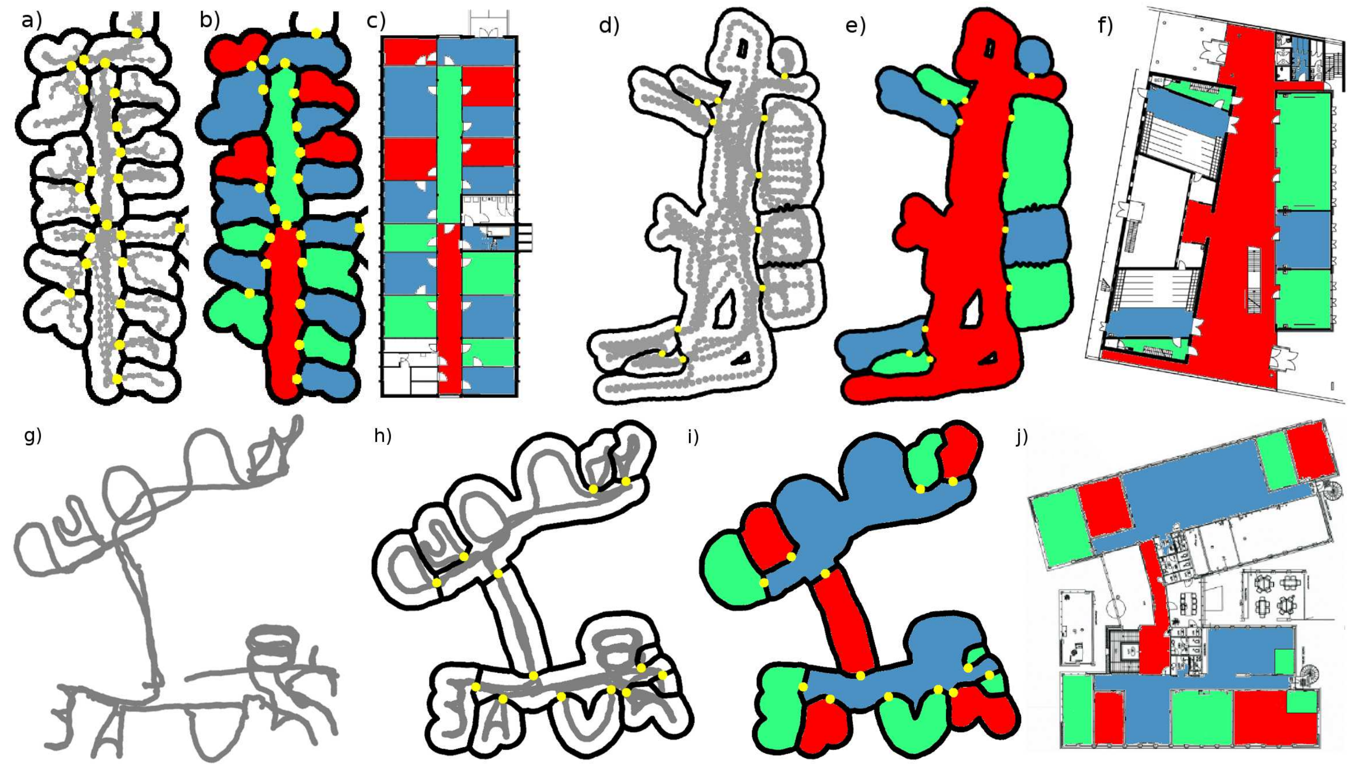


Fig. 7. Typical outcomes of our approximate mapping algorithm and the corresponding floor plan of the same building. There is a high correlation between the map generated by our approach and the corresponding floor plan. Our approximate mapping approach segments the trajectory into different rooms and calculates approximate location of walls. Due to the segmentation we also obtain a topological map which we colored in order to highlight the segmented rooms. The first images (a)-(c) show the result of the fourth experiment. The approximate map is shown in (a) and the colored according to detected rooms in (b). The corresponding floor plan is shown in (c) and colored respectively. The results for the fifth experiment are shown in (d)-(f). The bottom row shows the result for the last experiment, including the raw odometry trajectory (g) and the outcome using our approach (h)-(j). Note that the inner walls in (b) are present since all experiments were performed with a maximum distance of $d = 1.5\,\text{m}$, i.e., the distance of a wall to the nearest trajectory is at most 1.5 m.

Experiment No.	Trajectory length	No. of floors	Door detection		Step detection		Parameters				Error of estimated door locations
			Recall rate	False Positives	Recall rate	False Positives	λ_{new}	N	$\sigma^2_{x,y}$	σ^2_z	
1	2.2 km	2	0.968	1	0.914	1	0.03	3	0.03	0.1	0.31 m±0.17 m
2	2.85 km	3	0.983	1	0.869	0	0.03	3	0.1	0.1	1 m±0.41 m
3	1.46 km	5	0.933	1	0.968	0	0.03	3	0.1	0.1	0.67 m±0.40 m
4	1.6 km	1	0.94	0	n/a	0	0.03	3	0.03	0.1	0.5 m±0.24 m
5	1.3 km	1	1	0	n/a	0	0.03	3	0.1	0.1	0.61 m±0.17 m
6	0.4 km	1	0.889	0	n/a	0	0.03	3	0.1	0.1	n/a

TABLE I

SUMMARY OF ALL EXPERIMENTS. THE RECALL RATE IS CALCULATED AS THE RATIO OF TRUE POSITIVES VERSUS THE ACTUAL NUMBER OF EVENTS. N IS SHORT FOR N-SCAN-BACK, $\sigma^2_{x,y}$ IS SHORT FOR σ^2_x, σ^2_y, AND N/A IS SHORT FOR NOT AVAILABLE.

B. Approximate Mapping

In this section we show our results of approximate mapping for floors of different buildings. Figure 7(a)-(j) show typical outcomes of our approach and the buildings floor plans respectively. Note that our mapping technique segments the trajectory into different rooms. We therefore can calculate both, a geometrical and a topological map. The topological map colored wrt. different rooms (using 3 colors in total) is shown in Figure 7(b),(e), and (i). The corresponding floor plan have been colored respectively and are shown in Figure 7(c),(d), and (j). As can be seen, there exists a high correlation between the estimated and the real floor plans. Errors mainly arise from rotational errors as can be seen in the bottom left part of Figure 7(d). These rotational errors, however, can be corrected by including an additional loop around the building from the outside. The walls within the map of Figure 7(d) are present since all experiments were performed using a maximum distance of $d = 1.5\,\text{m}$ as described in Section V. Figures 7(g)-(j) depict the outcome of an experiment in a typical company environment including the raw odometry trajectory (g). Figure 8 shows the outcome of our segmentation approach for the second experiment. Here, we omit to plot the walls since the perspective view of the 3D structure in combination with the outer walls would render the figure completely black. However, the trajectory represents also the topological structure of the building as can be seen by comparing with the corresponding floor plans of the same floor. As can be seen from these experiments, our approach is robust and can be applied in different environments.

VIII. Conclusions and Future Work

In this paper, we presented a novel approach to accurately estimate the 3D trajectories of humans based on data gathered with a motion capture suit. Our approach extracts two different

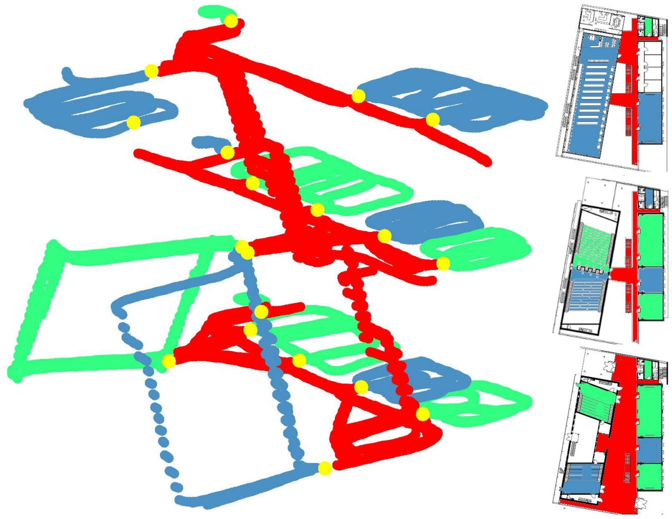


Fig. 8. Outcome of our approximate mapping algorithm and the corresponding floor plans of the same building for the second experiment. Please note, that we omit the plotting of walls, as the perspective view of the 3D structure in combination with outer walls would lead to a black figure. The three floor plans on the right hand side are colored wrt. the segmented trajectory and reflect the individual floors of the building (see also Figure 5).

activities from the motion data, namely door handling and stair climbing events. We consider the trajectory of the person and the height estimates of our step detection algorithm as motion constraints. The door handling events detected using specific motion templates are used as landmarks within a graph-based SLAM approach. To cope with the high data association uncertainty, we employ a multi-hypothesis tracking approach. Additionally, our method can create approximate geometrical as well as topological maps of the environment based on the estimated trajectory and activities. Our system has been implemented and successfully tested on real data recorded with different subjects in several buildings of a university campus as well as in a typical office environment. The experimental results demonstrate that our approach is able to robustly keep track of the true data association and accurately estimates the trajectory taken by the person. Furthermore, the resulting maps accurately resemble the corresponding environments. In future work we aim to make our algorithm more robust, especially with respect to magnetic disturbances.

IX. Acknowledgments

The authors would like to thank Gian Diego Tipaldi for the fruitful discussions. This work has partially been supported by the German Research Foundation (DFG) under contract number SFB/TR-8.

References

[1] Xsens MVN Suit http://www.xsens.com/en/general/mvn.

[2] OpenSLAM - Open Source Navigation Software Repository, http://www.openslam.org.

[3] S. S. Bao, L. Intille. Activity recognition from user-annotated acceleration data. pages 1–17. Springer, 2004.

[4] J. Borenstein, L. Ojeda, and S. Kwanmuang. Heuristic reduction of gyro drift for personnel tracking systems. *Journal of Navigation*, 62(01):41–58, 2009.

[5] B. Cinaz and H. Kenn. HeadSLAM - simultaneous localization and mapping with head-mounted inertial and laser range sensors. *Wearable Computers, IEEE International Symposium*, 0:3–10, 2008.

[6] B. Coley, B. Najafi, A. Paraschiv-Ionescu, and K. Aminian. Stair climbing detection during daily physical activity using a miniature gyroscope. *Gait & posture*, 22(4):287–294, 2005.

[7] Ingemar J. Cox and Sunita L. Hingorani. An efficient implementation of Reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(2):138–150, 1996.

[8] R. Feliz, E. Zalama, and J.G. García-Bermejo. Pedestrian tracking using inertial sensors. *Journal of Physical Agents*, 3(1):35–42, 2009.

[9] C. Fischer and H. Gellersen. Location and Navigation Support for Emergency Responders: A Survey. *IEEE Pervasive Computing*, 9(1):38–47, 2010.

[10] C. Fischer, K. Muthukrishnan, M. Hazas, and H. Gellersen. Ultrasound-aided pedestrian dead reckoning for indoor navigation. In *Proceedings of the first ACM international workshop on Mobile entity localization and tracking in GPS-less environments*, pages 31–36. ACM, 2008.

[11] E. Foxlin. Pedestrian tracking with shoe-mounted inertial sensors. *IEEE Computer Graphics and Applications*, pages 38–46, 2005.

[12] G. Grisetti, S. Grzonka, C. Stachniss, P. Pfaff, and W. Burgard. Efficient estimation of accurate maximum likelihood maps in 3d. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, San Diego, CA, USA, 2007.

[13] G. Grisetti, D. Lodi Rizzini, C. Stachniss, E. Olson, and W. Burgard. Online constraint network optimization for efficient maximum likelihood mapping. In *IEEE Int. Conf. on Robotics & Automation (ICRA)*, Pasadena, CA, USA, 2008.

[14] G. Grisetti, C. Stachniss, and W. Burgard. Improving grid-based slam with Rao-Blackwellized particle filters by adaptive proposals and selective resampling. In *IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 2443–2448, Barcelona, Spain, 2005.

[15] S. Grzonka, F. Dijoux, A. Karwath, and W. Burgard. Mapping indoor environments based on human activity. In *IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2010.

[16] D. Hähnel, W. Burgard, B. Wegbreit, and S. Thrun. Towards lazy data association in slam. In *Proc. of the Int. Symposium of Robotics Research (ISRR)*, Siena, Italy, 2003.

[17] S.-W. Lee and K. Mase. Activity and location recognition using wearable sensors. *IEEE Pervasive Computing*, 1(3):24–32, 2002.

[18] H. Liu, H. Darabi, P. Banerjee, and J. Liu. Survey of wireless indoor positioning techniques and systems. *Systems, Man and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 37(6):1067–1080, 2007.

[19] M. Müller and T. Röder. Motion templates for automatic classification and retrieval of motion capture data. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 137–146, 2006.

[20] L. Rabiner and B.H. Juang. *Fundamentals of speech recognition*. 1993.

[21] N. Ravi, N. Dandekar, P. Mysore, and M. L. Littman. Activity recognition from accelerometer data. In *In Proceedings of the Seventeenth Conference on Innovative Applications of Artificial Intelligence(IAAI)*, pages 1541–1546. AAAI Press, 2005.

[22] D. Reid. An algorithm for tracking multiple targets. *IEEE Transactions on Automatic Control*, 24:843–854, Dec 1979.

[23] M. Riedmiller and H. Braun. A direct adaptive method for faster backpropagation learning: The RPROP algorithm. In *Neural Networks, 1993., IEEE International Conference on*, pages 586–591. IEEE, 2002.

[24] G. Schindler, C. Metzger, and T. Starner. A wearable interface for topological mapping and localization in indoor environments. In Mike Hazas, John Krumm, and Thomas Strang, editors, *LoCA*, volume 3987 of *Lecture Notes in Computer Science*, pages 64–73. Springer, 2006.

[25] C. Toth, D.A. Grejner-Brzezinska, and S. Moafipoor. Pedestrian tracking and navigation using neural networks and fuzzy logic. In *Intelligent Signal Processing, 2007. WISP 2007. IEEE International Symposium on*, pages 1–6. IEEE, 2008.

[26] H. Wang, H. Lenz, A. Szabo, J. Bamberger, and U.D. Hanebeck. WLAN-based pedestrian tracking using particle filters and low-cost MEMS sensors. In *Positioning, Navigation and Communication, 2007. WPNC'07. 4th Workshop on*, pages 1–7. IEEE, 2007.

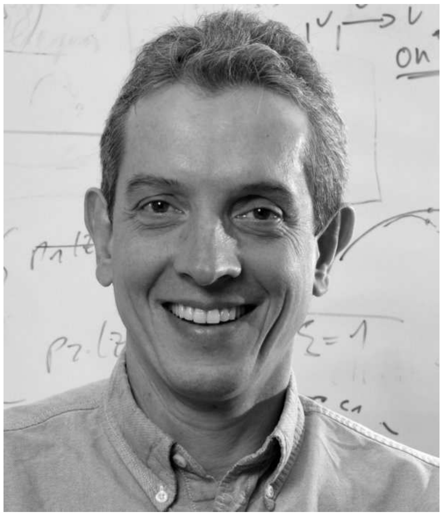
[27] O. Woodman and R. Harle. Pedestrian localisation for indoor environments. In *Proceedings of the 10th international conference on Ubiquitous computing*, pages 114–123. ACM, 2008.

[28] A. Zell, G. Mamier, M. Vogt, N. Mache, R. Huebner, K.-U. Herrmann, T. Soyez, M. Schmalzl, T. Sommer, A. Hatzigeorgiou, S. Doering, D. Posselt, and T. Schreiner. SNNS, Stuttgart neural network simulator, user manual, version 4.1. Technical Report 6/95, University of Stuttgart, 1995.

Slawomir Grzonka studied computer science at the University of Freiburg and received his Master degree in June 2006. He currently is a Ph.D. student in the Autonomous Intelligent Systems Lab of the University of Freiburg. His research interests lie in the areas of unmanned aerial vehicles, simultaneous localization and mapping, and human activity recognition. For his work on autonomous quadrotor systems he received best paper awards from the International Conference on Robotics and Automation (ICRA) in 2009 and from the International Conference and Exhibition on Unmanned Aerial Vehicles (UAV) in 2010.

Andreas Karwath is academic researcher in the research group for Autonomous Intelligent Systems Lab of the University of Freiburg. He received his Ph.D. from the University of Wales in 2003 in the Computational Biology and Machine Learning Lab. Since 2004, he holds a post-doc position at the University of Freiburg. His research areas are machine learning and datamining applied to real world problems. That has led to a number of publications in the field of bioinformatics, chemoinformatics, and activity recognition.

Frederic Dijoux studied computer science at the University of Freiburg and received his Master degree in January 2010. His research interests lie in the areas of human activity recognition and mapping.



Wolfram Burgard is a full professor for computer science at the University of Freiburg, Germany where he heads the Laboratory for Autonomous Intelligent Systems. He received his Ph.D. degree in computer science from the University of Bonn in 1991. His areas of interest lie in artificial intelligence and mobile robots. In the past, Wolfram Burgard and his group developed several innovative probabilistic techniques for robot navigation and control. They cover different aspects such as localization, map-building, path-planning, and exploration. For his work, Wolfram Burgard received several best paper awards from outstanding national and international conferences. In 2009, Wolfram Burgard received the Gottfried Wilhelm Leibniz Prize, the most prestigious German research award. Recently, Wolfram Burgard received an Advanced Grant from the European Research Council.