2012 International Conference on Indoor Positioning and Indoor Navigation, 13-15th November 2012

# ActionSLAM: Using location-related actions as landmarks in pedestrian SLAM

Michael Hardegger*, Daniel Roggen, Sinziana Mazilu, Gerhard Tröster
Wearable Computing Laboratory, ETH Zürich
Zürich, Switzerland
www.ife.ee.ethz.ch
*Corresponding author: michael.hardegger@ife.ee.ethz.ch

***Abstract*—Indoor localization at minimal deployment effort and with low costs is relevant for many ambient intelligence and mobile computing applications. This paper presents ActionSLAM, a novel approach to Simultaneous Localization And Mapping (SLAM) for pedestrian indoor tracking that makes use of body-mounted sensors. ActionSLAM iteratively builds a map of the environment and localizes the user within this map. A foot-mounted Inertial Measurement Unit (IMU) keeps track of the user's path, while observations of location-related actions (e.g. door-opening or sitting on a chair) are used to compensate for drift error accumulation in a particle filter framework. Location-related actions are recognizable from body-mounted IMUs that are often used in ambient-assisted living scenarios for context awareness. Thus localization relies only on on-body sensing and requires no ambient infrastructure such as Wi-Fi access points or radio beacons.**

**We characterize ActionSLAM on a dataset of 1.69km walking in three rooms and involving 241 location-related actions. For the experimental dataset, the algorithm robustly tracked the subject with mean error of 1.2m. The simultaneously built map reflects the building layout and positions landmarks with a mean error of 0.5m. These results were achieved with a simulated action recognition system consisting of an IMU attached to the wrist of a user and a smartphone in his pocket. We found that employing more complex action recognition is not beneficial for ActionSLAM performance. Our findings are supported by evaluations in synthetic environments through simulation of IMU signals for walks in typical home scenarios.**

## I. Introduction

Activity monitoring is commonly used for context-aware home assistance, rehabilitation, or implicit energy management applications ([1], [2], [3]). Besides activities, the user's location is a relevant source of context information in such scenarios [4]. For example by fusing location information to fall detection systems such as the Android application iFall [5], one could objectively create hazard maps, showing areas of frequent falls at home.

In order to minimize costs, decrease deployment effort, and increase user acceptance, it is interesting to avoid location tracking systems that require modifications of the environment, prior expert calibration, or prior maps of the indoor surroundings. This is particularly beneficial for location tracking in private homes and for location-aware consumer wearables.

Simultaneous localization and mapping is an algorithmic framework that can address this by allowing for iterative, autonomous map building of previously unknown environments and localization of a subject within this environment [6]. SLAM combines proprioceptive sensors for keeping track of the subject's motion, and exteroceptive sensors for observing landmarks in the environment. A key characteristic of SLAM implementations is the modality of the landmarks identified by the exteroceptive sensors. The choice of an appropriate sensor setup for landmark recognition influences both, tracking accuracy and usability of the system.

In this work we propose *ActionSLAM*, a SLAM implementation for pedestrian tracking whose novelty is to use location-related actions as landmarks. A single foot-mounted IMU is applied for proprioceptive motion sensing, together with an action recognition system for landmark observations. Location-related actions are body movements that only occur at specific locations in the environment. For example, the door-opening action is always executed close to doors, simply because it can only be executed when a door is in reach. Research in activity and gesture recognition showed that it is possible to identify such actions by applying machine learning techniques for classification of motion sensor readings [7]. Examples include recognizing "door opening" from a wrist-worn IMU ([8], [9]), or "sitting down on a chair" from a waist-mounted accelerometer [10]. In daily life at home, location-related actions occur frequently with respect to distance traveled (e.g. a person stands up from his chair, opens the door, walks to the kitchen to open the fridge, takes out food, walks back to the living room, closes the door, and sits down again). ActionSLAM is an instantiation of the FastSLAM algorithmic framework [11] adapted to handle the characteristics of action landmarks while being robust towards action recognition errors.

We evaluated ActionSLAM in an experiment simulating daily life at home that included $1.69km$ of walking and performing $241$ location-related actions. ActionSLAM was capable of mapping the experimental area with a mean error in landmark positioning of less than $0.5m$ and tracked the user's position with mean error of $1.2m$. Furthermore, we applied ActionSLAM to synthetic motion data of performing daily routines in four different buildings and found similar results as for the real dataset. Our analysis showed that an unobtrusive action recognition system consisting of a wrist-attached motion sensor and a smartphone may be sufficient for reliable mapping and localization at home.

978-1-4673-1954-6/12/$31.00 ©2012 IEEE

## II. Background

### A. Indoor Localization

The state-of-the-art approach to indoor localization of pedestrians is radio-frequency (RF) fingerprinting, either using Wi-Fi or GSM [12]. RF fingerprinting is a two-stage process: In an offline mapping phase, fingerprints are recorded at known positions to build a prior map. Then, in the online localization phase, RF signal strengths are measured and compared to the fingerprints in the pre-recorded map. The effort of building and maintaining a prior map is acceptable in public indoor environments with many users of a single map, but a drawback for deployment in private scenarios, where a map is only used by one or few people.

A set of tracking methods known as Pedestrian Dead Reckoning (PDR) does not require prior mapping and is therefore suitable for exploration tasks in previously unknown environments. In PDR, position is inferred from wearable motion sensors. The highest accuracy reported in literature is reached with foot-mounted inertial measurement units and so-called Zero-Velocity Updates (ZUPT-PDR). The observation of having the foot on the ground and therefore velocity zero is used to compensate for error accumulation when integrating the IMU acceleration and rotational velocity measurements to position ([13], [14]). An alternative is to combine step detection and heading estimation from body-attached smartphones to infer the user's path [15]. All PDR methods suffer from error accumulation with time and are therefore not intended for long-term pedestrian tracking.

### B. SLAM

SLAM combines odometry measurements (in human tracking typically PDR) with exteroceptive sensing to simultaneously estimate a mobile agent's pose $s_t$ (location $x_t = \{s_t^x, s_t^y\}$ and heading $s_t^\phi$) and build a spatial map $\Theta$ of his environment [6]. SLAM was originally introduced in robotics, but recently implementations for human tracking were presented (see Section II-C). The goal in SLAM is to find the pose $\hat{s}_t$ and map $\hat{\Theta}$ that maximize the following probability density function:

$$p(s_t, \Theta | u^t, z^t, n^t) \tag{1}$$

At time $t$, the proprioceptive sensors of the SLAM system provide a motion update measurement $u_t$, with $u^t = u_1, \ldots u_t$ being the history of measurments. At the same time, landmarks in the environment are observed. $z_t$ is a measurement of the landmark's position relative to the current user position. $n_t$ is the identifier of the landmark observed at time $t$ (in general, multiple landmarks may be observed at a time). $z^t = z_1, \ldots z_t$ is the history of landmark position observations, and $n^t = n_1, \ldots n_t$ the history of landmark identifications.

The probability density function $p(s_t, \Theta | u^t, z^t, n^t)$ can be estimated by means of an extended Kalman filter (EKF) and two independent models for position update $p(s_t | u_t, s_{t-1})$ and observation update $p(z_t | s_t, \theta_{n_t}, n_t)$. $\theta_{n_t}$ is the location of the landmark $n_t$ observed at time $t$. This approach is known as EKF-SLAM. It has been successfully applied in many robotics applications, e.g. [16].

An alternative is to estimate the posterior over all paths $s^t = s_1, \ldots, s_t$ instead of just the current poses $s_t$. The resulting algorithmic framework is known as FastSLAM [11], with a detailed discussion given by Thrun et al. in [17]. This framework can be instantiated to various kinds of motion and landmark sensing modalities by providing appropriate error models $p(s_t | s_{t-1}, u_t)$ and $p(z_t | s_t, \Theta)$. We use the notations of [17] throughout this work. Estimating the posterior over all paths enables the following factorization:

$$p(s^t, \Theta | z^t, u^t, n^t) = p(s^t | z^t, u^t, n^t) \prod_{n=1}^{N_L} p(\theta_n | s^t, z^t, n^t) \tag{2}$$

In this way, the SLAM problem is decomposed into seperate estimators for the person's path $s^t$ and each of the $N_L$ landmark locations. In FastSLAM, the path probability density function $p(s^t | z^t, u^t, n^t)$ is calculated by a particle filter, while the $N_L$ landmark location probability density functions $p(\theta_n | s^t, z^t, n^t)$ are estimated with EKFs. Since the landmark locations are conditioned on the person's path, each particle (indexed with $[m]$) must maintain its own set of landmarks $\Theta^{[m]}$ together with the current path estimation $s^{t,[m]}$.

A difficulty that often arises in SLAM is to uniquely associate observations to a landmark: In practice, the landmark identifier $n_t$ is often not known or uncertain since multiple landmarks may look identical. In EKF-SLAM, only the most likely data association hypothesis is tracked - if this association is incorrect, EKF-SLAM often fails to converge. FastSLAM has multiple advantages over EKF-SLAM such as lower computational requirements and the capability of coping with non-linear motion models. However, the key advantage is that FastSLAM is capable of multi-hypothesis tracking. The particle filter of FastSLAM can draw samples following multiple hypotheses and filters out incorrect data associations at a later stage, when more observations are available [18].

### C. Landmarks in pedestrian SLAM

Both EKF-SLAM and FastSLAM have been applied to human indoor tracking, making use of different exteroceptive sensing modalities. In Table I a selection of published work on SLAM applied to human indoor tracking is presented and differences in terms of position update strategy, landmark type and algorithmic framework are listed.

So far, visual and Wi-Fi landmarks have been applied in pedestrian SLAM. FootSLAM [23] and WiSLAM [25] do not use landmarks in the map representation, but instead count transitions between hexagonal map elements, assuming that some areas of the environment are traversed in only one direction and its opposite (e.g. floors). Computational requirements and obtrusive sensor placement are major limitations of visual landmarks for use in daily life [19]. Wi-Fi landmarks are computationally much more lightweight and offer room-level tracking accuracy if sufficient Wi-Fi coverage is available

TABLE I
PEDESTRIAN SLAM IMPLEMENTATIONS

| Researchers | PDR Method (Sensor) | Observation (Sensor) | Algorithmic framework | Evaluation results |
|---|---|---|---|---|
| Pradeep et al. [19] | Stereo vision odometry (stereocam) | Visual object recognition (stereocam) | FastSLAM | High-accuracy tracking, large computational and storage requirements |
| Ferris et al. [20] | Probabilistic motion model (no sensor) | Wi-Fi landmarks (Wi-Fi radio) | Gaussian process latent variable model (offline) | Mean error of $4m$ in offline analysis |
| Shin et al.: SMARTSLAM [21], [22] | Step counting and heading determination (smartphone in pocket) | Wi-Fi landmarks (Wi-Fi radio) | FastSLAM | Mean error of $3m$ for a smartphone implementation |
| Robertson et al.: FOOTSLAM [23], [24] | ZUPT-PDR (foot-mounted IMU) | Walking patterns (foot-mounted IMU) | Particle filter (similar to FastSLAM) | Mean error of $2m$, but requires a desktop computer for real-time execution |
| Bruno and Robertson: WISLAM [25] | ZUPT-PDR (foot-mounted IMU) | Walking patterns and Wi-Fi (foot-mounted IMU and Wi-Fi radio) | Particle filter (similar to FastSLAM) | Increased robustness of FootSLAM |

[22]. Providing maps and simultaneous localization with in-room accuracy on a low-power wearable platform meanwhile remains an open challenge, which we address with this work.

## III. ACTIONSLAM ALGORITHM

ActionSLAM is a specific instantiation of the FastSLAM framework optimized to operate with action landmarks. Action landmarks are different from visual or Wi-Fi landmarks in two ways: *(i)* the action associated to an action landmark is only observed when the user is at the location of this landmark, therefore resulting in trivial relative position measurements $z_t$, and *(ii)*, landmarks possess a type $a_n$ related to the activity performed at this landmark. Action recognition does not provide the identifier $n$ of an observed landmark, but it provides the type of the observed landmark, which simplifies the data association step in FastSLAM.

Figure 1 depicts the main blocks of ActionSLAM. A foot-mounted IMU is used for both, proprioceptive motion sensing and detection of location-related actions. Additional body-mounted sensors allow for recognition of more complex location-related actions.

### A. Position update

In ActionSLAM the beginning of a stance phase (i.e. when the foot is on the ground and stops moving) triggers a position update. To detect the beginning of a stance phase from the foot-mounted IMU, we apply threshold-based stance-phase detection as proposed by Jiménez et al. [14]. The index $t$ identifies the corresponding stance phase and the previous stride phase (when the foot was in the air). Steps $u_t = \binom{l_t}{\phi_t}$ with length $l_t$ and heading change $\phi_t$ bring the user from pose $s_{t-1}$ to $s_t$. We compute $\hat{u}_t = \binom{\hat{l}_t}{\hat{\phi}_t}$, an estimate of $u_t$, by applying quaternion-based ZUPT-PDR as in [26].

Given $\hat{u}_t$ we draw particles $[m]$ according to:

$$s_t^{[m]} \sim p(s_t^{[m]} | s_{t-1}^{[m]}, \hat{u}_t) \tag{3}$$

As a consequence, each particle will represent a variation of the measured ZUPT-PDR path. For notational simplicity we leave away the particle index $[m]$ subsequently. When assuming Gaussian-distributed PDR errors for both $\hat{l}_t$ and $\hat{\phi}_t$, Equation 3 can be rewritten as follows:

$$\begin{aligned} L_t &\sim N(\hat{l}_t, \sigma_l) && (4)\\ \Phi_t &\sim N(\hat{\phi}_t + \mu_{\phi,t}, \sigma_\phi) && (5)\\ s_t^x &= s_{t-1}^x + L_t \cos(s_{t-1}^\phi + \Phi_t) && (6)\\ s_t^y &= s_{t-1}^y + L_t \sin(s_{t-1}^\phi + \Phi_t) && (7)\\ s_t^\phi &= s_{t-1}^\phi + \Phi_t && (8) \end{aligned}$$

Here, $L_t$ is the particle's sampled step length and $\Phi_t$ the heading change in the particle's pose during the step. For heading change, we empirically found that it is beneficial to include a random walk error $\mu_{\phi,t} \sim N(\mu_{\phi,t-1}, \sigma_{\mu_\phi})$ that compensates for slow heading drift errors of PDR which we often observed in preliminary recordings. These errors might be caused by both, random walk contributions of IMU noise, and displacements of the wearable sensor.

### B. Observation update

At any time, location-related actions may occur and we assume an action recognition system can detect $N_A$ different types of such actions. For simplicity, we assume that only one location-related action $\hat{a}_t \in \{A_1, \dots A_{N_A}\}$ can be observed during stride and stance phase $t$. Furthermore, $\hat{a}_t$ is considered as the null-class action $A_0$ if no specific location-related action was recognized. If $\hat{a}_t = A_0$, no observation update is performed and the subsequent data association and resampling steps are skipped.

If however $\hat{a}_t$ is a known location-related action $A_{i|i\geq 1}$, we perform an observation update for each particle. In ActionSLAM, landmark observations are made up of *i)* the action $\hat{a}_t$ that was recognized, and *ii)* an estimation $z_t$ of the position difference between the landmark with identifier $n_t$ and the current user position $x_t = \{s_t^x, s_t^y\}$. Since action landmarks are only observed when the user is performing the action (and therefore at the location $\theta_{n_t}$ of the landmark), the relative landmark position observation in ActionSLAM is:

$$z_t = \theta_{n_t} - x_t = 0 \tag{9}$$

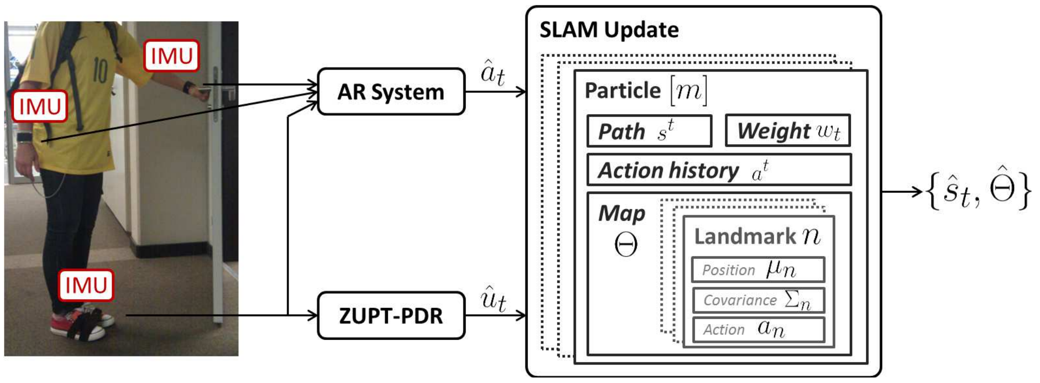

Fig. 1. The main blocks of ActionSLAM: ZUPT-PDR takes the readings of a foot-mounted IMU to estimate step length and heading change, while a parallel Action Recognition (AR) system detects actions from the foot-mounted IMU and further sensors attached to the user. Within the particle filter, multiple hypotheses about the user's path and actions are tracked and evaluated against each other. The currently best hypothesis is the output of the tracking system. The type $a_n$ of the landmark $n$ depicted here would be "door opening".

Each particle in ActionSLAM possesses an internal map $\Theta = \{\{a_1, \mu_1, \Sigma_1\}, \ldots, \{a_{N_L}, \mu_{N_L}, \Sigma_{N_L}\}\}$ with landmarks made up of the associated action $a_n$ (i.e. the landmark type) and a Gaussian landmark position estimation with mean $\mu_n$ and covariance matrix $\Sigma_n$. The differences between the particle's position and each of the landmark positions in the particle's map are calculated as follows:

$$\hat{z}_n = g(\mu_{n,t-1}, x_t) = \mu_{n,t-1} - x_t \tag{10}$$

For ActionSLAM, the Jacobian $G_n$ of $g(\mu_{n,t-1}, x_t)$ with respect to the landmark positions $\mu_n$ is simply the identity matrix. As a result, many of the original FastSLAM formulas can be simplified. The landmark observation covariance matrix is given by:

$$Q_n = \Sigma_{n,t-1} + R_t \tag{11}$$

$R_t$ is the measurement covariance matrix. In practice, we do not only have uncertainty about the location $\mu_n$ of a landmark $n$, but since any action recognition systems is prone to errors, also about its action type $a_n$ and the currently performed action $a_t$. In activity and gesture recognition, this uncertainty is typically described by a confusion matrix where each entry indicates the probability $p(a_t|\hat{a}_t)$ of the user having performed action $a_t$ when $\hat{a}_t$ was observed.

Given this confusion matrix and the position uncertainty $Q_n$, we calculate for each landmark $n$ in the particle's map the likelihood of correspondence with the action observation at stance phase $t$ as follows (for derivation see [11]; $\eta$ is a normalization factor):

$$p_n = \eta |2\pi Q_n|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}\hat{z}^T Q_n^{-1} \hat{z}\right) \cdot p(a_t|\hat{a}_t) \tag{12}$$

We furthermore assume that with likelihood $\eta \cdot p_0$, the current observation is unrelated to any landmarks in the particle's map and constitutes a new, previously unobserved landmark. For each of the $N_A$ actions types, the probability that a newly observed landmark has this action type is calculated as follows:

$$p_n = \eta p_0 \cdot p(a_t|\hat{a}_t) \tag{13}$$

## C. Data association and landmark update

We now sample the estimated landmark identifier $\hat{n}_t$ from the previously calculated probabilities $p_n$ of having observed landmark $n$ at step $t$. If the particle's map contains $N_L$ landmarks and $\hat{n}_t$ is smaller or equal $N_L$, we associate the observation to a previously known landmark and update the landmark position and covariance matrix based on the new observation, following the formulas for a Kalman filter update:

$$\begin{aligned} K &= \Sigma_{\hat{n},t-1} Q_{\hat{n}}^{-1} && (14) \\ \mu_{\hat{n},t} &= \mu_{\hat{n},t-1} - K \hat{z}_{\hat{n}}^T && (15) \\ \Sigma_{\hat{n},t} &= (I - K)\Sigma_{\hat{n},t-1} && (16) \end{aligned}$$

If $\hat{n}_t > N_L$ the particle tracks the assumption that a new, previously unseen landmark was observed for the first time. We therefore initialize a new landmark:

$$\begin{aligned} \mu_{N_L+1,t} &= s_t && (17) \\ \Sigma_{N_L+1,t} &= R_t && (18) \\ a_{N_L+1} &= a_t && (19) \end{aligned}$$

## D. Resampling

Each particle $[m]$ has a weight $w^{[m]}$, which is updated after each observation of a non-null class action following this formula:

$$w_t^{[m]} = w_{t-1}^{[m]} \cdot p_{\hat{n}}^{[m]} \tag{20}$$

The weight describes the confidence in each particle's hypothesis. Once all particle weights have been updated, the particle $[b]$ with highest weight $w_t^{[b]} \geq w_t^{[m]} \forall [m]$ is chosen as the current best guess with $\hat{s}_t = s_t^{[b]}$ being the current ActionSLAM position estimate and $\hat{\Theta} = \Theta^{[m]}$ the map estimate. In the following, $\hat{s}^t = \hat{s}_1, \ldots, \hat{s}_t$ is denoted as the

TABLE II
LIST OF THE LOCATION-RELATED ACTIONS THAT OCCURED IN THE EXPERIMENT AND THE SENSOR SETUPS THAT MAY RECOGNIZE THESE ACTIONS.

| **Abbr.** | **Multimodal** | **Rich IMU** | **Basic IMU** | **Smartphone** | **Foot IMU** |
|---|---|---|---|---|---|
| | *Same as Rich IMU, plus physilogical sensors (e.g. EOG)* | *Same as Basic IMU plus addition IMUs at other wrist, hip, etc.* | *Same as Smartphone plus one wrist-attached IMU* | *Same as Foot IMU plus smartphone in pocket* | *Single IMU on foot* |
| W1 | Writing (paper) | Writing | Sitting, few hand movements (Si1) | Sitting | Not moving |
| W2 | Writing (PC) | | | | |
| R1 | Reading (single page) | Reading | | | |
| R2 | Reading (Newspaper) | | | | |
| E1 | Eating (bread) | Eating | Sitting, many hand movements (Si2) | | |
| E2 | Eating (chocolate) | | | | |
| DW | Drinking water | Drinking | | | |
| AP | Answer phone | Calling | Standing, few hand movements (St1) | Standing | |
| WT | Using water tap | Using water tap | | | |
| WP | Watering plants | Watering | | | |
| BB | Putting butter on bread | Preparing food | | | |
| BT | Brushing teeth | Brushing teeth | Standing, many hand movements (St2) | | |
| CB | Cutting bread | Preparing food | | | |
| OCDo | Open/close door | Open/Close | Open/Close (OC) | | |
| OCDr | Open/close drawer | | | | |
| OCF | Open/close fridge | | | | |
| OCW | Open/close window | | | | |

ActionSLAM path estimate. As for all instances of FastSLAM, ActionSLAM can post-correct paths since particles store the full path information $s^t$ at any time. The path $s^{t,[b]}$ associated to the particle $[b]$ is denoted as the posterior path estimate in this work. After a while, some particles will have very low weights. To get rid of these particles, we apply systematic resampling [27] whenever the effective number of particles $N_{eff} = \frac{1}{\sum_{m=1}^{N_p}(w_t^{[m]})}$ is below a threshold.

## IV. DATASET

### A. Experimental setup

To evaluate ActionSLAM on real data, we set up a recording simulating daily life at home including activities such as eating breakfast, opening the windows, working in the home office, etc. The subject was instructed to fulfill tasks that induced location-related actions. For example, the task of eating a slice of bread involved opening the fridge to take out butter, getting a plate from a drawer, cutting the bread, and finally, eating while sitting at the kitchen table. See Table II for a list of the 17 action types that were induced by the 13 tasks in the experimental setup. The recording took about $70min$, involved $1.69km$ of walking and $241$ location-related actions. The setup involved three adjacent rooms (a kitchen and two office rooms) and the floor connecting them. The subject left this area five times to use a water tap located twenty meters from the kitchen.

We recorded the foot motion with an Xsens MTx sensor (www.xsens.com) placed at the right foot of the subject, sampling acceleration and rotational velocity at $100Hz$. Additional motion sensors were attached to the upper body and the waist. They are not used in this paper, but foreseen for future evaluations of action recognition performance.

To obtain reliable ground truth location data, we recorded the interior of the three main rooms of the experiment using three GoPro HD Hero2 cameras with 170-degrees fisheye lenses. The fisheye effect was removed with the Adobe After-Effects optical compensation tool. The same video processing software was also applied for stitching together the videos of the three cameras and rendering a geometrically aligned overview of the experimental area. In the resulting video we tracked the position of the sensorized foot by mouse. Parts of the experimental area were not covered by the three cameras, so we restricted our ground truth analysis to the visible area.

The actions of the subject were annotated online with a labeling tool. These labels, the video streams and the motion sensor data were synchronized during post-processing.

### B. Simulating action recognition

This work focuses on evaluating the feasibility of using actions as landmarks. Therefore we did *not* perform action recognition from motion sensor data, but used the online annotations given during the recording. However, the actions were chosen in accordance with gesture and activity recognition literature such as [7], [8] and [9], and we simulated different setups and recognition accuracies based on literature.

First, we analyzed five levels of sophistication in the action recognition system. Sensor-rich setups allow for detection and categorization of subtle actions. When the number of wearable sensors is reduced, the action types that can be differentiated tend to be coarser, although the user comfort may improve. The different granularity levels and a description of the involved sensor setups is given in Table II. The only exception where we performed actual action recognition is the *not-moving* action, which we detected whenever the foot was still for more than $2s$. This action can be detected from the foot-mounted IMU alone and it constitutes the simplest possible implementation of ActionSLAM.

For each sensor setup we also simulated action recognition errors based on typical confusion matrices reported in literature, e.g. [8] and [9]. Table III shows the confusion matrix for a *Basic IMU* setup as it was used for our work.

TABLE III
CONFUSION MATRIX FOR SIMULATING NON-IDEAL ACTION RECOGNITION.

| | null class | Si1 | Si2 | St1 | St2 | O/C |
|---|---|---|---|---|---|---|
| null class | 0.9 | 0 | 0 | 0.02 | 0.02 | 0.06 |
| Si1 | 0 | 0.8 | 0.2 | 0 | 0 | 0 |
| Si2 | 0 | 0.2 | 0.8 | 0 | 0 | 0 |
| St1 | 0.02 | 0 | 0 | 0.7 | 0.13 | 0.15 |
| St2 | 0.02 | 0 | 0 | 0.13 | 0.7 | 0.15 |
| O/C | 0.06 | 0 | 0 | 0.15 | 0.15 | 0.64 |

### C. *Landmarks*

Each of the actions in Table II could take place at one or multiple locations in the experimental area. From the experimental design we derived a ground truth map with action landmarks placed at the position where we expected the corresponding actions to take place. In total, 23 landmarks were inserted in this ground truth map. A subsection of the full map is depicted in Figure 2. At some locations, multiple actions could take place. For example, at the kitchen table, the subject could eat bread, eat chocolate or read the newspaper. Depending on the sensor setup used in the analysis, this location had either three associated landmarks (one for each action), two (for the *Basic IMU* setup), or only a single landmark (a landmark with action "sitting" in the *Smartphone* setup, and a "not-moving" landmark for the *Foot-IMU* setup).

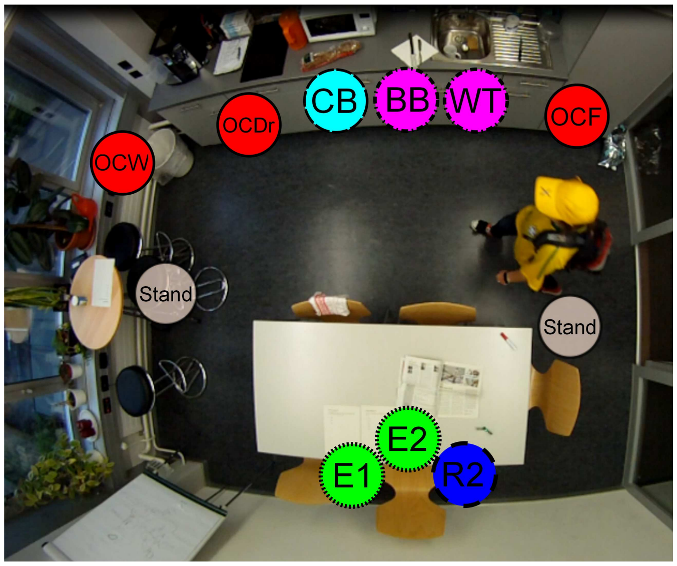

Fig. 2. This figure shows the kitchen section of the experimental area with the action landmarks as defined for the ground truth map. The colors and border marks indicate the association to the different action types for a *Basic IMU* sensor setup as described in Table II. The additional "stand" landmarks were not used for analysis, but depict locations where the subject often stood still for a while (and "not-moving" was detected by the foot-mounted IMU).

## V. RESULTS

### A. *Performance measurements*

Figures 3, 4 and 5 depict results for a single analysis run. We evaluated ActionSLAM according to the following measures:

- **Path accuracy:** For each stance phase, we calculated the euclidean distance between the estimated user position $\hat{x}_t = \{\hat{s}_t^x, \hat{s}_t^y\}$ and the ground truth position $x_t$. As shown in Figure 4, we observed a typical pattern in $\Delta\hat{x}_t = |\hat{x}_t - x_t|$ with a maximum in the first $300m$ (where the map was yet unknown) and a subsequent reduction and convergence of the positioning error (when landmarks were revisited). To account for this two-phase behaviour, we report both maximum tracking error and the mean tracking error during the last hundred meters of walking in a recording. Path accuracy was measured for both, the ActionSLAM path estimation $\hat{x}^t$ and the posterior path estimation $x^{t,[b]}$.
- **Map accuracy:** At every stance phase $t$, we also calculated the landmark positioning errors for the current SLAM map $\hat{\Theta}_t$. For each ground truth landmark, we checked for landmarks of identical action type within a minimal distance $d_{min} = 2m$. If no landmark was found, we counted the landmark as unobserved. For map accuracy, we calculated the mean distance between ground truth position $\theta_n$ and SLAM landmark position $\hat{\mu}_n$ of the observed landmarks only. Landmarks in the SLAM map that could not be associated to any ground truth landmarks were counted as insertion errors.
- **Robustness:** We considered an ActionSLAM algorithm execution to be successfull when the mean posterior path error was below $1m$. To account for the probabilistic nature of ActionSLAM we measured robustness as the percentage of successfull ActionSLAM runs with identical settings and inputs.

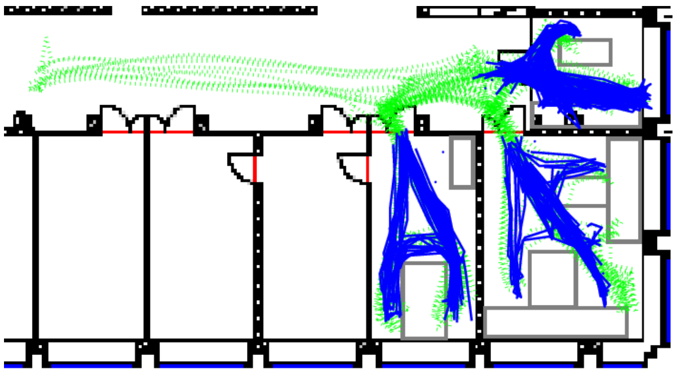

Fig. 3. The posterior path $x^{t,[b]}$ as estimated by ActionSLAM (dotted green), overlayed with the ground truth path where available (solid blue). The kitchen in Figure 2 is the upper room on the right side.

Note that SLAM maps and paths may be rotated, translated and scaled versions of the actual map and path. We therefore applied interior-point optimization to find the best fit between the ground truth path and $x^{t,[b]}$ at the end of the recording before calculating the performance measurses. The same rotation, translation and scaling factors were also applied to the map $\hat{\Theta}$ and the ActionSLAM track $\hat{x}^t$.

### B. *Parameter Optimization*

Before evaluating the algorithm performance on the dataset, we identified and investigated the most relevant ActionSLAM

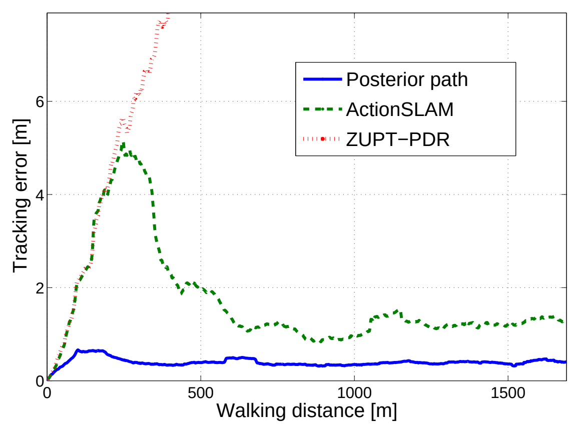

Fig. 4. Mean filtered tracking errors for the PDR path (dotted red), the ActionSLAM path (dashed green) and the posterior path $x^{t,[b]}$ (solid blue). During the first map exploration phase, the tracking error for ActionSLAM increases at a similar rate as for PDR. Then, after approximately $300m$ of walking, the subject started to revisit locations and the tracking error decreased to less than $1.2m$.

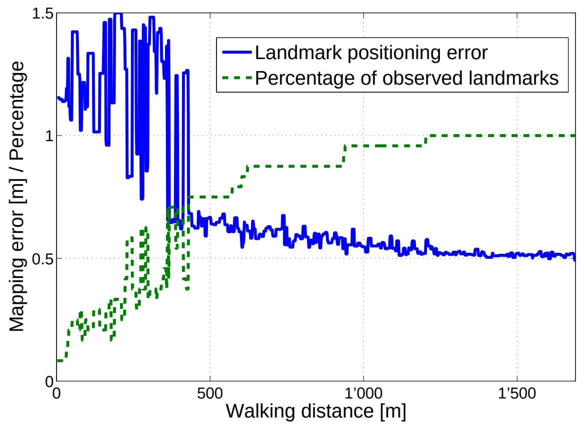

Fig. 5. Percentage of identified landmarks (dashed green) and the mean distance between the already identified landmarks and the corresponding ground truth landmarks (solid blue). The map error continuously decreases to about $0.51m$. After $1180m$ of walking all landmarks have been identified.

parameters. We optimized the parameters in terms of mean path accuracy for $\hat{x}^t$, given that the algorithm converged. To account for the probabilistic nature of ActionSLAM, we averaged tracking errors over 10 repeated analysis runs with identical parameters.

The parameter optimization results reported subsequently were found for the *Basic IMU* setup and assuming perfect action recognition. For other sensor setups, similar values were found. Therefore we consistently use the listed values throughout the evaluation.

- **Prediction update:** We performed a joint parameter sweep for $\sigma_\phi$ and $\sigma_{\mu_\phi}$ as well as an independent sweep for $\sigma_l$. The analysis showed that ActionSLAM is not sensitive to $\sigma_l$ and we fixed it to $0.01m$. The optimal heading change parameters were found to be $\sigma_\phi = 0.8°$ and $\sigma_{\mu_\phi} = 0.12°$. For lower heading change parameters, ActionSLAM did not always converge, while larger heading change parameters caused an increase in mean tracking error.
- **Observation update:** In general, the measurement covariance matrix $R_t$ is dependent on the observed landmark's action type. For simplicity, we restricted the analysis to a single matrix of the following form:

$$R_t = \begin{pmatrix} d_0^2 & 0 \\ 0 & d_0^2 \end{pmatrix}$$

  In a parameter sweep we found decreased SLAM robustness for $d_0 \leq 0.2m$ and $d_0 \geq 0.35m$. As a consequence we chose $d_0 = 0.25m$ for the further evaluation.
- **Data association:** The probability $p_0$ describes the likelihood of observing previously unknown landmarks with respect to the likelihood of reobserving a known landmark. It affects the number of incorrectly inserted landmarks when action recognition errors occur and consequently robustness of ActionSLAM. We performed a parameter sweep with non-ideal action recognition and found $p_0 = 0.1$ to be the best choice in terms of maximizing robustness.
- **Number of particles:** For an ideal action recognition system, ActionSLAM reliably converged with as few as $N_p = 250$ particles. For the $70min$ data stream, the average computation time for our Matlab implementation of ActionSLAM on a Lenovo T410 laptop (2.66 GHz) was $130.6s$.

## C. Performance analysis with ideal action recognition

In Table IV we summarize the mapping and path accuracy results averaged over 10 runs with the parameters found from the previous optimization and a *Basic IMU* setup. A video showing one of these runs was uploaded to https://vimeo.com/47293752. We repeated the analysis for the same dataset but different starting points, therefore simulating slight variations of the original scenario. ActionSLAM consistently produced accurate maps and allowed tracking with accuracy of $1.16\pm0.05m$ after initial map learning. The mean landmark positioning error was $0.46\pm0.04m$ and the posterior path mean error $0.41\pm0.04m$. We assume that the ground truth errors due to inaccurate optical correction and tracking are in a similar range. ActionSLAM converged for all runs with $N_p = 250$ particles and was considered as robust in this scenario and with ideal action recognition.

We compared the ActionSLAM performance for different sensor setups to find an ideal trade-off between tracking accuracy and system obtrusiveness. Figure 6 summarizes the main results for each sensor setup, again averaged over 10 runs. We found that the performance in terms of mean tracking error and map accuracy was similar for all setups, including the *Foot-IMU*-only setup. The similar performance is probably a consequence of the distance between landmark locations, which was mostly significantly larger than $d_0$. In that case, the additional type information of action landmarks was not

TABLE IV
PERFORMANCE AVERAGED OVER 10 RUNS (ACTIONSLAM CONVERGED FOR ALL RUNS). REPORTED NUMBERS ARE FOR A *Basic IMU* SENSOR SETUP.

| **Analysis start [m]** | **0** | **200** | **400** |
|---|---|---|---|
| Mean tracking error [m] | $1.18 \pm 0.05$ | $1.13 \pm 0.04$ | $1.16 \pm 0.07$ |
| Maximum tracking error [m] | $3.09 \pm 0.73$ | $1.48 \pm 0.15$ | $1.52 \pm 0.23$ |
| Posterior path mean tracking error [m] | $0.43 \pm 0.05$ | $0.39 \pm 0.02$ | $0.42 \pm 0.04$ |
| Posterior path maximum tracking error [m] | $0.75 \pm 0.17$ | $0.52 \pm 0.07$ | $0.55 \pm 0.19$ |
| Mean landmark positioning error [m] | $0.48 \pm 0.05$ | $0.44 \pm 0.02$ | $0.46 \pm 0.04$ |
| Number of erroneously inserted landmarks | 4.5 | 2 | 1.9 |

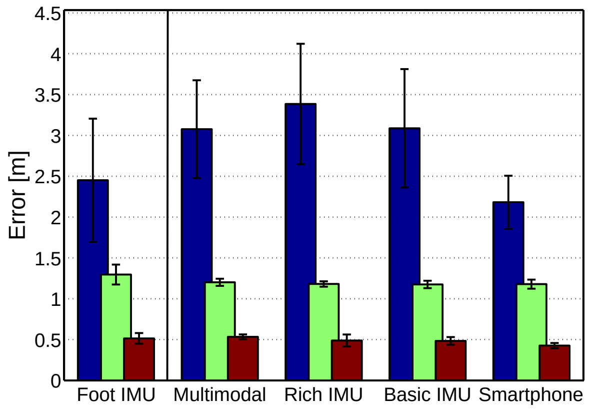

Fig. 6. Performance comparison for different sensor setups in terms of maximum ActionSLAM path error (left, blue), mean ActionSLAM path error (middle, green) and posterior map accuracy (right, red) are depicted. SLAM converged for all runs. For the *Foot-IMU* setup, actual action recognition was performed (i.e. detection of *not-moving*), while ground truth labels were used for the other setups. While the algorithm performed similar for all sensor setups, the maximum tracking errors for simple setups was lower then for sensor-rich settings

necessary for landmark identification with an ideal action recognition system.

In terms of maximum tracking errors, the simple *Smartphone* setup and the *Foot-IMU* setup outperformed the sensor-rich settings. The reason for this is that simple action recognition systems distinguished fewer landmarks, but observed these landmarks more frequently. For example, the actions E1, E2 and R2 all took place at the same location (see Figure 2), but were considered as observations of three different landmarks (with same location, but different type) in a sensor-rich setup. For a *Smartphone* setup, all these actions were associated to a single landmark. Due to the more frequent observations, the algorithm converged faster and the map learning phase was shorter, which resulted in lower maximum errors.

This indicates that a simple, unobtrusive setup may be sufficient for accurate and robust mapping of indoor environments. In general, we expected simpler setups to fail if actions take place at close (distance between landmarks smaller than $d_0$), but different locations. As an example, consider the action landmarks of types WT and BB in Figure 2. A basic sensor setup is not able to distinguish the two landmarks and might introduce errors in the map due to incorrect data association. For the experimental dataset, the analysis showed that this was not an issue with ideal action recognition.

## *D. Performance analysis with action recognition errors*

In the previous discussion we assumed an ideal action recognition system and found that simple sensor setups are sufficient for robust and accurate tracking with ActionSLAM. In this section we analyze the influence of non-ideal action recognition on ActionSLAM performance. Table V reports averaged results of an analysis with simulated action recognition errors (based on the confusion matrix in Table III) and a *Basic IMU* sensor setup. As expected we observed that many landmarks were added to the map due to insertion errors of action recognition. Most of these landmarks were only observed once during the experiment. By removing landmarks from the map when they are not observed for a minimum time, we could avoid this erroneous landmark insertion in future.

The main difficulty that arised as a consequence of action recognition errors was that with $N_p = 250$, ActionSLAM often failed to converge. In Figure 7 we show the percentage of ActionSLAM runs that failed as a function of $N_p$. As we can observe, for $250$ particles, the algorithm converged for less than $90\%$ of the runs. On the other side, even with more than $1000$ particles, there were still a few runs where ActionSLAM did not converge. This indicates that for some critical error insertion patterns, ActionSLAM is not capable of recovering. Fusion with other modalities such as Wi-Fi might improve robustness in such cases.

As for ideal action recognition, we analyzed the performance of different sensor setups (with identical confusion matrix). We found that incorrect landmark insertion has a stronger effect on robustness for simple setups, most likely due to error-prone data association. While robustness for a *Foot IMU* setup was only $77\%$ with $N_p = 250$, it increased to $85\%$ and $90\%$ for the *Smartphone* and *Basic IMU* setups. This confirms that the landmark type information helps to filter out incorrect action

TABLE V
PERFORMANCE ANALYSIS WITH ACTION RECOGNITION ERRORS AVERAGED OVER 100 RUNS FOR $N_p = 250$ AND 10 RUNS FOR $N_p = 2500$.

| **Number of particles** $N_p$ | **250** | **2500** |
|---|---|---|
| Mean tracking error [m] | $1.27 \pm 0.11$ | $1.21 \pm 0.09$ |
| Maximum tracking error [m] | $2.63 \pm 0.62$ | $2.39 \pm 0.56$ |
| Posterior path mean tracking error [m] | $0.48 \pm 0.14$ | $0.43 \pm 0.07$ |
| Posterior path maximum tracking error [m] | $1.36 \pm 0.88$ | $0.82 \pm 0.45$ |
| Mean landmark positioning error [m] | $0.70 \pm 0.11$ | $0.62 \pm 0.10$ |
| Number of erroneously inserted landmarks | 46 | 42 |
| Number of not-converged analyses (not included above) | $15/100$ | $1/10$ |

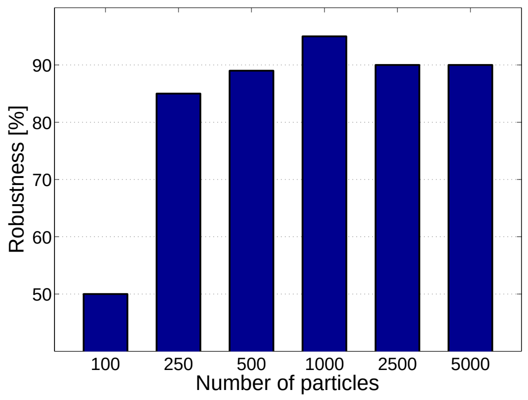

Fig. 7. Percentage of converged ActionSLAM runs. For $N_p \leq 500$, 100 runs each were executed. For $N_p = 1000$, we performed 20 ActionSLAM runs and for $N_p > 1000$ 10 ActionSLAM runs each.

observations and therefore is valuable for ActionSLAM. For sensor-richer setups, no further improvement was observed. We concluded that a *Smartphone* or a *Basic IMU* setup are good trade-offs between system obtrusiveness and robustness, although further evaluation is necessary.

## VI. Synthetic Data Analysis

### A. Generation of synthetic IMU data

Many of the influences on algorithm performance can not be tested with a single dataset. These include building layout, number of landmarks, sequence of performed actions etc. To generalize the results of our experiment, we developed a simulation tool for generation of IMU signals from a building layout and a sequence of activities. First, we fit a path that corresponds to a pre-defined action sequence into the building layout. Location-related actions take place uniformly distributed within a radius of $0.2m$ of pre-defined landmark locations with the same action type. The resulting path is then divided into steps. Since ZUPT-PDR works as an integrator in the stride phase and we assume purely additive noise, we used a very basic approximation of step acceleration patterns as given in [28]. At turns, we added a single non-zero sample to the gyroscope signal with full heading change $\phi = \Delta T \cdot \omega$. IMU errors were added according to the sensor datasheets. For the results presented here we used the error models for Xsens MTx sensors that are described in [29]. The validity of this simulation was confirmed by measuring the PDR error accumulation, which turned out to be similar to the PDR performance we obtained for preliminary recordings with foot-mounted IMUs (in the range of 1% of the distance traveled, mostly caused by heading drift).

### B. Building layout analysis

A simulation-based analysis was done for four different building layouts and action sequences inspired by daily life. As can be seen from Figure 8, the ActionSLAM posterior path accurately reflects the building layout in all scenarios. The mean ActionSLAM path error for a *Basic IMU* setup averaged

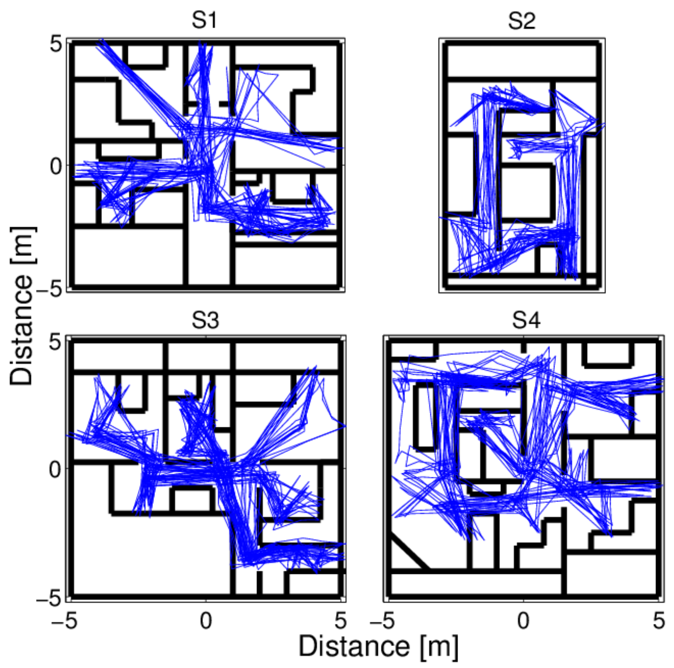

Fig. 8. Building layouts used for simulation of walks in daily life with the overlayed posterior paths as found by ActionSLAM. The total walking distance was between $476m$ for Scenario S2 and $960m$ for S3, with the number of landmarks being between 19 (S2) and 31 (S3). The same action types as for the real dataset were distinguished.

over all maps and ActionSLAM runs (for each map, results for 5 different action sequences an 10 ActionSLAM runs per action sequence were taken into account) was $1.03 \pm 0.42m$, the mean posterior path error $0.61 \pm 0.53m$ and the mean map accuracy $0.55 \pm 0.43m$, which is comparable to what we found for the real dataset. When action recognition errors were introduced, the robustness dropped to $75\%$ for Scenario S1, to $85\%$ for Scenarios S2 and S4 and stayed at $100\%$ for Scenario S3. All these values were found for $N_p = 500$. As seen previously, robustness improves when the particle number is increased.

## VII. Conclusions and Outlook

In this work we presented ActionSLAM, a simultaneous localization and mapping algorithm that makes use of location-related actions as landmarks in its internal representation of the environment. We showed experimentally and by simulating typical action recognition systems that the algorithm is robust towards errors in the recognition of such actions and capable of mapping the environment with mapping accuracy of $0.5m$ and path accuracy of $1.2m$. We confirmed these findings in simulated walking scenarios. By comparing the mapping capabilities of different sensor setups we found that simple setups result in faster convergence of ActionSLAM compared to sensor-rich action recognition systems while offering similar mapping accuracy. This may however come at the cost of lower robustness. Preliminary results suggest that a *Basic IMU* setup consisting of two motion sensors attached to the foot and the wrist and a smartphone in the trouser pocket may achieve sufficient robustness while being unobtrusive to the user.

Typical application scenarios of ActionSLAM will be in ambient-assisted living, home rehabilitation and person monitoring both at home and in office or manufacturing environments. ActionSLAM may be applied in any scenario where users repeatedly perform location-related actions without walking long distances in between. Due to the low computational requirements (only as few as $1000$ particles are required for robust mapping, compared to $\geq 10000$ particles in FastSLAM [24]), ActionSLAM is also well suited for real-time implementation on a smartphone. To achieve higher robustness, the algorithm could be extended in various ways, for example by fusion with other modalities such as Wi-Fi or deployed switches in the environment.

In future we will deploy and test ActionSLAM in real world, with state-of-the-art action recognition integrated. We will furthermore apply methods for tracking with unknown starting position in previously built maps, so that maps can be reused whenever a building is revisited. The particle filter approach is well-suited for this task, as it can generate and track particles with different starting positions at initialization. GPS may be used for anchoring of multiple indoor maps to global coordinates. Specifically, we will use ActionSLAM to support assistance of persons with Parkinson's disease at their home. In this context, analyzing gait and limb movements is required to assess efficacy of physical training and drugs. This is assessed by wearable sensors deployed on the feet, trunk and limb extremities. ActionSLAM will be used to provide user location "for free" from this existing sensor configuration.

## Acknowledgment

The research leading to these results has received funding from the European Union - Seventh Framework Programme (FP7/2007-2013) under grant agreement n°288516 (CuPiD project)

## References

[1] M. Skubic, G. Alexander, M. Popescu, M. Rantz, and J. Keller, "A smart home application to eldercare: Current status and lessons learned," *Technology and Health care*, vol. 17, no. 3, pp. 183 –201, 2009.

[2] A. Cesta, G. Cortellessa, R. Rasconi, F. Pecora, M. Scopelliti, and L. Tiberio, "Monitoring elderly people with the robocare domestic environment: Interaction synthesis and user evaluation," *Computational Intelligence*, vol. 27, no. 1, pp. 60–82, 2011.

[3] A. Roy, S. Das Bhaumik, A. Bhattacharya, K. Basu, D. Cook, and S. Das, "Location aware resource management in smart homes," in *Pervasive Computing and Communications, 2003. Proceedings of the First IEEE International Conference on*, 2003.

[4] D. Cook and S. Das, "How smart are our environments? an updated look at the state of the art," *Pervasive and Mobile Computing*, vol. 3, no. 2, pp. 53–73, 2007.

[5] F. Sposaro and G. Tyson, "ifall: An android application for fall monitoring and response," in *Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE*. IEEE, 2009, pp. 6119–6122.

[6] H. Durrant-Whyte and T. Bailey, "Simultaneous Localization and Mapping: Part I," *Robotics Automation Magazine, IEEE*, 2006.

[7] S. J. Preece, J. Y. Goulermas, L. P. J. Kenney, D. Howard, K. Meijer, and R. Crompton, "Activity identification using body-mounted sensors - a review of classification techniques," *Physiological Measurement*, vol. 30, pp. 1–33, 2009.

[8] H. Junker, O. Amft, P. Lukowicz, and G. Tröster, "Gesture spotting with body-worn inertial sensors to detect user activities," *Pattern Recognition*, vol. 41, no. 6, pp. 2010 – 2024, 2008.

[9] H. Sagha, S. T. Digumarti, J. d. R. Millán, R. Chavarriaga Lozano, A. Calatroni, D. Roggen, and G. Tröster, "Benchmarking classification techniques using the Opportunity human activity dataset," in *IEEE International Conference on Systems, Man, and Cybernetics*, 2011.

[10] M. Mathie, B. Celler, N. Lovell, and A. Coster, "Classification of basic daily movements using a triaxial accelerometer," *Medical and Biological Engineering and Computing*, vol. 42, no. 5, pp. 679–687, 2004.

[11] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM: a factored solution to the simultaneous localization and mapping problem," in *Eighteenth national conference on Artificial intelligence*. Menlo Park, CA, USA: American Association for Artificial Intelligence, 2002, pp. 593–598.

[12] E. Woyke, "Microsoft, Motorola, Nokia and RIM to battle Google over indoor location market," *Forbes Magazine*, 2011.

[13] E. Foxlin, "Pedestrian tracking with shoe-mounted inertial sensors," *Computer Graphics and Applications, IEEE*, 2005.

[14] A. Jiménez, F. Seco, J. Prieto, and J. Guevara, "Indoor pedestrian navigation using an INS/EKF framework for yaw drift reduction and a foot-mounted IMU," in *Positioning Navigation and Communication (WPNC), 2010 7th Workshop on*, 2010.

[15] S. Ayub, X. Zhou, S. Honary, A. Bahraminasab, and B. Honary, *Sensor Placement Modes for Smartphone-based Pedestrian Dead Reckoning*. Springer, 2010, pp. 123–132.

[16] M. Dissanayake, P. Newman, S. Clark, H. Durrant-Whyte, and M. Csorba, "A solution to the simultaneous localization and map building (SLAM) problem," *Robotics and Automation, IEEE Transactions on*, vol. 17, no. 3, pp. 229–241, 2001.

[17] S. Thrun, M. Montemerlo, D. Koller, B. Wegbreit, J. Nieto, and E. Nebot, "FastSLAM: An efficient solution to the simultaneous localization and mapping problem with unknown data association," *Journal of Machine Learning Research*, 2004.

[18] M. Montemerlo and S. Thrun, "Simultaneous localization and mapping with unknown data association using FastSLAM," in *Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on*, 2003.

[19] V. Pradeep, G. Medioni, and J. Weiland, "Robot vision for the visually impaired," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, 2010.

[20] B. Ferris, D. Fox, and N. Lawrence, "WiFi-SLAM using Gaussian process latent variable models," in *IJCAI 2007, Proceedings of*, 2007, pp. 2480–2485.

[21] H. Shin and H. Cha, "Wi-fi fingerprint-based topological map building for indoor user tracking," in *Embedded and Real-Time Computing Systems and Applications (RTCSA), 2010 IEEE 16th International Conference on*, 2010.

[22] H. Shin, Y. Chon, and H. Cha, "Unsupervised construction of an indoor floor plan using a smartphone," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 2011.

[23] P. Robertson, M. Angermann, and B. Krach, "Simultaneous localization and mapping for pedestrians using only foot-mounted inertial sensors," in *Proceedings of the 11th international conference on Ubiquitous computing*, ser. Ubicomp '09, 2009, pp. 93–96.

[24] M. Angermann and P. Robertson, "FootSLAM: Pedestrian Simultaneous Localization and Mapping Without Exteroceptive Sensors – Hitchhiking on Human Perception and Cognition," *Proceedings of the IEEE*, vol. 100, no. 13, pp. 1840 –1848, 13 2012.

[25] L. Bruno and P. Robertson, "WiSLAM: improving FootSLAM with WiFi," in *Indoor Positioning and Indoor Navigation (IPIN), 2011 International Conference on*, 2011.

[26] N. Castaneda and S. Lamy-Perbal, "An improved shoe-mounted inertial navigation system," in *Indoor Positioning and Indoor Navigation (IPIN), 2010 International Conference on*, 2010.

[27] J. D. Hol, T. B. Schon, and F. Gustafsson, "On resampling algorithms for particle filters," in *Nonlinear Statistical Signal Processing Workshop, 2006 IEEE*, 2006.

[28] F. Zampella, A. Jimenez, F. Seco, J. Prieto, and J. Guevara, "Simulation of foot-mounted IMU signals for the evaluation of PDR algorithms," in *Indoor Positioning and Indoor Navigation (IPIN), 2011 International Conference on*, 2011.

[29] O. J. Woodman, "An introduction to inertial navigation," Computer Laboratory, University of Cambridge, Tech. Rep. 696, 2007.