3D ActionSLAM: Wearable person tracking in multi-floor environments

Originally published in:
Personal and Ubiquitous Computing 19(1), https://doi.org/10.1007/s00779-014-0815-y

Pers Ubiquit Comput (2015) 19:123–141
DOI 10.1007/s00779-014-0815-y

ORIGINAL ARTICLE

3D ActionSLAM: wearable person tracking in multi-floor environments

Michael Hardegger · Daniel Roggen · Gerhard Tröster

Received: 7 February 2014 / Accepted: 22 April 2014 / Published online: 19 September 2014

Abstract We present 3D ActionSLAM, a stand-alone wearable system that can track people in previously unknown multi-floor environments with sub-room accuracy. ActionSLAM stands for action-based simultaneous localization and mapping: It fuses dead reckoning data from a foot-mounted inertial measurement unit with the recognition of location-related actions to build and update a local landmark map. Simultaneously, this map compensates for position drift errors that accumulate in open-loop tracking by means of a particle filter. To evaluate the system performance, we analyzed 23 tracks with a total walked distance of 6,489 m in buildings with up to three floors. The algorithm robustly (93 % of runs converged) mapped the areas with a mean landmark positioning error of 0.59 m. As ActionSLAM is fully stand-alone and not dependent on external infrastructure, it is well suited for patient tracking in remote health care applications. The algorithm is computationally light-weight and runs in real-time on a Samsung Galaxy S4, enabling immediate location-aware feedback. Finally, we propose visualization techniques to facilitate the interpretation of tracking data acquired with 3D ActionSLAM.

Keywords Simultaneous localization and mapping · Indoor tracking · Pedestrian dead reckoning · Location-aware computing · Wearable systems

M. Hardegger (✉) · G. Tröster
Wearable Computing Laboratory, ETH Zürich, Zurich, Switzerland
e-mail: michael.hardegger@ife.ee.ethz.ch

D. Roggen
University of Sussex, Brighton, UK

1 Introduction

The success of accurate outdoor localization through the Global Positioning System (GPS) suggests that location is an important source of information for and about people. Example applications include navigation, sports monitoring and location-based services. However, adults spend about 90 % of their time outside of the reach of GPS, i.e., inside buildings [29]. A survey in Germany [7] reports that for most of this time, people are at their private home (in average 15.7 ± 4.4 h per day). Indoors, walls block the GPS signals and alternative technologies such as GSM-based positioning do not provide sub-room accuracy as required by many applications.

In contrast to outdoor positioning, there are many competing, application- and environment-specific technologies available for indoor localization [25]. This includes Wifi fingerprinting (e.g., Google Indoor Maps[1] or NAO Cloud[2]) for navigation in large public buildings, and multilateration-based high-accuracy tracking (e.g., Cricket [41]). The existing systems differ in complexity, accuracy, installation and maintenance effort, and robustness.

In this work, we aim at facilitating localization in *home environments* and thereby extend person tracking to the space in which people spend most of their time. At-home position traces contain information about a person's daily routines and his or her current state with respect to these routines. This is for example of use in smart homes, which optimize lighting and heating schedules based on the location behavior of the inhabitants [47]. In remote health care, at-home tracking systems indicate long-term changes in a user's activity patterns (see e.g., [10]), or they may

[1] http://maps.google.com/help/maps/indoormaps/.
[2] https://www.nao-cloud.com/.

assist patients in hazardous areas such as zones of frequent falling [49]. A related example is location-induced rhythmic auditory stimulation (RAS) for Parkinson's disease patients [33]. RAS was shown to help patients when they move through areas in which they often experience walking difficulties, e.g., narrow corridors and doorways. When used by multiple inhabitants of the same home, a location tracker does not just reveal information about each individual's daily life routines, but also about their personal interactions and the social dynamics in the house [12].

Localization technologies for deployment in private settings should fulfill environment-specific requirements: Usually, people with little technical expertise install and maintain these systems, and only few people make use of a single deployed infrastructure. As a result, the one-off cost as well as the maintenance effort should be much smaller compared to indoor positioning systems for public environments, where building managers are in charge and up to thousands of people may benefit from a single installation. Furthermore, accuracy constraints in monitoring and related applications are higher (within-room accuracy) compared to navigation and friend-finder tools as for example supported by Google Indoor Maps [13].

ActionSLAM is a fully stand-alone, wearable localization system that we first introduced in [23]. The system fulfills the low deployment effort and high-accuracy requirements for at-home tracking and thereby fills an important gap toward ubiquitous all-day tracking. In this paper, we expand upon our previous work by presenting a 3D version of the algorithm, introducing an improved motion model, and providing systematic evaluations of the system's performance and boundary operating conditions. 3D ActionSLAM infers position from a single foot-mounted IMU and a hip-worn smartphone that acts as both, sensor and computing platform. This setup recognizes stair climbing, standing and sitting as observations of landmarks that form a simultaneously built multi-floor map. 3D ActionSLAM does not require any installation and maintenance effort, and can therefore easily be deployed to users without technical expertise. Nevertheless, it achieves within-room positioning accuracy. Our android implementation of the algorithm [22] can furthermore track a person in real-time on a Samsung Galaxy S4, enabling location-aware audio and visual feedback through a fully wearable system. The here presented version of 3D ActionSLAM supersedes all previous iterations of the algorithm described in [22, 23] and [24].

In addition, we present visualization techniques for 3D ActionSLAM tracks that facilitate the interpretation of a person's location behavior over time. In particular, we show how the paths of multiple users can be translated into narrative charts that describe the interpersonal location behavior, giving an insight into the social dynamics between home inhabitants.

Overall, our work supports the claim that 3D ActionSLAM is a robust and easy-to-use system for both, real-time location-awareness, and to post-analyze the walks of people in single- and multi-floor homes.

2 State of the art

2.1 Person tracking technologies

Person tracking is the problem of estimating the position $\hat{\boldsymbol{x}}$ of a person whose actual position is $\boldsymbol{x}$, from data of one or multiple sensor modalities. A key characteristic of tracking systems is the absolute accuracy, defined as the Euclidean distance $d_x(t) = \|\boldsymbol{x}(t) - \hat{\boldsymbol{x}}(t)\|_2$. The mean tracking accuracy $\bar{d}_x$ is the average distance $d_x(t)$ over the full track of the person. Further characteristics of a tracking system are the deployment effort, robustness, the obtrusiveness of the setup, privacy intrusion level, power requirements, and whether the system makes use of dedicated and/or anyway available infrastructure in the environment (see the surveys in [25] and [31]). Subsequently, we discuss the most relevant technologies with respect to the problem of tracking people in home environments.

Multilateration systems locate a person from the arrival times of signals from multiple synchronized sources (e.g., GPS satellites). For indoor applications, the so-called pseudolites (e.g., LOCATA[3]) deployed in the environment can emit signals that penetrate walls. Alternative implementations use a network of ultrasound or Ultra-WideBand (UWB) emitters (e.g., UBISENSE[4]) to ensure line-of-sight contact to the receiver at any time. If properly calibrated, indoor multilateration systems are very accurate ($\bar{d}_x < 0.1$ m), but they are sensitive to changes in the environment, and expensive to deploy and maintain [31]. On the other hand, *fingerprinting* approaches make use of already available signal sources in the environment such as Wifi, sound, and ambient light and position the user by comparing current measurements to a pre-recorded fingerprint map of the surroundings. Typically, fingerprinting is inaccurate ($\bar{d}_x > 3$ m), and it requires the recording and regular updating of prior fingerprint maps [14].

Two further approach families are independent of infrastructure and a-priori knowledge about the environment: Pedestrian Dead Reckoning (PDR) and Simultaneous Localization And Mapping (SLAM) (which typically complements PDR). Both approaches exploit body-worn sensors only to localize a person and are therefore fully

[3] http://www.locata.com.

[4] http://ubisense.net.

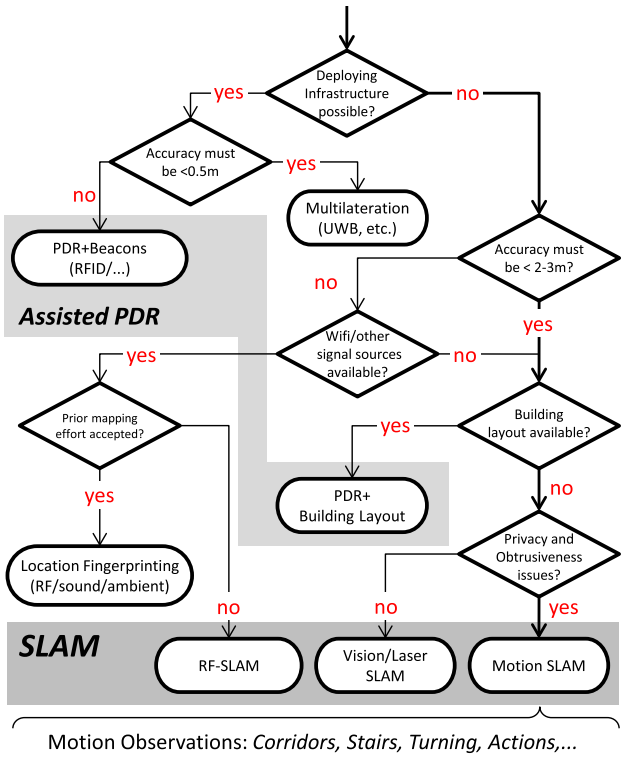

Fig. 1 This (simplified) decision flow chart depicts the choice of technology for *long-term* indoor tracking of humans. Unassisted PDR works only for short tracks and is therefore omitted in this chart

wearable. While PDR only uses body-mounted motion sensors, SLAM typically fuses multiple modalities such as motion sensors, cameras, lasers, and Radio-Frequency (RF) signals. SLAM implementations that fuse motion measurements with cameras or lasers achieve high accuracy (e.g., $\bar{d}_x < 0.2$ m in [8]). However, vision-based modalities are less desirable in mobile and ubiquitous computing as they may be considered as privacy-invading (with the camera capturing others or the user itself), have higher power consumption (e.g., active laser source), and are obtrusive (need of occlusion-free on-body placement) [13]. Motion sensors on the other side can be placed below clothing or inside shoes [37], they are small and lightweight, and they record motion information at a comparatively low data bandwidth, allowing for real-time processing on lower-power embedded systems.

The decision flow chart in Fig. 1 depicts how the problem of tracking people at their home relates to other indoor tracking scenarios. Given that we should avoid prior deployment effort in this scenario, require sub-room accuracy, and prefer an unobtrusive system, PDR and motion-sensor-based SLAM are the most promising approaches. We discuss both technologies in the remainder of this section.

2.2 Pedestrian dead reckoning

PDR systems estimate the *pose* $\hat{s}_t$ of a user at time t through integration of the measurements of one or multiple body-worn motion sensors. The pose s_t is made up of the person's floor coordinates and altitude $\boldsymbol{x}_t = \{x_t, y_t, h_t\}$, and his heading ϕ_t. Steps u_t link two subsequent poses s_{t-1} and s_t, i.e., $s_t = f(s_{t-1}, u_t)$. PDR systems therefore perform *open-loop* integration, and sensor noise accumulates in the pose estimate $\hat{s}_t$. As a result, d_x increases with time and PDR is only suited for short-term tracking. Accuracy in PDR is therefore reported as relative to the travelled distance. There are two common PDR approaches:

- *Step-and-Heading (SH-) PDR* systems either rely on hip- or belt-mounted IMUs [50], or they do not make any assumption regarding the on-body location of the sensor [5, 43]. SH-PDR combines step detection rules, step length heuristics and heading estimators to derive the user's trajectory $\hat{s}^t = \{\hat{s}_0, \ldots, \hat{s}_t\}$. For regular, straight gait, the error accumulation for SH-PDR may be less than 2 % of the distance travelled [27]. In recordings with more complex foot motions such as jumps and walking in place, SH-PDR, however, fails.
- *Zero-velocity UPdaTe (ZUPT-) PDR* on the other side integrates the raw acceleration and rotation velocity data of a *foot-mounted IMU*, and corrects the velocity whenever the foot is on the ground and therefore not moving to $v \approx 0 \frac{m}{s^2}$. Outside and in wooden buildings, where magnetometers provide in addition reliable information about the sensor's heading ϕ_t, the position drift can be as low as 0.3 % of the distance travelled [20]. In particular, in steel-framed constructions, the magnetic field close to the ground is strongly disturbed and can only be used as local reference [46]. ZUPT-PDR without magnetometer correction achieves a tracking accuracy of 0.6–1.2 % of the distance travelled [27]. Fusion of IMUs attached to both feet [40] and zero angular rate updates (ZARUs) [27] further decrease the error accumulation. To perform frequent zero-velocity updates, ZUPT-PDR requires that the feet are on the ground and not moving at regular intervals. Aside from that, ZUPT-PDR is independent of the user's motions and therefore better suited for real-world deployment than SH-PDR.

The accumulation of position errors in PDR may be alleviated by restoring the position when the person passes a pre-placed RFID beacon [28], or by constraining the path estimate to correspond with the building layout [51]. However, these *assisted PDR* solutions require the deployment of infrastructure and/or the availability of prior maps.

2.3 Simultaneous localization and mapping

If no a-priori information is available, additional observations from the same and/or complementary body-worn sensors must be fused to compensate for PDR drift errors. SLAM is a family of techniques in which one set of sensors estimates the *user trajectory* s^t (e.g., through PDR), while another set of sensors recognizes *landmarks* $\theta_{[i]}$ in the environment. By adding these landmarks to a *map* Θ, future observations of the same landmarks will compensate for errors in the user trajectory. SLAM was originally introduced in robotics for navigation of autonomous, camera- and laser-equipped agents in unknown environments [8]. However, some specific landmark categories can also be recognized from a person's motions, and therefore from the same body-worn IMUs as used to estimate the trajectory. The following motion landmark categories have been investigated: wall-constrained walking patterns (turns in corners [38], corridors [2]) and location-related actions such as door opening and sitting [21, 23]. Unlike in PDR, the position error d_x in SLAM does not increase with walking distance, and it is therefore usually reported as average $\bar{d}_x$ over time.

Mathematically, SLAM is the problem of estimating the pose s_t within a previously unknown map Θ, given step measurements $\hat{u}^t = \{\hat{u}_0, \ldots, \hat{u}_t\}$, environment observation $\hat{z}^t = \{\hat{z}_0, \ldots, \hat{z}_t\}$ and data associations $\hat{n}^t = \{\hat{n}_0, \ldots, \hat{n}_t\}$ [11]. Each data association $\hat{n}_t$ associates an observation $\hat{z}_t$ to a landmark $\theta_{[i]=\hat{n}_t}$ in the map Θ. In a probabilistic sense, SLAM approximates the probability distribution $p(s_t, \Theta|\hat{u}^t, \hat{z}^t, \hat{n}^t)$ and derives maximum likelihood estimates for the pose s_t and the map Θ. Most SLAM implementations calculate $p(s_t, \Theta|\hat{u}^t, \hat{z}^t, \hat{n}^t)$ in an iterative process: In the *motion update*, they estimate s_t given the prior pose s_{t-1} and the latest motion reading according to $p(s_t|s_{t-1}, \hat{u}_t)$. Then, the *observation update* fuses the environment observations with the trajectory to update the map Θ. This two-step process can be implemented by means of an Extended Kalman Filter (EKF) [8]. In some cases, however, nonlinear filtering methods such as particle filters perform better [35].

Table 1 lists SLAM implementations that were applied in pedestrian tracking. The systems differ in the algorithm for estimating the user trajectory, the type of environment observation, and the applied SLAM algorithm. SLAM represents maps either as a set of landmarks $\theta_{[i]}$ or by means of an occupancy grid (see e.g., [2]). All the systems listed in Table 1 except for [39] aim at voluntary mapping of larger buildings, for example to acquire radio fingerprint maps with low effort. Users in these applications walk with the goal of mapping, and they behave accordingly. To the best of our knowledge, the only work that aims at performing pedestrian SLAM from natural human behavior is ActionSLAM, as first proposed in [23] and extended in the next section. ActionSLAM applies an alternative type of landmark observations to SLAM, i.e., the recognition of location-specific actions. These actions are frequent in a person's daily life and thus enable robust and accurate SLAM execution. A further limitation to the studies in Table 1 is that they only present one or few test walks and lack a detailed evaluation. We here demonstrate the capabilities of 3D ActionSLAM in 23 recordings, 17 of them performed under real-life conditions.

3 The 3D ActionSLAM algorithm

3.1 Overview and definitions

In this section, we present an updated version of the ActionSLAM algorithm from [23], with a novel motion model and the modifications added to allow for 3D mapping. Figure 2 depicts the overall framework of 3D ActionSLAM, with its two main blocks: The *pre-processing phase*, and the *SLAM update phase*. In pre-processing, 3D ActionSLAM derives a step estimate $\hat{u}$ and recognizes location-related actions $\hat{A}$ from body-worn inertial sensors. A Rao-Blackwellized particle filter then fuses these measurements as proposed in [35].

3D ActionSLAM segments paths into stance phases t with poses $s_t = \{x_t, y_t, h_t, \phi_t\}$, and steps u_t that connect s_{t-1} and s_t. In this notation, $\{x_t, y_t, h_t\}$ denotes the 3D position of the user at time t, and ϕ_t the foot's heading. The outputs of 3D ActionSLAM are a path $\bar{s}^t = \{\bar{s}_0, \ldots, \bar{s}_t\}$ composed of poses $\bar{s}_t = \{\bar{x}_t, \bar{y}_t, \bar{h}_t, \bar{\phi}_t\}$, and a map $\bar{\Theta}_t$ made up of $N_{l,t}$ landmarks $\bar{\theta}_{t,[i]}$. $[i] \in \{1 \ldots N_{l,t}\}$ is the index of the landmark, and $N_{l,t}$ the number of landmarks in the map.

3.2 Pre-processing

3.2.1 User trajectory

The first output of pre-processing is the open-loop estimate $\hat{s}^t$ of the person's trajectory, made up of steps $\hat{u}_t$. We use ZUPT-PDR for estimation of 3D foot coordinates. Our implementation follows the guidelines in [18] and is similar to the open-source code of the OpenShoe project.[5] The stance detection of ZUPT-PDR segments the walking path into steps $\hat{u}_t$ described by horizontal step length $\hat{l}_t$, altitude change $\delta\hat{h}_t$ and heading change $\delta\hat{\phi}_t = \hat{\phi}_t - \hat{\phi}_{t-1}$ (see Fig. 4). Preliminary recordings showed that a major error contribution in real-life scenarios is heading drift during long phases without movement, mainly as a consequence of gyroscope random walk noise. Figure 3 depicts the

[5] www.openshoe.org.

Table 1 Body-worn SLAM systems proposed for human tracking

System	Trajectory	Observation	Algorithmic framework	Evaluation and applications
Pradeep et al. [39]	Stereo vision	Visual object recognition	FastSLAM (particle filter) with object landmarks	High-accuracy tracking, large computational, and storage requirements
Ferris et al. [17]	Probabilistic motion model (no sensor)	Wifi fingerprint landmarks	Gaussian process latent variable model	Mean error of 4 m in simple walking scenarios
Shin et al.: SmartSLAM [48]	SH-PDR from smartphone in pocket	Wifi fingerprint landmarks	FastSLAM with Wifi landmarks	Mean error of 3 m in simple walking scenarios
Faragher et al.: Opportunistic Radio SLAM [15]	SH-PDR from hip-mounted IMU	Radio (Wifi, GSM, etc.) fingerprints	DPSLAM (particle filter)	Position error of 4 m after 15 min of walking in office buildings
Mirowski et al.: SignalSLAM [34]	SH-PDR from smartphone	Signal (Wifi, BT, LTE, and magnetic field) fingerprints	GraphSLAM (posterior optimization) with signal landmarks	Median difference between tracks in same office building less than 5 m
Robertson et al.: MagSLAM [45]	ZUPT-PDR	Magnetic Field Strengths in combination with FootSLAM	FastSLAM with hexagonal occupancy grid	Mean error of $\approx$ 15 cm after 17 min mapping walk with 12 repetitions in residential flat, similar in other scenarios
Park et al. [38]	SH-PDR from smartphone in pocket	Corner landmarks (recognized from path)	FastSLAM	Correct maps of simple floor plans
Angermann et al.: FootSLAM [2]	ZUPT-PDR	Floors (recognized from path)	FastSLAM with hexagonal occupancy grid	Mean error of $2m$ for mapping of office buildings
Robertson et al.: PlaceSLAM [44]	ZUPT-PDR	User labels at revisited places in combination with FootSLAM	FastSLAM with hexagonal occupancy grid	Similar performance as FootSLAM, but at reduced computational costs
Grzonka et al. [21]	Xsens motion suit post-processing	Stairs and doors as recognized from motion suit data	Scan-matching with multi-hypothesis tracking	Voluntary mapping of large buildings

introduced, slow drift in heading for standard ZUPT-PDR while a person is sitting. As described in [24], we correct for this drift by constraining the heading changes that ZUPT-PDR predicts to be similar to those measured with a magnetometer. This uses the fact that the earth magnetic field is locally constant with time, even in steel-framed buildings [45].

3.2.2 Action recognition

Previous work [23, 24] indicated that for ActionSLAM execution in at-home tracking, the identification of the two basic actions *sitting* and *standing still* is sufficient. These actions are often related to a location: *Sitting* can only take place on chairs and sofas, and *standing still* most frequently occurs in front of sinks, drawers, windows, etc. Both actions can be recognized from motion sensors attached to the hip and one foot.

For 3D tracking in multi-floor buildings, two further location-related actions may be recognized with the same sensor setup: the reaching of the lower end of a stair (*stair low*), and the reaching of the upper end of a stair (*stair high*). To detect these actions, the recognition algorithm calculates the variance $\mathrm{var}(h(t))$ of the ZUPT-PDR altitude output $h(t)$ in a sliding window of length ΔT. If $\mathrm{var}(h(t))$ stays for at least τ_0 above a threshold h_0, the phase is identified as a stair ascent or descent.[6] Thanks to this requirement, jumps and similar short vertical motions are not misinterpreted as stairs. The beginning and end of a stair ascent or descent phase are then observations of *stair high* and *stair low*.

As final output, the action recognition block of 3D ActionSLAM (see Fig. 2) provides observations $\hat{A}_t \subset \{\text{sitting, standing still, stair high, stair low}\}$ associated to stance phases t.

3.3 3D ActionSLAM update

3.3.1 Particle filter

In contrast to the standard SLAM problem of estimating $p(s_t, \Theta | \hat{u}^t, \hat{z}^t, \hat{n}^t)$ (see Sect. 2.3), the landmarks of 3D

[6] We used a sliding window length $\Delta T = 2$ s, given that this time span should always cover a full step, and experimentally chose $\tau_0 = 4$ s and $h_0 = 0.05$ m.

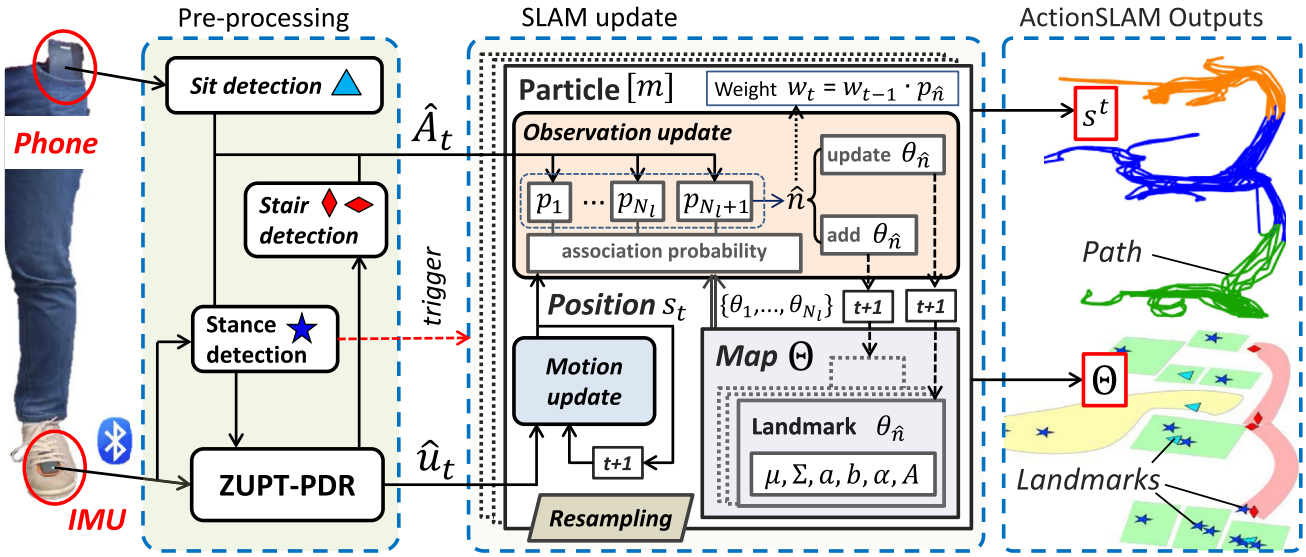

Fig. 2 The 3D ActionSLAM system takes measurements from body-worn motion sensors and processes them to a map Θ made up of landmarks $\theta_{[i]}$, and a path s^t within this map (according to [24])

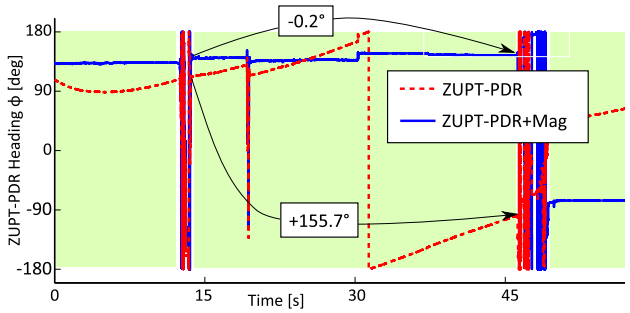

Fig. 3 This plot shows the heading estimation ϕ_t of ZUPT-PDR with and without magnetometer correction for a recording with long no-movement phases (marked), interrupted by two short walks. For ZUPT-PDR without magnetic field correction, we estimate a slow heading drift at times when the user did actually not move his foot

ActionSLAM are not uniquely identifiable with $\hat{n}_t$, but only their action type $\hat{A}_t$. Furthermore, the estimated position of a landmark observed at time t is always equal to the person's position at the same time. Therefore, $\hat{z}_t$ can be derived from s_t and Θ alone, which reduces the SLAM problem to approximating $p(s_t, \Theta_t | \hat{u}^t, \hat{A}^t)$. For fusion of motion and observation measurements, 3D ActionSLAM uses the Rao-Blackwell factorization proposed in [35]:

$$p(s^t, \Theta_t | \hat{u}^t, \hat{A}^t) = p(s^t | \hat{u}^t) \prod_{[i]=1}^{N_{l,t}} p(\theta_{[i],t} | s^t, \hat{A}^t) \quad (1)$$

This factorization decomposes the SLAM problem of estimating a path s^t in a previously unknown environment Θ_t into separate estimators for the person's path s^t and each of the $N_{l,t}$ landmarks $\theta_{[i],t}$. 3D ActionSLAM estimates the path probability $p(s^t | \hat{u}^t)$ in a particle filter with N_p particles, thus being capable of approximating non-Gaussian distributions and performing nonlinear filtering. Meanwhile, $N_{l,t}$ individual filters estimate the landmark probability distributions $p(\theta_{[i],t} | s^t, \hat{A}^t)$. As the landmark characteristics $\theta_{[i],t}$ are conditioned on the person's path, each particle $[m]$ must maintain its own map $\Theta^{[m]}$, together with the pose $s_t^{[m]}$.

3D ActionSLAM estimates $s_t^{[m]}$ in the *motion update*, while the updating of the map $\Theta^{[m]}$ is done in the *observation update* and only in response to action observations. Algorithm 1 outlines the main functions of the 3D ActionSLAM updating.

Algorithm 1 3D ActionSLAM Update

Input: $\hat{A}^t, \hat{u}^t, T^t, N_p, N_{th}$
Output: $\bar{s}^t, \bar{\Theta}_t$

```
{s_0^[m], Θ_0^[m], w^[m]} ⇐ initializeParticles(N_p)
for τ = 1, ..., t do
    for [m] = 1, ..., N_p do
        s_τ^[m] ⇐ motionUpdate(s_{τ-1}^[m], û_τ, T_τ)
        Θ_τ^[m], w^[m] ⇐ obsUpdate(Θ_{τ-1}^[m], s_τ^[m], Â_τ, w^[m])
        Θ_τ^[m] ⇐ mapMaintenance(Θ_τ^[m])
    end for
    if N_eff = 1 / Σ_{m=1}^{N_p} (w_t^[m]) < N_th then
        {s_τ^[m], Θ_τ^[m], w^[m]} ⇐ resample({s_τ^[m], Θ_τ^[m], w^[m]})
    end if
end for
[m̄] ⇐ findBestParticle({Θ_t^[m], w^[m]})
s̄^t ⇐ s^{t,[m̄]}
Θ̄_t ⇐ Θ_t^[m̄]
```

3.3.2 Motion update

At each beginning of a stance phase t, 3D ActionSLAM performs a motion update and sequentially calculates $p(s^t|\hat{u}^t)$ through sampling of particle poses from:

$$s_t^{[m]} \sim p\left(s_t^{[m]}|s_{t-1}^{[m]}, \hat{u}_t\right) \tag{2}$$

The *motion model* describes the probability density function $p(s_t^{[m]}|s_{t-1}^{[m]}, \hat{u}_t)$. In previous work, we used a Gaussian error model for describing errors in step length and heading estimation [23]. This model did not account for the exponential error growth in long phases without stance detections. We now define T_t as the time that passes between the two subsequent stance phases $t-1$ and t. According to [52], the position error introduced by a perfectly calibrated accelerometer in this phase follows a second-order random walk with standard deviation $\sigma_{x,acc}(T_t) = k_{x,acc} \cdot T_t^{\frac{3}{2}}$, where $k_{x,acc}$ is a characteristic of the accelerometer. The heading error introduced by an unbiased, calibrated gyroscope follows a random walk with $\sigma_{\phi,gyro}(T_t) = k_{\phi,gyro} \cdot \sqrt{T_t}$, with $k_{\phi,gyro}$ being a gyroscope property. Assuming constant forward velocity during the foot swing, the gyroscope adds furthermore a random walk position error with $\sigma_{x,gyro}(T_t) \approx k_{x,gyro} \cdot T_t^{\frac{3}{2}}$. In [52], the author shows with simulations and measurements that gyroscope errors are predominant for $T_t > 0.3$ s, therefore we omitted $\sigma_{x,acc}(T_t)$ in the 3D ActionSLAM motion model.

In addition to sensor errors, inaccuracies in stance detection affect the ZUPT-PDR position and heading estimates. We modeled these errors as additive Gaussian noise with standard deviations $\sigma_{x,0}$ for position and $\sigma_{\phi,0}$ for heading, both independent of T_t. The new pose $s_t^{[m]}$ of a particle after a step u_t equals the sum of the previous pose $s_{t-1}^{[m]}$, the ZUPT-PDR estimate $\hat{u}_t$, and a sampled error $\rho^{[m]}$. Figure 4 depicts the variable definitions in two dimensions. The resulting motion equations are:

$$\rho_\phi^{[m]} \sim \mathcal{N}\left(0, \sigma_{\phi,0} + k_{\phi,gyro} \cdot \sqrt{T_t}\right) \tag{3}$$

$$\rho_x^{[m]} \sim \mathcal{N}\left(0, \sigma_{x,0} + k_{x,gyro} \cdot T_t^{\frac{3}{2}}\right) \tag{4}$$

$$\rho_y^{[m]} \sim \mathcal{N}\left(0, \sigma_{x,0} + k_{x,gyro} \cdot T_t^{\frac{3}{2}}\right) \tag{5}$$

$$\rho_h^{[m]} \sim \mathcal{N}\left(0, \sigma_{h,0} + k_{h,gyro} \cdot T_t^{\frac{3}{2}}\right) \tag{6}$$

$$\phi_t^{[m]} = \phi_{t-1}^{[m]} + \delta\hat{\phi}_t + \rho_\phi^{[m]} \tag{7}$$

$$x_t^{[m]} = x_{t-1}^{[m]} + \hat{l}\cos\phi_t^{[m]} + \rho_x^{[m]} \tag{8}$$

$$y_t^{[m]} = y_{t-1}^{[m]} + \hat{l}\sin\phi_t^{[m]} + \rho_y^{[m]} \tag{9}$$

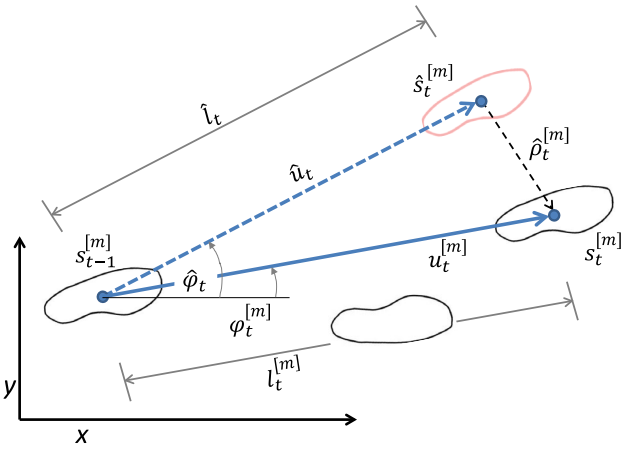

Fig. 4 Definitions of step measurements for particle $[m]$. $\hat{u}_t$ is the ZUPT-PDR estimate of the step u_t, $\hat{s}_t^{[m]}$ the particle's new position before adding the motion model noise $\hat{\rho}_t^{[m]}$, and $s_t^{[m]}$ the final position of particle $[m]$ after the motion update

$$h_t^{[m]} = h_{t-1}^{[m]} + \delta\hat{h} + \rho_h^{[m]} \tag{10}$$

3.3.3 Observation update

For non-empty $\hat{A}_t$, 3D ActionSLAM performs observation updates after the motion update associated to stance phase t. If more than one action occurs during a single stance phase (e.g., *stairs high* and *standing still*), the action recognition triggers multiple subsequent observation updates. The observation update itself did not change compared to the previous ActionSLAM version in [24], except that we now assign an additional coordinate h for vertical displacement to landmarks.

During the observation update, 3D ActionSLAM modifies the maps $\Theta_{t-1}^{[m]}$ of each particle given its current pose $s_t^{[m]}$ and the observation $\hat{A}_t$. First, the algorithm decides whether the observation corresponds to a landmark that is already in the map, and if yes, to which of these landmarks. It then either adds a new landmark $\theta_{N_{l,t},t}^{[m]}$ with $N_{l,t} = N_{l,t-1} + 1$, or modifies the associated $\theta_{[i],t}^{[m]}$. Figure 5 illustrates the decision procedure. Landmarks in 3D ActionSLAM have a planar elliptic shape, which accounts for the fact that different foot placements in a plane can correspond to the same location-related action. Consider for example sitting: The foot may move in an area of $\sim$0.5 m diameter without a change in upper-body posture. The parameters we use in 3D ActionSLAM to describe a landmark are therefore the centroid location $\{x_{[i],t}, y_{[i],t}\}$, the ellipse shape parameters $\{a_{[i],t}, b_{[i],t}, \alpha_{[i],t}\}$, and the altitude of the landmark $h_{[i],t}$. In addition, each landmark has an associated action type $A_{[i]} \in \{\text{sitting, standing still, stair high, stair low}\}$ that remains fix.

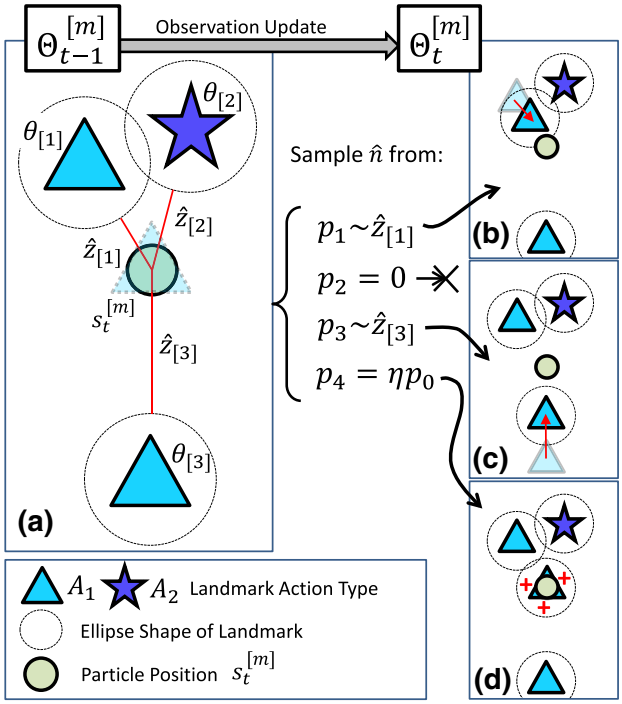

Fig. 5 At time t, an action of type A_1 is recognized. Part (*a*) depicts the state of particle $[m]$ before the observation update (map $\Theta_{t-1}^{[m]}$, pose $s_t^{[m]}$). In the data association step, 3D ActionSLAM calculates the probability of each association $\hat{n}$ based on the distances to the *ellipse-shaped* landmark in $\Theta_{t-1}^{[m]}$. As the action type of $\theta_{[2]}$ is not A_1, the association probability for $\hat{n} = 2$ is $p_2 = 0$. Part (*b*) shows the consequence of choosing $\hat{n} = 1$ (update position of $\theta_{[1]}$, indicated by *arrow*), (*c*) for $\hat{n} = 3$ (update $\theta_{[3]}$), and (*d*) for $\hat{n} = 4$ (insertion of a new landmark $\theta_{[4]}$ at the current particle position)

For better readability, we leave away the particle index $[m]$ from now on. First, 3D ActionSLAM calculates the probability for associating the current observation to each of the previously inserted landmarks in the particle's map. If $\hat{A}_t \neq A_{[i]}$, the probability of association $p_{[i],t}$ equals 0. In all other cases, the difference vector $\hat{z}_{[i]}$ between the current location of the foot and the landmark's shape is calculated, using the formulas for ellipse intersection (with $q_{[i]}$ as intersection angle):

$$q_{[i]} = \operatorname{atan}\left(\frac{a_{[i],t-1}}{b_{[i],t-1}} \cdot \frac{x_t - x_{[i],t}}{y_t - y_{[i],t}}\right) \tag{11}$$

$$\tilde{x}_e = a_{[i],t-1}\cos(q_{[i],t})\cos(\alpha_{[i],t-1}) \tag{12}$$

$$- b_{[i],t-1}\sin(q_{[i]})\sin(\alpha_{[i],t-1}) \tag{13}$$

$$\tilde{y}_e = a_{[i],t-1}\cos(q_{[i],t})\sin(\alpha_{[i],t-1}) \tag{14}$$

$$+ b_{[i],t-1}\sin(q_{[i]})\cos(\alpha_{[i],t-1}) \tag{15}$$

$$\hat{z}_{[i]} = \begin{pmatrix} \max(0, x_{[i],t-1} - \tilde{x}_e) \\ \max(0, y_{[i],t-1} - \tilde{y}_e) \\ h_{[i],t-1} - h_t \end{pmatrix} \tag{16}$$

The probability $p_{[i]}$ for associating the action observation $\hat{A}_t$ to the landmark $\theta_{[i],t-1}$ is then:

$$p_{[i]} = \begin{cases} 0, & \text{if } \hat{A}_t \neq A_{[i]}, [i] \leq N_{l,t-1} \\ \eta \cdot |2\pi Q_{[i]}|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}\hat{z}_{[i]}^T Q_{[i]}^{-1} \hat{z}_{[i]}\right), & \text{if } \hat{A}_t = A_{[i]}, [i] \leq N_{l,t-1} \\ \eta \cdot p_0, & \text{if } [i] = N_{l,t-1} + 1 \end{cases} \tag{17}$$

η is a normalization factor so that the sum of all $p_{[i]}$ with $[i] = 1 \ldots N_{l,t-1} + 1$ is 1. The observation covariance matrix $Q_{[i]}$ is the sum of the landmark position covariance $\Sigma_{[i],t-1}$, and the measurement covariance R_t.

$$Q_{[i]} = \Sigma_{[i],t-1} + R_t \text{ with } R_t = \begin{pmatrix} r_0^2 & 0 & 0 \\ 0 & r_0^2 & 0 \\ 0 & 0 & r_h \end{pmatrix} \tag{18}$$

Given the probabilities $p_{[i]}$, 3D ActionSLAM samples the data association decision $\hat{n} \in \{1, \ldots, N_{l,t-1} + 1\}$. If the outcome is $\hat{n} = N_{l,t-1} + 1$, 3D ActionSLAM adds a new landmark with the following characteristics to the map Θ_{t-1} of the particle:

$$x_{\hat{n},t} = x_t,\ y_{\hat{n},t} = y_t,\ h_{\hat{n},t} = h_t \tag{19}$$

$$a_{\hat{n},t} = b_{\hat{n},t} = r_1,\ \alpha_{\hat{n},t} = 0o \tag{20}$$

$$A_{\hat{n}} = \hat{A}_t \tag{21}$$

$$\Sigma_{\hat{n},t} = R_t \tag{22}$$

If, however, $\hat{n} \leq N_{l,t-1}$, 3D ActionSLAM updates the associated landmark's position in a Kalman filter with gain K:

$$K = \Sigma_{\hat{n},t-1} Q_{\hat{n}}^{-1} \tag{23}$$

The new position of the landmark $\theta_{\hat{n},t}$ and the updated position covariance $\Sigma_{\hat{n},t}$ are:

$$\begin{pmatrix} x_{\hat{n},t} \\ y_{\hat{n},t} \\ h_{\hat{n},t} \end{pmatrix} = \begin{pmatrix} x_{\hat{n},t-1} \\ y_{\hat{n},t-1} \\ h_{\hat{n},t-1} \end{pmatrix} - K\hat{z}_{\hat{n}}^T \tag{24}$$

$$\Sigma_{\hat{n},t} = (I - K)\Sigma_{\hat{n},t-1} \tag{25}$$

If the center of the landmark $\theta_{\hat{n},t}$ is within the ellipse shape of another landmark $\theta_{[j]}$ of identical action type after this step, the two ellipses are combined to a single ellipse landmark $\theta'_{\hat{n},t}$. 3D ActionSLAM fits an ellipse around all observation locations of the landmarks $\theta_{[j]}$ and $\theta_{\hat{n},t}$, with the semi-major axis lengths constrained to be ≤ 0.8 m.

3D ActionSLAM executes the observation update for each particle individually and in the end calculates the new weights $w_t^{[m]}$ according to Montemerlo et al. [35]:

$$w_t^{[m]} = w_{t-1}^{[m]} \cdot p_{\hat{n}}^{[m]} \tag{26}$$

3.4 Resampling and post-processing

After each observation update, 3D ActionSLAM calculates the effective particle number $N_{eff} = \frac{1}{\sum_{m=1}^{N_p}\left(w_t^{[m]}\right)}$ and performs systematic resampling as in [9] if $N_{eff} < N_{th}$. In this way, the filter gets rid of particles with very low weight, and better approximates $p\left(s^t, \Theta_t | \hat{A}^t, \hat{u}^t\right)$ in areas which are not close to zero.

Sometimes, the observation update step adds landmarks due to *not* location-related actions, for example because a person stops in the middle of a room to answer a phone call. Such landmarks should be removed, since they may lead to erroneous data association in later observation updates. For this purpose, 3D ActionSLAM applies the landmark removal method proposed in [24]: The system stores an observation time $L_{[i]}$ for each landmark and resets it to the current time whenever $\hat{n} = [i]$. $L_{[i]}$ is measured in distance walked within the indoor area. *Map maintenance* removes landmarks from the particle's map if they have not been observed for at least $L_{\max}$.[7]

The 3D ActionSLAM particle filter approximates the probability density $p\left(s^t, \Theta_t | \hat{A}^t, \hat{u}^t\right)$, but we are actually interested in s^t and Θ_t. The 3D ActionSLAM estimates $\bar{s}^t$ and $\bar{\Theta}_t$ of s^t and Θ_t are the paths and map of the *best particle* $[\bar{m}]$. This is the particle which best reflects the current belief of the filter according to the following, heuristically defined rules: 1) Candidates for $[\bar{m}]$ are all particles with the minimum number of landmarks $N_l^{[\bar{m}]} \leq N_l^{[m]} \; \forall \; [m] = 1 \ldots N_p$, and 2), among these candidates, $[\bar{m}]$ is the particle with the highest weight $w_t^{[\bar{m}]}$.

4 System evaluation

4.1 Datasets

We characterize 3D ActionSLAM on a rich set of at-home and office recordings with multiple sensor setups and settings, reflecting the wide range of usage of the system. Figure 6 shows the sensor setups we employed in this work, and Table 2 summarizes the most relevant characteristics and settings of the corresponding ETHOS [26], EXLs1[8] and EXLs3[9] IMUs. Table 3 gives an overview of all recordings that we investigate. In all experiments, we attached one IMU to the right foot, and a second IMU or a smartphone to the right upper leg, both measuring synchronized acceleration, rotation velocity and magnetic field data. We used the upper leg sensor to distinguish between sitting and standing, while the foot-mounted IMU provided the input for ZUPT-PDR. The IMU noise characteristics improved from ETHOS to EXLs1 and EXLs3, but more importantly, the EXLs3 accelerometer has a higher dynamic range (16 g) than ETHOS and EXLs1 (6 g). We increased f_s for the EXLs3 setup compared to the previous recordings, following the suggestions for optimal ZUPT-PDR performance in [36].

The dataset is subdivided into three categories:

- Regular walking (R1–6): In these recordings, a person walked between pre-placed markers in a self-chosen order, trying to perform regular steps and standing still for a few seconds at the markers. [22] presents a detailed discussion of these recordings, including a comparison of SH-PDR and ZUPT-PDR in ActionSLAM preprocessing. In R1–6, there was only one type of landmarks (*standing still*).
- At-home recordings (H1–9): These were performed at the home of volunteers, either with the experimenter present and giving basic instructions, e.g., *cook dinner*, *close the windows*, etc. (H1–H5), or completely free (H6–H9). In the second case, we assisted participants in mounting and starting the sensor system, but did not further regulate their activities. H6, H7, and H9 were multi-floor recordings, and in H6, H8, and H9, multiple people were simultaneously equipped with sensors for tracking. All four landmark types (*standing still, sitting, stair low and stair high*) occurred in these experiments.
- Office recordings (O1-2): Here, people wore the sensors during a normal morning at work, including activities such as *getting print-outs*, *going to the bathroom*, and *making coffee in the kitchen*. They did not receive any specific instructions. The participants did not climb stairs, therefore ActionSLAM only observed *standing still* and *sitting* landmarks for these two experiments.

Except for the recordings R1–6, H8a, H9c and O1–2, the volunteers were not familiar with the principles of SLAM. However, all the participants knew that they were being tracked while wearing the sensors, which might have affected their behavior, as stated by the participant in H7. For the experiments R1–R6, H1–H5 and O1–O2, we collected detailed ground truth floor plans, in which we marked the expected locations of landmarks (chairs, water

[7] We experimentally chose the distance $L_{\max} = 250\,\text{m}$ for all our analyses.

[8] http://www.cupid-project.eu/node/44.

[9] http://www.exelmicroel.com/products_medical_exls3.html.

Fig. 6 Illustration of the sensor setups referenced in Table 3

Table 2 Characteristics and settings for the foot-mounted IMUs used in the experiments of Table 3

Sensor	ETHOS	EXLs1	EXLs3
Size (mm)	$46 * 24 * 14$	$45 * 32 * 5$	$54 * 33 * 15$
f_s (Hz)	127.5	100	200
Acc. range (g)	6	6	16
Gyr. range (°/s)	2,000	2,000	2,000

taps, windows, etc.). Our quantitative evaluation of 3D ActionSLAM is limited to these recordings. We qualitatively analyzed the remaining experiments by plotting the 3D ActionSLAM tracks and comparing them to the actual building layouts. Furthermore, we use synchronized videos as ground truth position measurements.

4.2 Performance measures

All iterations of ActionSLAM are probabilistic and the outcome tracks $\bar{s}^t$ and maps $\bar{\Theta}_t$ change with every execution of the algorithm, even for identical input data and parameter settings. Consequently, there is no guarantee for 3D ActionSLAM to converge to the correct map Θ. Furthermore, 3D ActionSLAM outcomes differ from ground truth data in scaling, rotation and translation. For performance analysis, we therefore need to first scale, rotate, and translate $\bar{\Theta}_t$ for best fit with the ground truth map Θ, using the multi-start interior point optimization algorithm from the MATLAB Optimization Toolbox.[10] We then assess 3D ActionSLAM along the probabilistic measures *robustness* r and *mean map accuracy* $\bar{d}_\Theta$:

- *Robustness* r is the percentage of *successful* 3D ActionSLAM runs. A successful 3D ActionSLAM run fulfills the following conditions: (1) the percentage of landmarks in the final 3D ActionSLAM map $\bar{\Theta}$ that is within the boundaries of the building must be $c_1 \geq 90$ %, and (2), the percentage of landmarks in the ground truth map that has a correspondence with distance below 1.5 m in $\bar{\Theta}$ must be $c_2 = 100$ %.
- *Map accuracy* $\bar{d}_\Theta$ is the mean Euclidean distance in 3D between the ground truth landmark coordinates and the center of the closest 3D ActionSLAM landmark of identical action type. We only take landmarks with ground truth correspondence into account. Map accuracy $\bar{d}_\Theta$ is related to *tracking accuracy* $\bar{d}_x$, since every landmark observation resets the error in estimating the person's position $\boldsymbol{x}_t$ to approximately $\bar{d}_\Theta$. In between action observations, the position error increases following the motion model rules.

4.3 Parameter optimization

We optimized the main parameters ($\sigma_{\phi,0}$, $k_{\phi,gyro}$, $\sigma_{x,0}$, $k_{x,gyro}$, r_0, r_1, p_0) of 3D ActionSLAM with respect to the performance measures r and $\bar{d}_\Theta$ in extensive parameter sweeps. For each parameter configuration and experimental dataset, we repeated 20 runs of 3D ActionSLAM with $N_p = 1{,}000$, applied to the datasets R1–R6, H1–H5 and O1–O2. Assuming that motion and observation model parameters only weekly influence each other, we performed independent sweeps for (1) $\sigma_{\phi,0}$ and $k_{\phi,gyro}$, (2) $\sigma_{x,0}$ and $k_{x,gyro}$, (3) r_0 and r_1, and (4) p_0.

We ran the optimization separately for each recording, and found similar, well-defined optima around the values listed in Table 4 for all experiments. The only exceptions were $\sigma_{x,0}$ and $k_{x,gyro}$, which seem to have little influence on the overall performance. Given that independent parameter optimizations for 13 recordings all resulted in similar optima, we conclude that the algorithm is insusceptible to parameter overfitting. 3D ActionSLAM is also insensitive to the motion model parameters for vertical displacement, so that we fixed them to $\sigma_{h,0} = 0.01$ m, $\sigma_{h,gyro} = 0.08$ m and $r_h = 0.1$ m after an initial experimental evaluation with recording H7. A sweep of the particle number $N_p = 100, \ldots, 5{,}000$ confirmed previous work (see [24]), indicating that $N_p = 1{,}000$ provides a good trade-off between robustness and computational costs.

We continued to use the settings in Table 4 as a *general parameter set* throughout the performance analysis. Similarly, we used the standard ZUPT-PDR parameters from [18] with all setups, although we expect slightly better results for ZUPT-PDR parameters that are optimized to the choice of foot-mounted IMU.

4.4 Performance analysis

Figure 7 shows example 3D ActionSLAM maps for H1–H8 and O1–O2, and Table 5 summarizes the performance

[10] http://www.mathworks.ch/ch/products/optimization/.

Table 3 Summary of the evaluation datasets with walking distances estimated from 3D ActionSLAM output tracks

Name	Foot sensor	Hip sensor	# Floors	Duration (min)	Distance (m)	Description
R1	EXLs1	Galaxy S4	1	3	57	Walk between 2 markers
R2	EXLs1	Galaxy S4	1	3	40	Walk between 3 markers
R3	EXLs1	Galaxy S4	1	4	127	Walk between 3 markers
R4	EXLs1	Galaxy S4	1	2	54	Walk between 3 markers
R5	EXLs1	Galaxy S4	1	3	100	Walk between 4 markers
R6	EXLs1	Galaxy S4	1	7	291	Walk between 7 markers
H1	ETHOS	ETHOS	1	101	675	Evening at home
H2	ETHOS	ETHOS	1	143	198	Evening at home
H3	ETHOS	ETHOS	1	74	353	Evening at home
H4	ETHOS	ETHOS	1	103	586	Evening at home
H5	EXLs1	Galaxy S4	1	44	168	Evening at home
H6a–b	EXLs1	Galaxy SII	1	197/195	141/151	2 People in flat cooking, having dinner, chatting on couch; synchronized
H7	EXLs1	Galaxy S4	3	113	722	Morning activities (breakfast, garden and basement work) in 3-floor house
H8a–c	EXLs1	Galaxy SII	2/2/3	152/111/88	381/213/260	3 People in shared flat at different evenings
H9a–d	EXLs3	Galaxy SII	2/1/2/3	116/96/67/104	500/228/194/402	4 People in shared flat (same as H8), morning activities; synchronized
O1	ETHOS	ETHOS	–	100	516	Office activities
O2	EXLs1	Galaxy S4	–	90	132	Office activities

analysis outcomes with $N_p = 1{,}000$ and 100 repeated runs. For all experiments in Table 5, the robustness r was between 84 and 99, and 93 % in average. The mean landmark positioning accuracy $\bar{d}_\Theta$ was 0.59 m, and never above 0.66 m. Figure 8 illustrates the convergence of r and $\bar{d}_\Theta$ with additional observations for the recordings H1–4. For the recordings R1–6, the robustness was 100 %, and the mean landmark positioning error $\bar{d}_\Theta = 0.47 \pm 0.11$ m.

In the single-floor scenarios, 3D ActionSLAM results are similar to the outcomes of previous implementations (see [24]), which confirms that adding the vertical displacement h to landmarks does not affect the algorithm's performance. The novel motion model introduced in Sect. 3.3.2 overcomes some issues regarding long phases without stance observation that we discussed in [24]: There we had to manually remove two long no-stance phases in H4 for convergence. This is not necessary anymore with the new motion model.

We only visually analyzed the 3D ActionSLAM performance in the multi-floor scenarios, by comparing the tracks $\vec{s}^t$ with the corresponding building floor plans. For the experiments H7–H9, the 3D ActionSLAM paths with standard settings as in Table 4 well correspond to the actual building layouts. Figure 7 presents exemplary maps and tracks for the multi-floor recordings H7 and H8. Videos on http://vimeo.com/search?q=actionslam demonstrate the mapping procedure: *ActionSLAM with Synchronized Video (H4)* shows a video of the experiment participant walking around at his together with the ActionSLAM position estimate, *ActionSLAM Tracking in Office Scenario* plots the person's path into the video of ceiling-attached cameras, and *ActionSLAM Convergence in At-Home Tracking Scenarios* presents the iterative mapping for the recordings H1–H5.

4.5 Real-time performance

While the evaluations above were done offline, we also developed SmartActionSLAM, an android application that collects data from a foot-mounted IMU and runs ZUPT-PDR, action recognition and the SLAM updating in real time on two of the phone's processors [22]. The current implementation runs a previous iteration of ActionSLAM, but the modifications for 3D ActionSLAM will only add minor complexity. Critical for real-time SmartActionSLAM execution is that the computation times per motion and observation update are smaller than the duration between two subsequent updates. Otherwise, delays in the location estimation will accumulate.

A profiling of the SmartActionSLAM app running on a Samsung Galaxy S4 phone showed that the computationally most costly functions are the observation update, and the resampling. In comparison, the execution times for ZUPT-PDR, action recognition and motion updates are negligible (< 10 % of total computation time). The computation times

Table 4 ActionSLAM parameter configuration used in performance analysis

Parameter	$\sigma_{\phi,0}$	$k_{\phi,gyro}$	$\sigma_{x,0}$	$k_{x,gyro}$	r_0	r_1	p_0	$\sigma_{h,0}$	$\sigma_{h,gyro}$	r_h	N_p
Value	0.25°	1.5°	0.01 m	0.01 m	0.25 m	0.25 m	0.02	0.01 m	0.08 m	0.1 m	1,000

per observation update T_{obs} and resampling T_{res} are both linear functions of the number of landmarks N_l in the particle maps, and the number of particles N_p. For $N_l = 5$ and $N_p = 1{,}000$, the mean T_{obs} was 0.5 s and $T_{res} = 0.69$ s. The total time per SLAM update if also taking the time per motion update $T_{mot} = 0.04$ s into account was $T_{tot} = T_{obs} + T_{res} + T_{mot} = 1.23$ s. As the time between two subsequent actions in our recordings was never below 5 s, SmartActionSLAM would not accumulate delays with this N_l and N_p. An extrapolation with fixed $N_p = 1{,}000$ indicates that the maximum number of landmarks with $T_{tot} < 5$ s is $N_l = 22$ (see Fig. 9), which is higher than the final number of landmarks in all our experiments, except for H7 ($N_l = 33$). Further code optimization and the use of the graphical processing unit will improve this performance. In addition, we may fix well-known landmarks, i.e., only use them for localization and no longer adapt their position in the observation update.

5 Discussion

Figure 7 illustrates that 3D ActionSLAM maps and paths are geometrically accurate and they well reflect the topology of building layouts: Rooms and the different areas within rooms are clearly distinguishable, in particular the position of sitting opportunities. The 3D ActionSLAM paths $\vec{s}^t$ connect landmarks without crossing walls, except when sometimes missing a door by up to 1 m. Complementary modalities such as Wifi may further improve the algorithm's robustness and reduce the number of required particles.

The deployment of 3D ActionSLAM is straightforward. However, the following considerations should be taken into account when using the system in real-world scenarios:

5.1 Accelerometer dynamic range

In particular, during stair descent, we often observed accelerations above 6 g, which exceeds the accelerometer dynamic range of our ETHOS and EXLs1 IMUs. This added large errors to the ZUPT-PDR estimate of vertical displacement $\hat{\delta h}_t$, typically underestimating the descent altitude change by 20–30 %. In the recordings H7 and H8, 3D ActionSLAM could compensate for this issue, but in two further experiments not presented here, we had to manually correct the ZUPT-PDR path estimates for convergence. We therefore discarded these recordings from further analyses. In the latest experiment (H9), we used EXLs3 IMUs with a dynamic range of 16 g, which solved the issue. We furthermore observed that the accelerations during stair descent were smaller when people were wearing shoes, rather than walking in socks. This is probably due to damping effects of shoe soles.

5.2 Missing stance phases

An important requirement for ZUPT-PDR to provide reliable odometry estimates $\hat{s}^t$ to the 3D ActionSLAM update phase is the regularity of stance observations, i.e., that the time T_t between subsequent stances remains small. During the swing phase of a step, the error of ZUPT-PDR grows exponentially, and therefore also the position uncertainty in $p(s_t|s_{t-1}, u_t)$ modeled with the 3D ActionSLAM particle cloud. There were incidents with $T_t > 3$ s in all the at-home and office experiments, typically when the person walked fast or swung his legs back and forth while sitting. Previous versions of ActionSLAM could already handle most of these incidents, for example by softening the stance detection criterion [24]. However, in H4, there are two phases of leg swinging with $T_t > 10$ s, which only the novel 3D ActionSLAM motion model can handle. This new model also works in the challenging multi-floor experiments, where phases of $T_t > 6$ s are frequent during fast stair descent. Nevertheless, long phases without stance remain a major source of errors and decrease the 3D ActionSLAM robustness r. An alternative approach would be to increase N_p when $N_p = 1{,}000$ particles can no longer adequately represent the probability distribution $p(s_t|s_{t-1}, u_t)$. KLD-resampling [19] is one technique supporting adaptive N_p that we will investigate in future.

5.3 User behavior

A requirement of 3D ActionSLAM is that the person sits and stands still repeatedly at the same locations. In the experiments we conducted, this was the case: People walked mostly between a set of specific places in their home, such as sofas, chairs, kitchen facilities, drawers, water taps, windows and light switches and they stood still or sat down at these locations. The few landmarks that 3D ActionSLAM inserted because participants stood still at random spots did not affect the overall convergence of the algorithm. Map maintenance automatically removed most of these landmarks. However, for some specific daily life

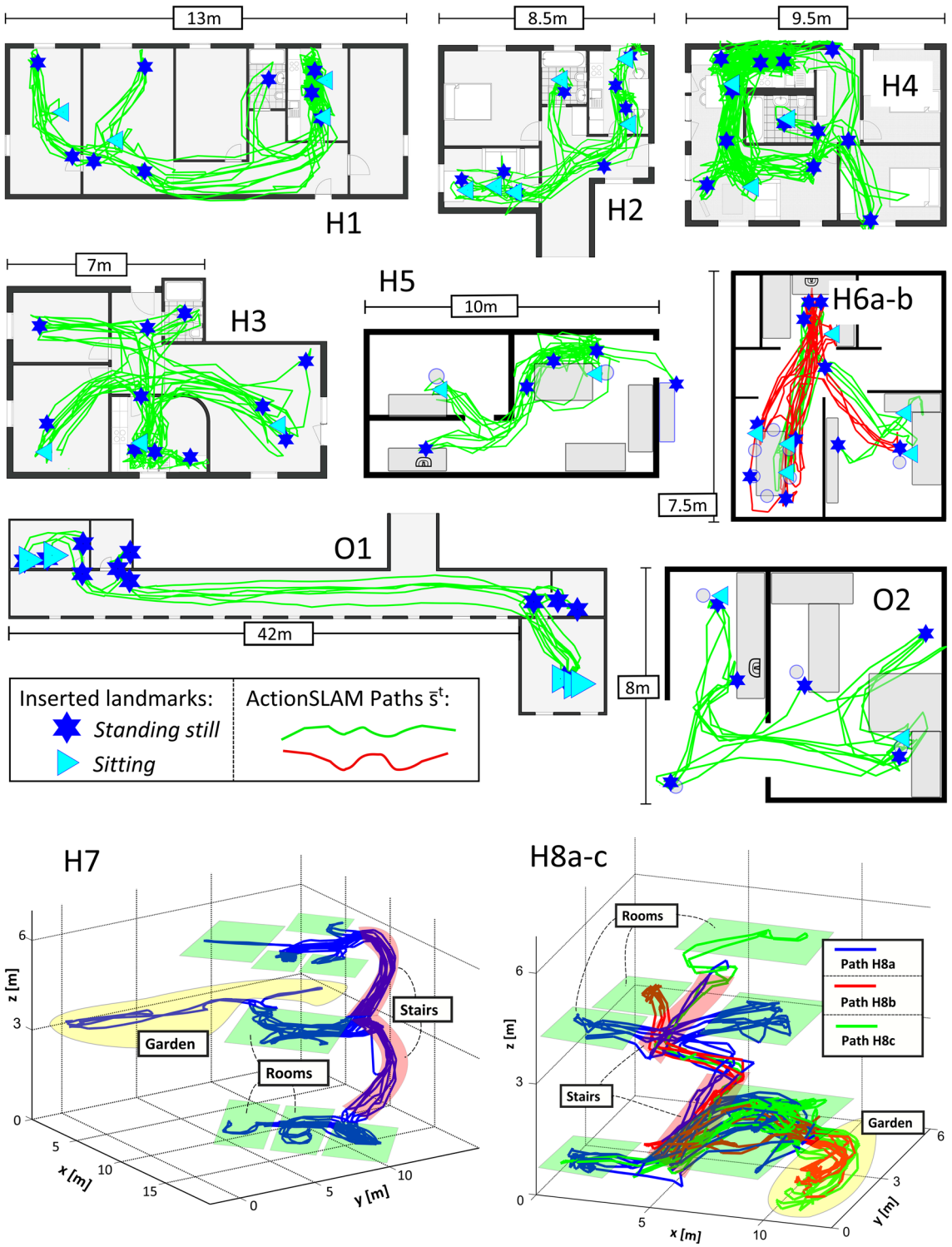

Fig. 7 Exemplary 3D ActionSLAM outputs, with the building layouts as background

activities such as hovering and cleaning, the ActionSLAM assumption regarding human behavior at home does not hold, as people are likely to stand still often and at non-specific places. It may be necessary to detect that the user is performing such an activity, and assign a state *unknown* to the observations during the activity, indicating that the current user behavior should not affect the mapping procedure. Liao et al. [30] proposed such an approach to

Table 5 Averaged performance results (see measures in Sect. 4.2) for 100 repeated 3D ActionSLAM runs with $N_p = 1{,}000$

Recording	H1	H2	H3	H4	H5	O1	O2
r (%)	84	90	96	96	98	86	99
$\bar{c}_1$ (%)	84	90	96	99	100	94	100
$\bar{c}_2$ (%)	85	91	96	97	98	90	99
$\bar{d}_\Theta$ in (m)	0.63 ± 0.11	0.61 ± 0.11	0.66 ± 0.08	0.44 ± 0.10	0.54 ± 0.11	0.79 ± 0.25	0.43 ± 0.09
$\bar{N}_l$	12.8 ± 1.7	13.1 ± 1.1	15.1 ± 1.2	16.9 ± 1.5	10.7 ± 1.1	15.7 ± 1.2	8.5 ± 0.7

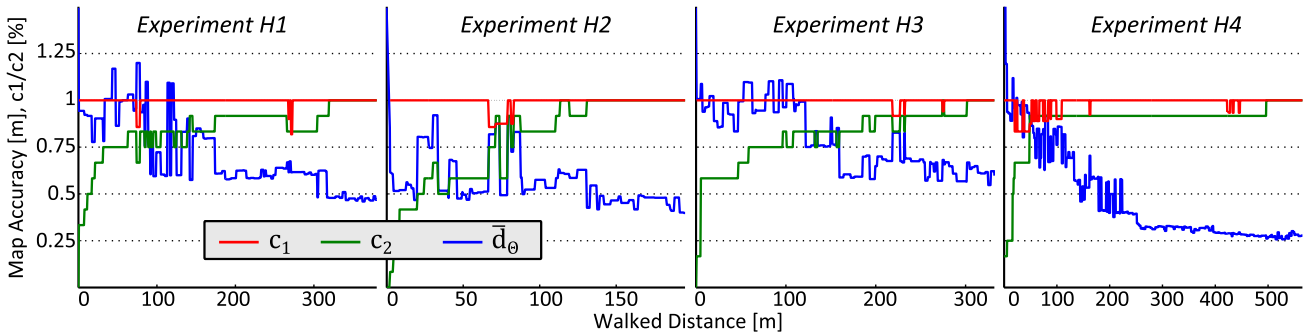

Fig. 8 Convergence of 3D ActionSLAM for the experiments H1–4. c_1 is the percentage of landmarks within the boundaries of the ground truth map, and c_2 the percentage of ground truth landmarks that the system observed. $\bar{d}_\Theta$ is the map accuracy. In the beginning, $c_2 = 0$ because the 3D ActionSLAM map is empty

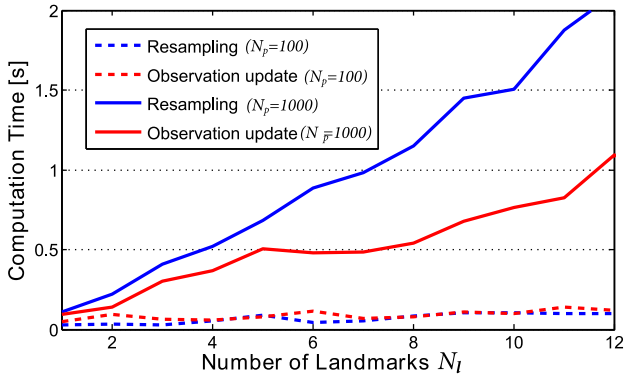

Fig. 9 Experimentally measured computation times per observation update and resampling as a function of N_l (according to [22]

account for unexpected user behavior when inferring transportation routines from GPS traces.

5.4 Battery lifetime

In Bluetooth streaming mode and with a sampling rate $f_s = 100$ Hz, the life time of our EXLs1 IMUs is about 4 h, which is sufficient for tracking during one evening. The ETHOS and EXLs3 sensors have similar lifetimes. However, for long-term deployment, people may not be willing to recharge batteries every 4 h. To solve this issue, we recently developed a new in-sole IMU with external battery, which runs for ~ 20 h.

5.5 Real-time feedback

A still open issue for real-time execution of 3D ActionSLAM is the recognition of false runs. For offline execution, robustness $r > 80\,\%$ is sufficient: To find the true map and path, we simply run 3D ActionSLAM multiple times and apply a majority vote, or let the user choose the best outcome track. In a real-time system, we, however, need to detect false maps automatically and restart the mapping process in response.

To sum up, with appropriate IMUs, 3D ActionSLAM can be deployed as it is to track people at home (and at work, if the work includes revisiting locations in a constrained area), except during some special activities such as hovering of the house. A typical use-case would be the monitoring of one or multiple house inhabitants during an evening at home. In the subsequent section, we introduce complementary techniques for visualizing and interpreting 3D ActionSLAM tracks.

6 Interpretation of 3D ActionSLAM tracks

3D ActionSLAM is a novel approach to collect location data of one or multiple home inhabitants with low deployment effort, low computational effort, and with comparatively small privacy intrusion. However, the algorithm as such only finds a path $\bar{s}^t$ within a local coordinate system,

and a list of landmarks $\bar{\Theta}_t$ that indicate where the person stood still, climbed stairs or sat down. For practical applications in health and behavior monitoring (see e.g., [4, 10, 12]), this data is difficult to interpret, in particular since paths and maps alone do not reveal any spatio-temporal information such as the duration and sequence of visits to a certain location. We therefore subsequently propose SLAM-specific techniques to align, segment, and visualize 3D ActionSLAM paths before presenting the tracking data to users. Figure 10 depicts the overall data flow for producing these visualizations: First of all, if multiple people are tracked, the individual paths $\vec{s}_P^t$ of people $P_1, \ldots, P_M$ are aligned into the same coordinate system. From a single path or multiple aligned paths, heat maps H_v are generated, which depict features such as the frequency of incidents (e.g., falls). In Sect. 6.3 we describe a novel method for deriving the building's room layout R_v from a heat map H_v that depicts the time spent at each location. Finally, given R_v and the aligned paths $\vec{s}_P^t$ of all inhabitants, we generate a narrative chart NC_v that projects the location of people to a single dimension before plotting it against time.

6.1 Multi-track alignment

6.1.1 Motivation

Each 3D ActionSLAM path $\vec{s}_{P_i}^t$ represents the person's motions in a different coordinate system C_i. To compare multiple paths and proceed with further analyses, the paths should be rotated and translated into a single coordinate system C_1. People may only share a part of an indoor environment, while other rooms are private and visited by a subset of the inhabitants. Therefore, a simple least-squares optimization of the distance between the visited coordinates of each path will usually fail. For collaborative exploration of large buildings, Puyol et al. [42] proposes to search for an optimal overlay of path subsets. This method requires that the participants share at least three joint corridor segments in the building.

6.1.2 Approach

We here present an alternative solution for tracks that start at the same place ($s_0 = \{0, 0, 0, \phi_0\}$). In a practical application, this could be the place where users recharge their sensors, or the entrance of the building. If s_0 is equal for all participants, there will only be a rotational offset $\delta\phi_{C_i}$ between the coordinate systems C_i and C_1. To find the rotation difference between a path $\vec{s}_{P_2,C2}^t$ and $\vec{s}_{P_1,C_1}^t$, our multi-track alignment solution counts for each possible heading the frequency of its occurrence in the path. Given the orthogonal layout of most buildings, the corresponding

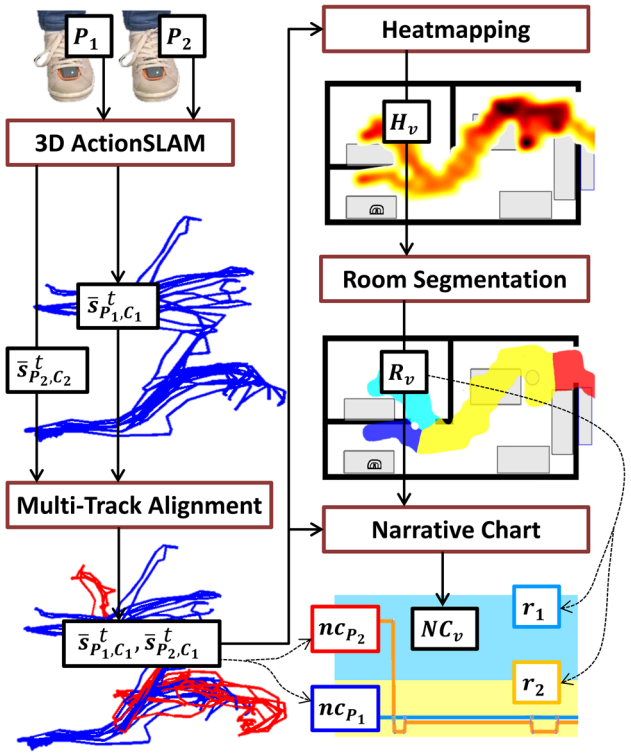

Fig. 10 This figure demonstrates the data flow in generating visualizations of 3D ActionSLAM outputs for a 2-person experiment (P_1 and P_2). As the two paths $\vec{s}_{P_1,C_1}^t$ and $\vec{s}_{P_2,C_2}^t$ have different coordinate systems C_1 and C_2, we first align them both in C_1. Then, we produce the heat map H_v and use it as input for room segmentation. The resulting segments R_v and the aligned paths can finally be combined to a narrative chart NC_v

histogram typically shows four peaks with 90° offset in between. Except in square-shaped buildings, only two out of these peaks have about the same height (at 180° angle difference). By looking for the best fit between the heading distribution of the walk by P_1 and the walk by P_2, we can reduce the number of possible rotation differences to two ($\delta\phi_{C_i}$, and $\delta\phi_{C_i} + 180°$). If P_1 and P_2 furthermore start facing in the approximately same direction ϕ_0, the rotation offset between C_1 and C_2 is small and we can just take the smaller of the two angles.

6.1.3 Results

Figure 7 shows overlays of the 3D ActionSLAM tracks H6a–b and H8a–c as found with this procedure.

6.2 Heat maps

6.2.1 Motivation

Hotspots in tracking are places at which a certain incident occurs particularly often. Some location-related incidents

of importance that wearable systems can detect are *falling* [49], and *freezing of gait* in Parkinson's disease [33]. 3D ActionSLAM offers an easily deployed extension to such systems for synchronized at-home position tracking, enabling the efficient identification of hotspots.

6.2.2 Approach

We use Kernel density heat maps [32] with a Gaussian kernel and bandwidth 0.4 m to visualize the frequency of events in an environment.

6.2.3 Results

Figure 11 depicts two example heat maps for *time spent at location*. The marked hotspots in this case correspond to areas at which the experiment participants spent most of their time (chairs and sofas).

6.3 Room segmentation

6.3.1 Motivation

For interpretation of a person's location behavior, it is often helpful to distinguish the rooms and semantic subareas of a home. Automated segmentation of SLAM paths to rooms can be achieved through synchronized recognition of actions that indicate the entering in a new space, e.g., *door opening* [21]. Recognizing door opening however requires additional, wrist-worn sensors. Another approach is to classify paths into room and corridor segments and then cluster these segments to find the room center points [1]. This method was demonstrated for simulated traces in an office building, but it requires that corridors are straight and long compared to the size of rooms.

6.3.2 Approach

Here, we propose a room segmentation algorithm inspired by image processing that is better suited for typical home floor plans. The first step of this method is to apply a threshold to the location heat map H_v such that $I_v = H_v > H_0$. I_v is a binary image that shows all accessed areas of the environment (see Fig. 11). Topological skeletonization with pruning of small side branches as in [6] (Code from http://www.cs.smith.edu/~nhowe/research/code/) then results in a skeleton S_v that highlights the major connections between subspaces in the environment. Given the extracted environment topology, the branches of S_v and the closest nonzero pixels of I_v get a room label r_i assigned.

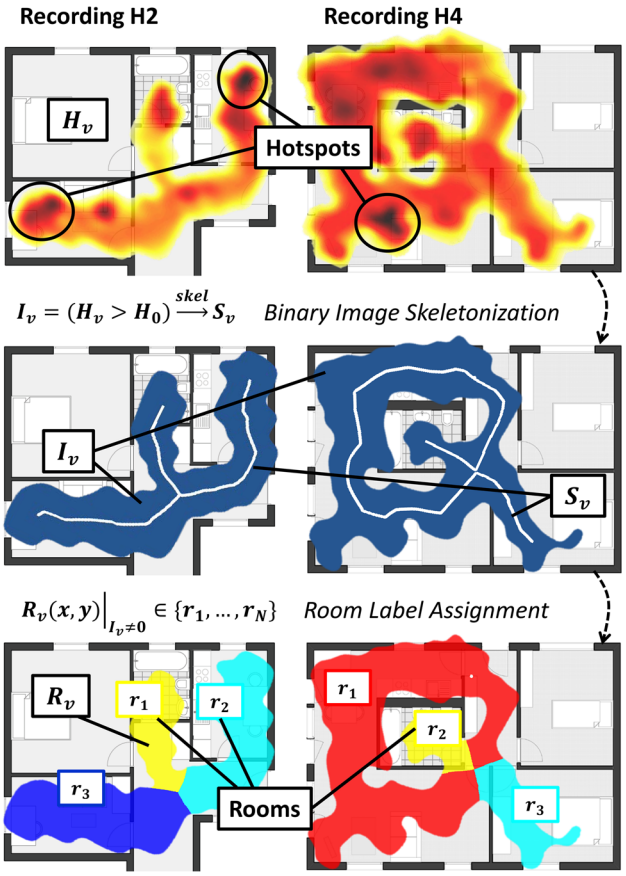

Fig. 11 The heat maps H_v show the time spent at locations for H2 and H4 (logarithmic color scale). These heat maps are then used to find the walking area skeletons S_v, and from them the room segmentation R_v

6.3.3 Results

With this approach, we could derive accurate representations R_v of the room layouts in all recordings (H1-9, O1-2). In multi-floor recordings, we applied the room segmentation to each floor separately.

6.4 Narrative charts

6.4.1 Motivation

Narrative charts visualize the interpersonal proximity within a group as a function of time. Such charts were for example used to visualize group performance parameters in firefighting (see [16]). In at-home tracking, narrative charts indicate in which room or subspace the inhabitants were at each time, and to which other people in the same room they were close at that moment. Therefore, narrative charts capture and reveal the social dynamics in a group of people sharing the same home.

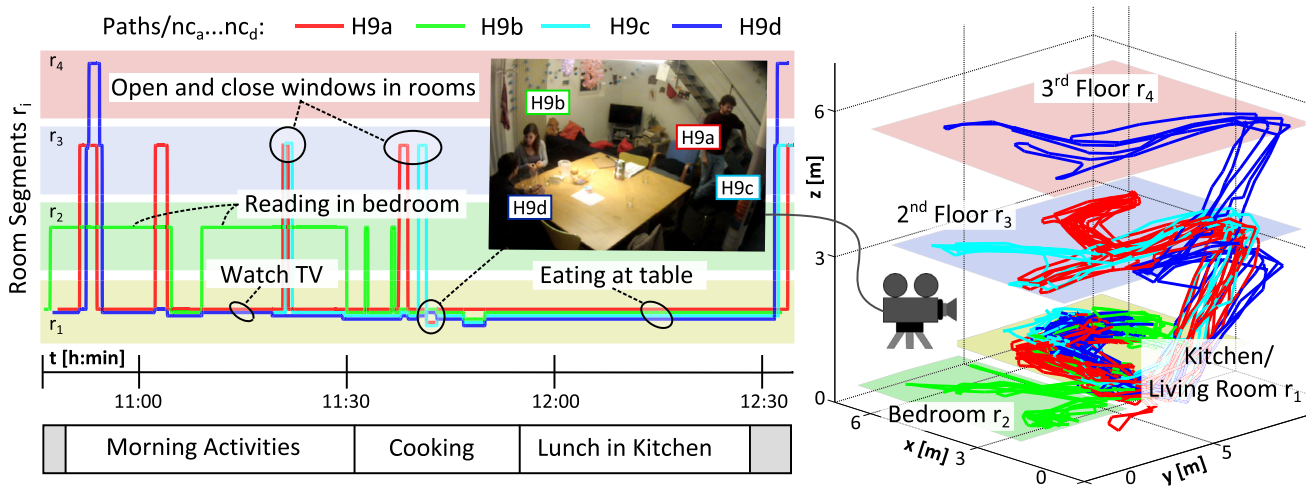

Fig. 12 This extract of the narrative chart (*left*) for recording H9 indicates that H9a and H9d spent most of the time together in the kitchen/living room, while H9b was in the bedroom (H9c started the recording later)—this information is not visible from the aligned tracks plotted on the *right*. The photograph depicts a situation where two pairs of people (H9a, c and H9b, d) communicated in separate groups, even though they were in the same room. The narrative chart visualizes this information through forking nc_a and nc_c from nc_b and nc_d within the same room category

6.4.2 Approach

To generate the narrative chart NC_v from aligned 3D ActionSLAM paths, the Euclidean distance $d_{i,j}$ between the people P_i and P_j is calculated for each combination of i and j. If P_i and P_j are in different rooms, $d_{i,j}$ is set to be large. Single-linkage clustering applied to the matrix $d_{i,j}$ with a maximum distance $d_{\max} = 3$ m then groups the people such that every person is within $d_{\max}$ of at least one other person in the same cluster. Cluster changes lead to the addition of nodes in the narrative chart and a regrouping of links in the diagram.

6.4.3 Results

Figure 12 depicts the narrative chart for recording H9. Different levels in NC_v correspond to different rooms/floor levels in R_v, and the lines nc_i correspond to the position of person P_i with respect to the other people. The video http://vimeo.com/90007230 aligns the chart evolution with the walks of the four experiment participants. None of the steps involved in the creation of this video from 3D ActionSLAM tracks requires user input.

7 Conclusion

We showed experimentally that stand-alone tracking of people in multi-floor homes is possible from a foot-mounted IMU and a hip-worn smartphone. 3D ActionSLAM is easy to deploy, unobtrusive, and nevertheless capable of robust tracking and mapping, inserting landmarks with a mean position error of 0.59 m. The addition of a novel motion model improved the handling of long phases without stance. Crucial to achieve reliable tracking in multi-floor environments is that the foot-mounted IMU has a sufficiently large accelerometer dynamic range to capture all foot movements. We confirmed the potential of the 3D ActionSLAM system in 23 individual recordings with a total walked distance of 6,489 m. Furthermore, we introduce multi-path alignment, kernel density heat maps, room segmentation and narrative charts as tools for visualization and interpretation of SLAM output tracks.

3D ActionSLAM is easy to set up, which facilitates a variety of location-aware applications in health care and remote monitoring. For example, we are applying 3D ActionSLAM to investigate the location correspondence of freezing of gait events in Parkinson's disease [33]. In future, 3D ActionSLAM may replace camera-based tracking in behavior analysis studies such as [3], and provide location-awareness as an additional feature in activity recognition tasks [53].

Acknowledgments The research leading to these results has received funding from the European Union—Seventh Framework Program (FP7/2007-2013) under Grant Agreement No 288516 (CuPiD project). We would furthermore like to thank all experiment participants as well as Sinziana Mazilu, Dario Caraci and Frederik Hess for their assistance with the SmartActionSLAM application, Sebastian Feese for his introduction to narrative charts, and Diego Romanelli, Bojan Milosevic and Filippo Casamassima for their support with the EXL sensors.

References

1. Alzantot M, Youssef M (2012) CrowdInside: automatic construction of indoor floorplans. In: Proceedings of the 20th international conference on advances in geographic information systems. ACM, New York, USA, pp 99–108
2. Angermann M, Robertson P (2012) FootSLAM: pedestrian simultaneous localization and mapping without exteroceptive sensors—hitchhiking on human perception and cognition. Proc IEEE 100:1840–1848
3. Ayers D, Shah M (2001) Monitoring human behavior from video taken in an office environment. Image Vis Comput 19(12):833–846
4. Boric-Lubeke O, Lubecke V (2002) Wireless house calls: using communications technology for health care and monitoring. Microw Mag IEEE 3(3):43–48
5. Brajdic A, Harle R (2013) Walk detection and step counting on unconstrained smartphones. In: Proceedings of the 2013 ACM international joint conference on pervasive and ubiquitous computing. ACM, New York, pp 225–234
6. Brandt JW, Algazi VR (1992) Continuous skeleton computation by voronoi diagram. CVGIP Image Underst 55(3):329–338
7. Brasche S, Bischof W (2005) Daily time spent indoors in german homes-baseline data for the assessment of indoor exposure of german occupants. Int J Hyg Environ Health 208(4):247–253
8. Dissanayake MWMG, Newman P, Clark S, Durrant-Whyte H, Csorba M (2001) A solution to the simultaneous localization and map building (slam) problem. IEEE Trans Robot Autom 17(3):229–241
9. Douc R, Cappé O (2005) Comparison of resampling schemes for particle filtering. In: Proceedings of the 4th international symposium on image and signal processing and analysis, pp 64–69
10. D'Souza M, Ros M, Karunanithi M (2012) An indoor localisation and motion monitoring system to determine behavioural activity in dementia afflicted patients in aged care. Electr J Health Inf. http://ro.uow.edu.au/hbspapers/3108/
11. Durrant-Whyte H, Bailey T (2006) Simultaneous localization and mapping: part I. IEEE Robot Autom Mag 13(2):99–110
12. Eagle N, Pentland A (2006) Reality mining: sensing complex social systems. Pers Ubiquitous Comput 10(4):255–268
13. Eisa S, Moreira A (2012) Requirements and metrics for location and tracking for ambient assisted living. In: 2012 international conference on indoor positioning and indoor navigation (IPIN), pp 1–7
14. Elnahrawy E, Li X, Martin RP (2004) The limits of localization using signal strength: a comparative study. In: 2004 First Annual IEEE communications society conference on sensor and ad hoc communications and networks, 2004. IEEE SECON 2004. pp 406–414
15. Faragher RM, Sarno C, Newman M (2012) Opportunistic radio slam for indoor navigation using smartphone sensors. In: Position location and navigation symposium (PLANS), 2012 IEEE/ION, pp 120–128
16. Feese S, Arnrich B, Troster G, Burtscher M, Meyer B, Jonas K (2013) Coenofire: monitoring performance indicators of firefighters in real-world missions using smartphones. In: Proceedings of the 2013 ACM international joint conference on pervasive and ubiquitous computing, ACM, pp 83–92
17. Ferris B, Fox D, Lawrence ND (2007) Wifi-slam using gaussian process latent variable models. In: IJCAI, vol 7, pp 2480–2485
18. Fischer C, Talkad Sukumar P, Hazas M (2012) Tutorial: implementation of a pedestrian tracker using foot-mounted inertial sensors. In: IEEE pervasive computing, pp 1–19
19. Fox D (2003) Adapting the sample size in particle filters through KLD-sampling. Int J Robot Res 22(12):985–1003
20. Foxlin E (2005) Pedestrian tracking with shoe-mounted inertial sensors. In: IEEE Computer Graphics and Applications, pp 38–46
21. Grzonka S, Karwath A, Dijoux F, Burgard W (2012) Activity-based estimation of human trajectories. IEEE Trans Robot 28(1):234–245
22. Hardegger M, Mazilu S, Caraci D, Hess F, Roggen D, Tröster G (2013) ActionSLAM on a smartphone: at-home tracking with a fully wearable system. In: 2013 International conference on indoor positioning and indoor navigation (IPIN), pp 99–106
23. Hardegger M, Roggen D, Mazilu S, Tröster G (2012) Action-SLAM: using location-related actions as landmarks in pedestrian SLAM. In: 2012 international conference on indoor positioning and indoor navigation (IPIN)
24. Hardegger M, Tröster G, Roggen D (2013) Improved Action-SLAM for long-term indoor tracking with wearable motion sensors. In: Proceedings of the 17th annual international symposium on International symposium on wearable computers, ACM, pp 1–8
25. Harle R (2013) A survey of indoor inertial positioning systems for pedestrians. IEEE Commun Surv Tutor 15(3):1281–1293
26. Harms H, Amft O, Winkler R, Schumm J, Kusserow M, Tröster G (2010) Ethos: miniature orientation sensor for wearable human motion analysis. In: 2010 IEEE sensors, pp 1037–1042
27. Jimenez A, Seco F, Prieto C, Guevara J (2009) A comparison of pedestrian dead-reckoning algorithms using a low-cost MEMS IMU. In: IEEE international symposium on intelligent signal processing, pp 37–42
28. Jimenez Ruiz AR, Seco Granja F, Prieto Honorato JC, Guevara Rosas JI (2012) Accurate pedestrian indoor navigation by tightly coupling foot-mounted imu and rfid measurements. IEEE Trans Instrum Meas 61(1):178–189
29. Leech JA, Nelson WC, Burnett RT, Aaron S, Raizenne ME (2002) It's about time: a comparison of canadian and american time-activity patterns. J Expo Anal Environ Epidemiol 12(6):427–432
30. Liao L, Patterson DJ, Fox D, Kautz H (2007) Learning and inferring transportation routines. Artif Intell 171(5):311–331
31. Liu H, Darabi H, Banerjee P, Liu J (2007) Survey of wireless indoor positioning techniques and systems. IEEE Trans Syst Man Cybern Part C Appl Rev 37(6):1067–1080
32. Maciejewski R, Rudolph S, Hafen R, Abusalah A, Yakout M, Ouzzani M, Cleveland WS, Grannis SJ, Ebert DS (2010) A visual analytics approach to understanding spatiotemporal hotspots. IEEE Trans Vis Comput Graph 16(2):205–220
33. Mazilu S, Hardegger M, Zhu Z, Roggen D, Tröster G, Plotnik M, Hausdorff J (2012) Online detection of freezing of gait with smartphones and machine learning techniques. In: Proceedings of the 6th international conference on pervasive computing technologies for healthcare, vol 3, pp 123–130
34. Mirowski P, Ho TK, Yi S, MacDonald M (2013) Signalslam: simultaneous localization and mapping with mixed wifi, bluetooth, LTE and magnetic signals. In: 2013 international conference on indoor positioning and indoor navigation (IPIN), pp 357–366
35. Montemerlo M, Thrun S, Koller D, Wegbreit B (2002) Fastslam: a factored solution to the simultaneous localization and mapping problem. In: Proceedings of the national conference on artificial intelligence, pp 593–598
36. Munoz Diaz E, Heirich O, Khider M, Robertson P (2013) Optimal sampling frequency and bias error modeling for foot-mounted imus. In: 2013 international conference on indoor positioning and indoor navigation (IPIN), pp 245–253
37. Nilsson JO, Skog I, Handel P, Hari K (2012) Foot-mounted ins for everybody-an open-source embedded implementation. In: Position location and navigation symposium (PLANS), 2012 IEEE/ION, pp 140–145

38. Park K, Shin H, Cha H (2013) Smartphone-based pedestrian tracking in indoor corridor environments. Pers Ubiquitous Comput 17(2):359–370
39. Pradeep V, Medioni G, Weiland J (2010) Robot vision for the visually impaired. 2010 IEEE computer society conference on computer vision and pattern recognition—workshops, pp 15–22
40. Prateek G, Girisha R, Hari K, Handel P (2013) Data fusion of dual foot-mounted ins to reduce the systematic heading drift. In: 2013 4th international conference on intelligent systems modelling simulation (ISMS), pp 208–213
41. Priyantha NB, Chakraborty A, Balakrishnan H (2000) The cricket location-support system. In: Proceedings of the 6th annual international conference on mobile computing and networking, ACM, pp 32–43
42. Puyol MG, Frassl M, Robertson P (2012) Collaborative mapping for pedestrian navigation in security applications. In: Future security, Springer, Berlin, pp 49–60
43. Qian J, Ma J, Ying R, Liu P (2013) Rpnos: Reliable pedestrian navigation on a smartphone. In: Geo-informatics in resource management and sustainable ecosystem, Springer, Berlin, pp 188–199
44. Robertson P, Angermann M, Khider M (2010) Improving simultaneous localization and mapping for pedestrian navigation and automatic mapping of buildings by using online human-based feature labeling. In: Position location and navigation symposium (PLANS), 2010 IEEE/ION, pp 365–374
45. Robertson P, Frassl M, Angermann M, Doniec M, Julian BJ, Garcia Puyol M, Khider M, Lichtenstern M, Bruno L (2013) Simultaneous localization and mapping for pedestrians using distortions of the local magnetic field intensity in large indoor environments. In: 2013 international conference on indoor positioning and indoor navigation (IPIN), pp 925–934
46. Romanovas M, Goridko V, Al-Jawad A, Schwaab M, Traechtler M, Klingbeil L, Manoli Y (2012) A study on indoor pedestrian localization algorithms with foot-mounted sensors. In: 2012 international conference on indoor positioning and indoor navigation (IPIN), pp 1–10
47. Scott J, Bernheim Brush A, Krumm J, Meyers B, Hazas M, Hodges S, Villar N (2011) Preheat: controlling home heating using occupancy prediction. In: Proceedings of the 13th international conference on ubiquitous computing, ACM, pp 281–290
48. Shin H, Chon Y, Cha H (2012) Unsupervised construction of an indoor floor plan using a smartphone. IEEE Trans Syst Man Cybern Part C Appl Rev 42(6):889–898
49. Sposaro F, Tyson G (2009) iFall: an android application for fall monitoring and response. In: Annual international conference of the IEEE engineering in medicine and biology society, pp 6119–6122
50. Steinhoff U, Schiele B (2010) Dead reckoning from the pocket—an experimental study. In: 2010 IEEE international conference on pervasive computing and communications (PerCom), pp 162–170
51. Woodman O, Harle R (2008) Pedestrian localisation for indoor environments. In: Proceedings of the 10th international conference on Ubiquitous computing, ACM, pp 114–123
52. Woodman OJ (2007) An introduction to inertial navigation. Technical report, Cambridge University Press, Cambridge
53. Zhu C, Sheng W (2011) Motion- and location-based online human daily activity recognition. Pervasive Mobile Comput 7(2):256–269