S-SMART: A Unified Bayesian Framework for Simultaneous Semantic Mapping, Activity Recognition, and Tracking

MICHAEL HARDEGGER, Wearable Computing Laboratory, ETH Zürich
DANIEL ROGGEN, Sensor Technology Research Centre, University of Sussex
ALBERTO CALATRONI and GERHARD TRÖSTER, Wearable Computing Laboratory, ETH Zürich

The machine recognition of user trajectories and activities is fundamental to devise context-aware applications for support and monitoring in daily life. So far, tracking and activity recognition were mostly considered as orthogonal problems, which limits the richness of possible context inference. In this work, we introduce the novel unified computational and representational framework S-SMART that simultaneously models the environment state (semantic mapping), localizes the user within this map (tracking), and recognizes interactions with the environment (activity recognition). Thus, S-SMART identifies which activities the user executes where (e.g., *turning a handle* next to a *window*), and reflects the outcome of these actions by updating the world model (e.g., *the window is now open*). This in turn conditions the future possibility of executing actions at specific places (e.g., *closing the window* is likely to be the next action at this location). S-SMART works in a self-contained manner and iteratively builds the semantic map from wearable sensors only. This enables the seamless deployment to new environments.

We characterize S-SMART in an experimental dataset with people performing hand actions as part of their usual routines at home and in office buildings. The framework combines dead reckoning from a foot-worn motion sensor with template-matching-based action recognition, identifying objects in the environment (windows, doors, water taps, phones, etc.) and tracking their state (open/closed, etc.). In real-life recordings with up to 23 action classes, S-SMART consistently outperforms independent systems for positioning and activity recognition, and constructs accurate semantic maps. This environment representation enables novel applications that build upon information about the arrangement and state of the user's surroundings. For example, it may be possible to remind elderly people of a window that they left open before leaving the house, or of a plant they did not water yet, using solely wearable sensors.

Categories and Subject Descriptors: H.1.2 [**User/Machine Systems**]: Human Information Processing; I.2.6 [**Learning**]: Knowledge Acquisition; I.5.1 [**Models**]: Statistical

General Terms: Algorithms, Design, Experimentation

Additional Key Words and Phrases: Semantic mapping, localization, SLAM, activity recognition, template matching, particle filter, wearable sensors, context awareness

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).

Authors' addresses: M. Hardegger, A. Calatroni, and G. Tröster, Wearable Computing Laboratory, Gloriastrasse 35, ETH Zürich, Switzerland; email: michael.hardegger@ife.ee.ethz.ch; D. Roggen, Sensor Technology Research Centre, University of Sussex, UK.

DOI: http://dx.doi.org/10.1145/2824286

ACM Reference Format:
Michael Hardegger, Daniel Roggen, Alberto Calatroni, and Gerhard Tröster. 2016. S-SMART: A unified bayesian framework for simultaneous semantic mapping, activity recognition and tracking. ACM Trans. Intell. Syst. Technol. 7, 3, Article 34 (February 2016), 28 pages.
DOI: http://dx.doi.org/10.1145/2824286

1. INTRODUCTION

The increase in popularity of wearable devices with integrated sensing (e.g., smartphones, smart watches, smart glasses, sensorized shoe insoles, etc.) triggered a trend toward monitoring various aspects of personal behavior [Bajarin 2014] and toward context-aware assistance. Two commonly analyzed characteristics of daily life are the location of a person and the activities he or she performs. Location monitoring may reveal places of frequent falls [Huang et al. 2009] and expose the social dynamics in a household or workplace [Eagle and Pentland 2006]. On the other hand, wearable activity recognition can count the number of smoked cigarettes per day [Scholl et al. 2013], activate real-time audio cueing in Parkinsonian gait [Mazilu et al. 2012], and assess sport skills through virtual trainers [Ahmadi et al. 2014]. Changes in daily-life activity routines may be valuable outcome measures for medical therapy [Seiter et al. 2013]. Furthermore, feedback on performed activities can motivate people to adhere to a healthy lifestyle [Free et al. 2013]. While ambient sensing is often explored for assisted living in smart homes [Van Kasteren et al. 2008; Rashidi and Cook 2013], wearable sensing, which we consider here, has the advantage of easy deployment (possibly as simple as downloading an app) and being available wherever the user goes.

So far, activity recognition and location tracking from wearable sensors have mostly been considered as orthogonal problems, with specific setups and signal processing techniques developed for each task. *Wearable activity recognition systems* spot signal patterns from body-worn sensors corresponding to specific motions [Bao and Intille 2004], and possibly the sequence of executed tasks [Patterson et al. 2005].

State-of-the-art activity recognition systems generally do not keep track of an environment representation that describes the a priori probability of performing activities given the past behavior at a place. Such prior information would prevent recognition errors: for example, a certain hand movement can only correspond to *opening a window* when the user is actually next to a closed window. The lack of a suitable environment representation in state-of-the-art systems also imposes limits on the complexity of scenarios in which wearable activity recognition may be applied. Interactions with multiple identical objects are indistinguishable to such systems. For example, the opening of two separate windows will usually result in similar motion measurements, which cannot be differentiated without knowledge about the location at which the motion takes place. To overcome these limitations, previous work suggested to enhance activity recognition with location awareness (e.g., Wang et al. [2012a], Alvarez-Alvarez et al. [2010], and Stiefmeier et al. [2008]). However, these systems require the supervised learning or preprogramming of the conditional probabilities for performing activities at each place. These approaches also do not adapt the conditional probabilities in response to user activities that modify the environment, for example, the closing of a window.

Wearable-only localization systems (i.e., not requiring ambient infrastructure) typically integrate the signals of body-worn motion sensors to estimate position [Foxlin 2005]. This open-loop approach suffers from error accumulation due to the integration of noisy sensor signals and is only useful for short-term tracking. Simultaneous Localization and Mapping (SLAM) addresses the issue by correcting the open-loop path when environmental landmarks are reobserved [Durrant-Whyte and Bailey 2006]. Internally, SLAM iteratively builds a map that represents the environment. The most common exteroceptive sensors in SLAM are laser range scanners [Cinaz and Kenn 2008] and cameras [Kourogi and Kurata 2003], although both of them are not suitable in many scenarios due to privacy concerns and obtrusiveness. Recently, it was suggested to exploit a user's location-specific motion patterns as environment observations, for example, characteristics of the walk [Angermann and Robertson 2012] or location-related activities [Grzonka et al. 2012; Hardegger et al. 2012]. However, these existing approaches do not handle potential recognition errors in the observations. This either leads to obtrusive sensor setups, to ensure correct action recognition, or to limited robustness and accuracy in the tracking outcome as landmark observations may be erroneous. Also, the maps these algorithms create are static; that is, they do not consider that a user's activities may affect his or her surroundings.

In summary, the state-of-the-art approaches for activity recognition and location tracking only inform each other in limited ways. This imposes constraints on the complexity of context-aware scenarios that can be envisioned currently. In this article, we demonstrate a novel fusion approach that considers location tracking and activity recognition as a joint estimation problem, which exploits the complementary information between *what we do* and *where we are*. The key to the proposed unified Bayesian framework is the environment state representation by means of a dynamic semantic map, which we define as a set of objects with state attributes and associated activity probabilities. For example, this can be the set of windows and doors in a flat, each with possible states *open* and *closed*. As a Bayesian representation, the position, type, and state of each object remain probabilistic, which is crucial considering that the inputs to the framework tend to be error prone. For example, an observation of the action *opening a window* will lead to the insertion of an object landmark in the semantic map with type *window* and state *open* at the current user location. However, future observations may indicate that the initial observation was false and the actual object at this place is a door. It can also be that the once-opened window was closed in between by another person, or that the actual position of the window was incorrectly estimated at the first visit. In all these cases, the algorithm must be able to correct errors through continuous accumulation of location-related context information. We refer to the problem of building a map with probabilistic object attributes and using it in return as a prior in activity recognition and localization as *Simultaneous Semantic Mapping, Activity Recognition and Tracking (S-SMART)*. To the best of our knowledge, S-SMART is the first stand-alone approach for learning and updating a probabilistic map representation of the environment and then applying this map to stateful activity recognition and localization. Since objects in S-SMART can change their state in response to user interactions, for example, the closing of a window, we refer to the semantic map as *dynamic*. Figure 1 depicts the data flow and main components of the proposed framework.

The main advantages of S-SMART compared to systems that consider localization and activity recognition as orthogonal problems are the following:

—**Unsupervised learning of location-to-activity mapping:** S-SMART internally builds a semantic map that models the probability of executing an activity at each place, without the need for any pretrained location-to-activity models. This enables fast deployment to new users and environments. Training is only necessary for matching motions to actions.

—**Handling of errors in landmark-type observation:** Even activities that the system recognizes with low confidence may improve the overall location and action estimation, as S-SMART handles the activity observations in a probabilistic manner. Existing activity-based localization systems (e.g., Hardegger et al. [2015]) are restricted to activities that they recognize with high confidence.

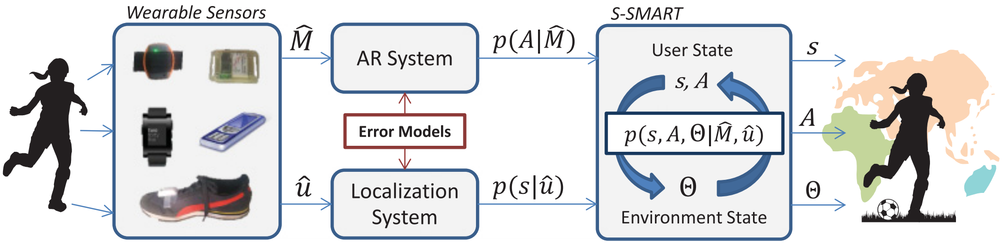

Fig. 1. Data flow of the S-SMART processing chain. The data of wearable sensors (raw motion data $\hat{M}$, preprocessed foot step data $\hat{u}$) is processed into estimates of the user's pose s and his or her activities A, with potential errors described by appropriate models. Both the localization system and the Action Recognition (AR) system are assumed to be influenced by errors, which appropriate models describe. To correct these errors, S-SMART fuses the location and activity inputs in an iterative Bayesian estimation process: it builds a semantic map Θ and uses this environment representation in return as a prior for better estimating the person's path s and actions A.

—**Activity disambiguation:** The combined framework automatically distinguishes a person's activities performed at different places, even if two activities are identical except for their location (e.g., *opening a window* in the kitchen and in the living room, or *working* at home and working in the office).

—**Stateful Mapping:** The system models the state of the environment. As a consequence, it can adapt the prior probabilities for activities and location according to the past sequence of activities at a location.

Aside from enhancing activity recognition and location tracking from independent systems (e.g., pedestrian dead reckoning for localization; template matching for action recognition), the state representation through dynamic semantic maps enables fundamentally new context-aware applications. As an example, consider the elusive "cognitive prosthesis" for patients with dementia. Such an assistant can only be realized through a wearable system having the same characteristics as S-SMART. In particular, it needs to remember which windows in a home are open to tell the user that he or she should close them. S-SMART may also keep track of how often the user opened the fridge or remember which plants have been watered, and it could inform a person about the drawers he or she used throughout the day (e.g., when he or she is looking for his or her keys). Another application scenario for S-SMART is quality control in factories. In these settings, it is often essential that a certain sequence of actions at different workstations is performed in the right order, and without forgetting intermediate steps. Quality control systems based on a combination of predeployed infrastructure and wearable sensors were, for example, investigated in Stiefmeier et al. [2008]. In the future, S-SMART may simplify deployment of such tools, as it requires no prior installation effort.

In this article, we introduce a Rao-Blackwellized particle filter implementation of S-SMART and present the generic algorithm framework, as well as a specific system for fusion of short hand motions (*actions*) with open-loop tracking from a foot-mounted Inertial Measurement Unit (IMU). We evaluate this system with people performing daily-life activities in workplace and home environments. In total, we collected 20 recordings with people executing 960 labeled actions while walking a total distance of 8.42km. In scenarios with instructed task sequences and up to 23 action classes, the mean S-SMART action recognition F_1 score was 89%. For real-life scenarios with people working in an office or cooking and cleaning at home, the recognition performance dropped to 68% but was still better than in the location- and state-agnostic case (53%). Except for two recordings, the tracking and mapping robustness was always 100%, meaning that every single execution of the probabilistic S-SMART algorithm with optimized parameter settings resulted in a topologically accurate posterior path and map estimate. As such, S-SMART outperforms systems that do not fuse location and activity (here, we used pedestrian dead reckoning and template matching as reference algorithms), and it achieved similar accuracies to systems that require either ground-truth location or activity input for enhancing the estimate of the other.

2. RELATED WORK

2.1. Activity Recognition from Body-Worn Sensors

The state of the art in wearable activity recognition aims at spotting activity-specific patterns in the signals of body-attached sensors, in particular accelerometers and gyroscopes. For this purpose, classifiers map the input signals to predefined activity classes. Depending on the type and duration of the activities to be recognized, the raw signals are preprocessed into descriptive features [Bulling et al. 2014], or the system compares the input data to prerecorded templates with similarity metrics such as Dynamic Time Warping (DTW, Berndt and Clifford [1994]) or Longest-Common-SubSequence matching (LCSS, Nguyen-Dinh et al. [2012]). While state-of-the-art approaches can robustly differentiate some activity classes (e.g., sitting, walking, running, etc., Bao and Intille [2004]), the interpersonal and intrapersonal variability of human motion is a challenge for disambiguation of more complex activity categories (e.g., opening a dishwasher, cleaning the table, etc., Chavarriaga et al. [2013]).

Motion-based systems may be enhanced through complementary modalities such as location. For example, GPS traces can indicate when a person works (at his or her workplace), shops (in the mall), eats (in the restaurant), and so forth [Zheng et al. 2010] and preselect domain-specific classifiers for the subset of activities possible at a location. Such location-aware activity recognition systems need to know the conditional probabilities for performing each activity at a given place. Previous work employed supervised training for this purpose [Lu and Fu 2009] or preprogrammed the mapping from prior knowledge about the activity routines [Stiefmeier et al. 2008; Wang et al. 2012a]. For real-life applications, both approaches are impractical due to the large deployment effort involved in extracting the location-activity relationship. Liao et al. [2007] present an alternative semisupervised system for recognizing transportation routines. However, this system requires voluntary user labels to build and update a model of how location patterns relate to transportation modes. Instead of GPS, indoor positioning technologies as described in the next section may be used for location-to-activity mapping, but the limitations remain the same.

2.2. Location Tracking from Body-Worn Sensors

Most indoor localization systems require prior information about the environment (e.g., building layouts [Woodman and Harle 2008], fingerprints of radio-frequency signals [Honkavirta et al. 2009]) or predeployed infrastructure (e.g., RFID beacons [Ruiz et al. 2012]). The effort of installing these systems and collecting prior maps is considerable and not appropriate for localization in private environments, where the cost per user becomes prohibitive [Harle 2013]. To avoid this effort, a number of fully stand-alone wearable tracking systems have been proposed. These may be categorized into two groups: First are the *open-loop* tracking systems that estimate position through integration of the motion signals from body-worn IMUs. The best results were achieved with foot-mounted sensors and corrections of the velocity estimate when the foot is on the ground (error accumulation may be as low as 0.5% of the traveled distance [Foxlin 2005]). Nevertheless, the positioning error of every open-loop tracking system grows with time, making it unreliable for long-term use. *Closed-loop* approaches combine

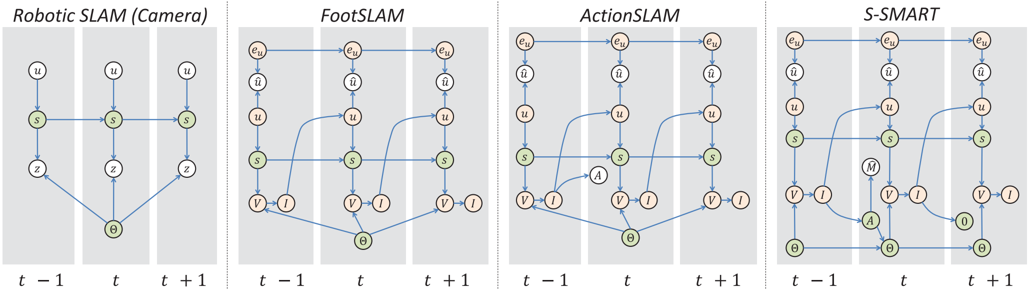

Fig. 2. Dynamic Bayesian Network (DBN) representation of SLAM algorithms and S-SMART. In robotics, for example, cameras estimate the relative position of landmarks z^t in a map Θ and then correct errors in the robot's odometry u^t. In pedestrian tracking (FootSLAM, ActionSLAM, S-SMART), the odometry is usually unknown, and only observations $\hat{u}^t$ of the persons's path are observable, for example, from body-worn motion sensors. Other than in robotics, there is direct access neither to the sensory information V^t on which the central nervous system bases its decisions nor to the intentions I^t the user tries to realize by performing steps u^t and activities A^t. Instead, FootSLAM [Angermann and Robertson 2012] takes the person's steps u^t themselves as environment observation, assuming that people will repeatedly walk straight in corridors. ActionSLAM [Hardegger et al. 2012] takes observations of simple activities A^t, which it recognizes with high confidence, and builds a map Θ by assigning locations to these activities. The proposed S-SMART framework applies motion measurements $\hat{M}$ that it relates to activities through a probabilistic model $p(A_t|\hat{M}_t)$. As a result, S-SMART is robust with respect to errors in the recognition of A_t and can correct action recognition errors in return. Furthermore, the map Θ^t in S-SMART is dynamic; that is, it can change in response to a user's interactions A^t.

open-loop tracking with exteroceptive sensing (e.g., laser range scanners [Cinaz and Kenn 2008] or cameras [Kourogi and Kurata 2003]) and reset the accumulated position error when landmarks in the environment are reobserved. As these methods internally build a map of the environment that assigns a location to observed landmarks, they are referred to as SLAM [Durrant-Whyte and Bailey 2006]). SLAM is often implemented as a Bayesian filter that fuses motion measurements with exteroceptive sensing to learn what the environment looks like, and where the person is within this environment. Common methods for solving this fusion task build on Extended Kalman Filters (EKF, Dissanayake et al. [2001]) and Rao-Blackwellized Particle Filters (RBPF, [Montemerlo et al. 2002]). Particle filters inherently support multihypothesis tracking, and they can model non-Gaussian open-loop tracking and observation errors.

Aside from the implementation algorithm, the choice of the exteroceptive sensor modality is one of the main distinctions of a SLAM system. Lasers and cameras tend to be impractical modalities for use in daily life, due to obtrusiveness, privacy concerns, and battery requirements [Liu et al. 2007]. As an alternative for pedestrian SLAM systems, location-specific motions that can be recognized from inertial measurements were proposed. Examples for environments that induce specific motions include corridors (leading to straight walks) [Angermann and Robertson 2012], corners [Park et al. 2013], stairs [Grzonka et al. 2012], and elevators [Wang et al. 2012b]. A common limitation is that these landmark types are not present in all indoor environments or are only infrequently observed in daily life. An alternative for tracking of people at home is ActionSLAM [Hardegger et al. 2012], which uses frequently performed activities (sitting, standing still, stair climbing) as landmark observations, assuming that they predominantly occur at specific locations. By fusing these observations with open-loop tracking from a foot-mounted sensor, ActionSLAM can track a person during hour-long walks in multifloor buildings [Hardegger et al. 2015]. Figure 2 depicts some state-of-the-art SLAM formulations as Dynamic Bayesian Networks (DBNs) and relates them to the S-SMART framework we propose. Table I lists the most important variable and parameter names that we use throughout this article.

Table I. List of Variables and Parameters

Variable Name	Symbol	Variable Name	Symbol	Variable Name	Symbol
User step	u	$\rightarrow$ step length	l	User activity	A
$\hookrightarrow$ measured step	$\hat{u}$	$\hookrightarrow$ heading change	$\delta\phi$	$\hookrightarrow$ measured motion	$\hat{M}$
Semantic map	Θ			**Parameter Name**	**Symbol**
$\hookrightarrow$ object	θ	$\rightarrow$ object position	χ	Number of particles	N_p
$\hookrightarrow$ object index	n	$\hookrightarrow$ position covariance	Σ	Object insertion probability	p_0
$\hookrightarrow$ number of objects	N_θ	$\hookrightarrow$ object interaction	α_t	Minimal observation probability	p_1
User pose	s	$\rightarrow$ user position	x	State transition probability	c_0
$\hookrightarrow$ start pose	s_0	$\hookrightarrow$ user heading	ϕ	Map maintenance threshold	V_0

We use the same notations as in Thrun et al. [2004] wherever possible. We apply superscript notations to indicate sequences, e.g., $s^t = \{s_1, \ldots, s_t\}$ is the sequence of user poses up to a time t, also referred to as the user's *path*.

2.3. Environment Representations in Localization and Activity Recognition

The maps in SLAM are models of how the world around the person looks. In the simplest case, this model is a list of anonymous landmarks with fixed location, for example, high-contrast features in camera images. Adding semantic information to each landmark, for example, a type attribute, can enhance the map, reduce the risk of landmark confusions, and speed up the convergence of SLAM [Meyer et al. 2011]. For example, if a map contains two landmark types (A and B), and a future observation is of a type A landmark, it cannot confuse the observation with any of the type B landmarks. In pedestrian tracking, for example, Grzonka et al. [2012] distinguish landmarks of different types (stairs and doors). More common are semantic maps in robot localization, where they can, for example, help a mobile agent to find semantically important places (e.g., the fridge that stores the medication of a patient [Nüchter and Hertzberg 2008]). However, all these works assume that the real-life equivalent of the internally estimated semantic map is static and does not change over time. Also, these systems do not consider that type observations may be erroneous. For example, even if the landmark observation system recognizes type A, there is a chance that the landmark was actually of type B.

As a further level of detail, some robotic SLAM systems assign state attributes to their internal environment representation, thus being able to account for map modifications due to actions by the robot or other influences. For example, Wang et al. [2007] apply Bayesian tracking to predict the position of moving objects (in that case cars) in their map. Other works in robotics allow the robot itself to interact with objects. For example, Magnenat et al. [2012] present a mobile agent that autonomously collects resources in the environment and uses them to construct towers and bridges, thus manipulating its surroundings. By combining semantic maps with ontologies, a robot can determine which interactions are possible with close-by objects, and how they will affect the environment [Galindo et al. 2008].

All these algorithms are not immediately applicable to person tracking and activity recognition, because in contrast to robotics, the decision making and task execution of people are not directly observable. Environment and state representations in person tracking therefore have to account for possible recognition errors by means of probabilistic models. LocAFusion [Hardegger et al. 2014] builds such a probabilistic model in an unsupervised manner by accumulating all activity observations for a location over time and then extracting the maximum likelihood estimate about what object is at this place at the end of a recording. It then uses the created semantic map to postcorrect all previous observations such that they are in line with the object type estimation. LocAFusion is limited in its design as a postprocessing algorithm, and it does not take state modifications into account.

Only very few works in context awareness considered that interactions with the environment affect the conditional probability for a user's future behavior. Patterson et al. [2005] consider Hidden Markov Models (HMMs) to describe activity sequences, which leads to the robust recognition of breakfast activities. However, the state representation in this case does not reflect the state of the environment, but of the user him- or herself with respect to his or her typical, prelearned activity routine. More complex state estimation methods are necessary to disentangle the overall activity sequence into an estimation of the environment state. In particular, the system must know which object the user actually manipulates with each activity. For example, opening a window can be followed by closing it again or by opening another window. Only if the identity of the manipulated window is known can the previous action actually influence the upcoming recognition in this case. Van Kasteren et al. [2008] and Rashidi and Cook [2013] achieved this disentanglement by predeploying sensors in the environment and then learning interaction patterns that correspond to higher-level activities such as *taking medication*. Similarly, Chen et al. [2012] apply ontologies and semantic reasoning to recognize a person's activities in a smart home from the interactions with instrumented objects. These approaches require external infrastructure and they typically depend on domain knowledge. To the best of our knowledge, there is no system available yet that tracks a complex environment state representation from wearable-only sensor setups, and without supervised training of the location-to-activity mapping.

3. THE S-SMART ALGORITHM

3.1. Overview

S-SMART addresses the limitations of the state of the art outlined previously through fully Bayesian fusion of error-prone activity recognitions and open-loop tracking. The proposed algorithm iteratively learns how the person's environment looks by building and updating a probabilistic dynamic semantic map. It then uses this model to constrain the person's position within the environment, and as a prior estimate on the activities he or her may perform. We designed S-SMART with fully stand-alone, wearable sensor setups in mind. The algorithm learns the location-to-activity mapping in an unsupervised process, which enables fast deployment to novel environments. Figure 3 depicts the main steps of S-SMART graphically for an example scenario.

3.2. Problem Statement

The S-SMART problem consists of calculating an estimate of the person's *pose* $s_t = \{x_t, \phi_t\}$ (*position* x_t and *heading* ϕ_t) and his or her location-related *activities* A_t in a *dynamic semantic map* Θ_t, given data measured with wearable sensors. See Table I for a list of variable names. The S-SMART task is closely related to the robotic SLAM problem of estimating an agent's pose s_t within a map Θ_t. However, other than in robotics, there is direct access neither to the sensory information on which the central nervous system bases its decision making nor to the intentions that the user's motor system tries to realize. As Figure 2 depicts, the result of a user's intentions in the S-SMART problem is *steps* $u_t = \{l_t, \delta\phi_t\}$ (each step has a *length* l_t and changes the user's *heading* by $\delta\phi_t$) and activities A_t that affect the S-SMART state. Open-loop tracking algorithms can process the data of the body-worn motion sensors into observations $\hat{u}^t = \{\hat{u}_1, \ldots, \hat{u}_t\}$ of the steps u^t. Inertial sensors can also measure *body motions* $\hat{M}^t = \{\hat{M}_1, \ldots, \hat{M}_t\}$ that coincide with activities $A^t = \{A_1, \ldots, A_t\}$. In this work, $\hat{M}_t$ are the triaxial accelerometer and gyroscope measurements associated to a stance phase t. The mapping from $\hat{M}_t$ to A_t is usually not unique, but it can be described with a probabilistic model $p(A_t|\hat{M}_t)$. This is because multiple actions may produce similar measurements $\hat{M}_t$ in the wearable sensor setup.

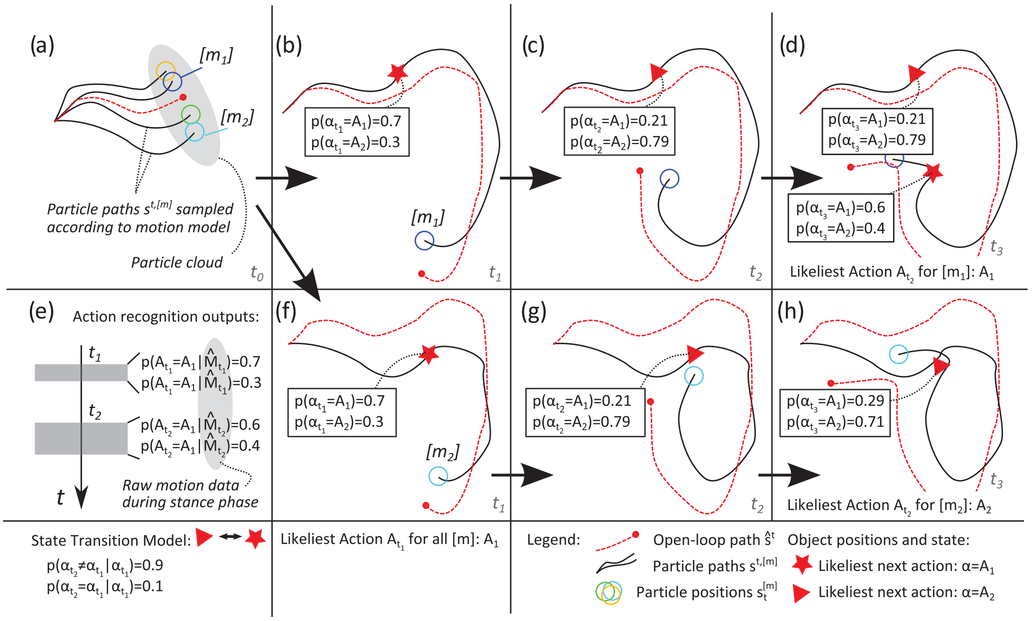

Fig. 3. This figure demonstrates the main steps of S-SMART. First of all, the algorithm spreads particle paths $s^{t,[m]}$ according to the motion model (see figure part (a) and Section 3.4). When action recognitions occur (see part (e)), the algorithm adds corresponding object landmarks to each particle's internal map $\Theta^{[m]}$. For all objects, the exact type and state are unknown. However, probabilities $p(\alpha_t)$ describe the likelihood of upcoming interaction types with this object, given the previous observations. Figure parts (b)–(d) show the map construction for particle $[m_1]$ in response to observation updates as explained in Section 3.5. For this particle, the second action recognition A_{t_2} occurs far from the first observation (i.e., the distance between $s_{t_1}^{[m_1]}$ and $s_{t_2}^{[m_1]}$ is large and thus, the position factor in Equation (6) small), and S-SMART decides to insert a new landmark to the map after the recognition at stance phase t_2. Figure parts (f)–(h) show the mapping with particle $[m_2]$, for which the second action recognition occurs close to the first action. Thus, the object position factor is close to 1. The action type factor in this case is $p(A_{t_2} = A_1|\hat{M}_{t_2}) * p(\alpha_{t_2} = A_1) + p(A_{t_2} = A_2|\hat{M}_{t_2}) * p(\alpha_{t_2} = A_2) = 0.44$. For a small object insertion probability, for example, $p_0 = 0.1$, this particle is likely to decide that the action A_{t_2} is a second interaction with the object first seen at t_1. In this case, it is likelier that the user performed the activity A_2 given the previous observation of A_1 at this place and the state transition model, which says that after executing A_1, the next interaction with this object is usually A_2. As a result, particle $[m_2]$ ends up with a different semantic map to $[m_1]$. Also, this particle supports a different hypothesis on the sequence of performed actions compared to the location-free action recognition system. After completing the depicted steps, S-SMART recalculates the weights of each particle, potentially resamples the particle cloud, and finally decides on the best particle $[\bar{m}]$.

The dynamic semantic map $\Theta_t = \{\theta_{n,t}\}_{n=1,\ldots,N_{\theta,t}}$ in S-SMART represents the environment by linking locations and actions to so-called *objects*. We denote these objects as $\theta_{n,t} = \{\alpha_{n,t}, \chi_{n,t}\}$, where $\alpha_{n,t}$ are the *interactions* possible with the object at time t. The prior probability for an interaction $\alpha_{n,t}$ depends on the underlying state of the object. χ_n is the object's *location* in a 3D coordinate system. For simplicity, we assume that all objects are space fixed, that is, that the object's true position χ_n does not change with time. The *object index* $n = 1 \ldots, N_{\theta,t}$ is a unique number that distinguishes the objects in the map Θ_t. The *object association* n_t associates the action A_t to a specific object with index $n = n_t$ in the map Θ_t. Therefore, it indicates that the user's action A_t modifies the object $\theta_{n=n_t,t}$ (assuming users can only modify one object at a time).

In conclusion, the goal of S-SMART is to take a maximum likelihood estimation on the user's pose $\bar{s}_t$, action $\bar{A}_t$, and map $\bar{\Theta}_t$ from $p(s_t, A_t, \Theta_t|\hat{u}^t, \hat{M}^t, n^t, s_0)$. Here, s_0 is the initial pose, $\hat{u}^t$ and $\hat{M}^t$ are the preprocessed input measurements from the wearable sensors, and n^t are the object associations. The remainder of this section describes a solution to the S-SMART problem that extends the FastSLAM algorithm by Montemerlo et al. [2002].

3.3. Particle Filter Formulation

Particle filters provide an intuitive solution to the S-SMART problem, accounting for the nonlinear and non-Gaussian error characteristics of the open-loop tracking input $\hat{u}^t$ and the activity observations $p(A_t|\hat{M}_t)$. As in Montemerlo et al. [2002], S-SMART estimates the probability distribution for the full path s^t, rather than for the pose s_t of the person at time t, which enables the following factorization:

$$p(s^t, A_t, \Theta_t|\hat{u}^t, \hat{M}^t, n^t, s_0) = \underbrace{p(s^t|\hat{u}^t, s_0)}_{\text{Motion Model}} \cdot \underbrace{\prod_{n=1}^{N_{\theta,t}} p(A_t, \theta_{n,t}|s^t, \hat{M}^t, n^t)}_{\text{Observation Model}}. \tag{1}$$

Equation (1) splits the posterior into an estimation of the person's path s^t, and of the corresponding map Θ_t and activity A_t, conditioned on the path s^t. S-SMART implements the motion model as a particle filter, sampling paths from $s^{t,[m]} \sim p(s^t|\hat{u}^t, s_0)$, with $[m]$ as *particle index*. For the observation model $p(A_t, \theta_{n,t}|s^{t,[m]}, \hat{M}^t, \hat{n}^{t,[m]})$, S-SMART uses a closed-form representation that Section 3.5 explains in detail. The combination of a particle filter with other filtering techniques is known as Rao-Blackwellization [Liu and Chen 1998]. Practically, this means that S-SMART creates a set of hypotheses on the path the person takes and then builds a different semantic map for each of these paths based on where action observations $p(A_t|\hat{M}_t)$ take place along the path $s^{t,[m]}$.

As the object association n_t in Equation (1) is typically unknown, it needs to be estimated given all available information about the environment state and the observations at the time of stance phase t. As proposed in Thrun et al. [2004], we account for the unknown object association by sampling $\hat{n}_t^{[m]}$ for each particle $[m]$ according to

$$\hat{n}_t^{[m]} \sim p(A_t, \theta_{n_t,t}|s^{t,[m]}, \hat{M}^t, n_t, \hat{n}^{t-1,[m]}). \tag{2}$$

The implementation of Equation (2) is analogous to the derivation of the observation model in Equation (1), which Section 3.5 presents in detail. Figure 4 outlines the data flow in the update calculation of a particle for an example implementation.

3.4. Motion Model

Assuming that the pose s_t depends only on the previous pose s_{t-1} and the performed step u_t, S-SMART can approximate $p(s^t|\hat{u}^t, s_0)$ by sampling particles after each step from

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

The *motion model* implements this update function, reflecting the error distribution in the open-loop tracking estimates as depicted in Figure 3(a). For other location measurements, for example, GPS scans, alternative pose estimation functions with dedicated error models must be implemented.

3.5. Observation Model

Depending on the sampled object association $\hat{n}_t^{[m]}$ of particle $[m]$, S-SMART distinguishes two observation update equations. Objects $\theta_{n,t}$ with $n \neq \hat{n}_t^{[m]},\ n \in \{1, \dots, N_{\theta,t}\}$ are objects for which the particle $[m]$ assumes that the person did not modify them.

Therefore, the posterior for these objects remains unchanged (for readability, we leave away the particle identification $[m]$ henceforth):

$$p(A_t, \theta_{n,t}|s^t, \hat{M}^t, \hat{n}^t) = p(A_{t-1}, \theta_{n,t-1}|s^{t-1}, \hat{M}^{t-1}, \hat{n}^{t-1}). \tag{4}$$

On the other hand, for the object with which the user interacts according to the particle's belief, that is, for the object with index $n = \hat{n}_t^{[m]}$, the following factorization applies:

$$\begin{aligned} p(A_t, \theta_{n,t}|\ s^t,\ \hat{M}^t, \hat{n}^t) &= p(A_t, \alpha_{\hat{n}_t,t}, \chi_{\hat{n}_t,t}|s^t, \hat{M}^t, \hat{n}^t) & (5)\\ &= \underbrace{p(A_t, \alpha_{\hat{n}_t,t}|\hat{M}^t, \hat{n}^t)\cdot}_{\text{Action Type Factor } w_{\hat{n}}^{ac}}\ \underbrace{p(\chi_{\hat{n}_t,t}|s^t, \hat{n}^t)}_{\text{Position Factor } w_{\hat{n}}^{pos}}\ . & (6)\end{aligned}$$

In the step from Equations (5) to (6), we exploit the conditional independence between the interaction $\alpha_{\hat{n}_t,t}$ with the object $\theta_{\hat{n}_t,t}$ and its location $\chi_{\hat{n}_t,t}$. In conclusion, the probability of an action A_t being linked to a specific object $\theta_{\hat{n}_t,t}$ is the product of an activity-based factor and a position factor.

The *action type factor* $w_{\hat{n}}^{ac} = p(A_t, \alpha_{\hat{n}_t,t}|\hat{M}^t, \hat{n}^t)$ describes the likelihood of performing an interaction A_t with the object, given all previously observed interactions with the same object. As actions can only correspond to an interaction with an object if the object's state allows for it, $p(A_t, \alpha_{\hat{n}_t,t}|\hat{M}^t, \hat{n}^t) = 0$ if $A_t \neq \alpha_{\hat{n}_t,t}$. Otherwise, the following equations apply:

$$\begin{aligned} p(\alpha_{\hat{n}_t,t} &= A_t|\hat{M}^t, \hat{n}^t) = \sum_{\alpha_{\hat{n}_t,t-1}} p(\alpha_{\hat{n}_t,t}, \alpha_{\hat{n}_t,t-1})|\hat{M}^t, \hat{n}^t) & (7)\\ &= \sum_{\alpha_{\hat{n}_t,t-1}} p(\alpha_{\hat{n}_t,t}|\alpha_{\hat{n}_t,t-1}, \hat{M}^t, \hat{n}^t) \cdot p(\alpha_{\hat{n}_t,t-1}|\hat{M}^t, \hat{n}^t) & (8)\\ &= \eta_A \cdot p(A_t|\hat{M}_t) \cdot \sum_{\alpha_{\hat{n}_t,t-1}} p(\alpha_{\hat{n}_t,t}|\alpha_{\hat{n}_t,t-1}) \cdot p(\alpha_{\hat{n}_t,t-1}|\hat{M}^{t-1}, \hat{n}^{t-1}). & (9)\end{aligned}$$

In the step from Equations (7) to (8), we apply the chain rule. From Equations (8) to (9), we use Bayes' Theorem and the conditional independences of the variables. η_A is a normalization factor. With Equation (9), the algorithm can iteratively estimate $p(A_t, \alpha_{\hat{n}_t,t}|\hat{M}^t, \hat{n}^t)$ by forward probability propagation. The *action type factor* $w_{\hat{n}}^{ac}$ equals the cosine similarity between $p(A_t|\hat{M}_t)$ and $p(\alpha_{\hat{n}_t,t-1}|\hat{M}^{t-1}, \hat{n}^{t-1})$ (see the example in Figure 3):

$$w_{\hat{n}}^{ac} = p(A_t, \alpha_{\hat{n}_t,t}|\hat{M}^t, \hat{n}^t) = \eta_A \cdot \sum_{\alpha_{\hat{n}_t,t}} p(A_t = \alpha_{\hat{n}_t,t}|\hat{M}_t) p(\alpha_{\hat{n}_t,t}|\hat{M}^t, \hat{n}^t). \tag{10}$$

The object position factor $w_{\hat{n}}^{pos}$ measures the probability of being at the location of the object $\theta_{\hat{n}_t}$, given the particle's pose $s_t^{[m]}$. The calculation of the *position factor* $w_{\hat{n}}^{pos}$ is analogous to FastSLAM [Thrun et al. 2004], using Bayes' Theorem and introducing the normalization factor η_P:

$$w_{\hat{n}}^{pos} = p(\chi_{\hat{n}_t,t}|s^t, \hat{n}^t) = \eta_P \cdot p(s_t|s^{t-1}, \chi_{\hat{n}_t,t-1}, \hat{n}^t) \cdot p(\chi_{\hat{n}_t,t}|s^{t-1}, \hat{n}^{t-1}). \tag{11}$$

The actual implementation of the factor $p(s_t|s^{t-1}, \chi_{\hat{n}_t,t-1}, \hat{n}^t)$ depends on the position observation error model. We present an example implementation in Section 4.

In summary, any S-SMART observation update implementation requires two models: (1) the *state transition model* $p(\alpha_{\hat{n}_t,t}|\alpha_{\hat{n}_t,t-1})$ that describes the probability of state changes in response to interactions with the object, and (2) the *position update model*

$p(\chi_{\hat{n}_t,t}|s^t, \hat{n}^t)$, which reflects the particle's estimate of the object positions, given the newest observation.

3.6. Weight Update, Resampling, and Map Maintenance

Each particle tracks a path variation according to the motion model, while associated weights $w_t^{[m]}$ represent the likelihood of this path according to the observation model. Every observation update changes the particle weight as follows:

$$w_t^{[m]} = \eta \cdot w_{\hat{n}_t}^{[m]} = \eta \cdot w_{\hat{n}_t}^{ac,[m]} \cdot w_{\hat{n}_t}^{pos,[m]}. \tag{12}$$

S-SMART sets the probability of inserting a new object in the map, that is, of $\hat{n}_t = N_{\theta,t-1} + 1$, to p_0, and thus $w_{\hat{n}_t} = \eta \cdot p_0$. The factor η normalizes the particle weights after every update such that they sum up to $\sum_{[m]=1}^{N_p} w_t^{[m]} = 1$. The current implementation regularly applies systematic resampling [Douc and Cappé 2005] to replace particles that have very low weights, thus avoiding degeneration of the particle cloud with time.

A further necessary tool to limit the deterioration of the state approximation in S-SMART is *map maintenance*, which handles object insertions in the semantic map that are due to false-positive activity recognitions. To identify such falsely inserted objects, S-SMART counts every visit in the proximity of an object and registers whether an interaction with the object takes place. If the ratio of visits with object interaction versus visits with no interaction is larger than the *map maintenance threshold* V_0, the object is removed from the active mapping process. Every particle performs map maintenance for its own semantic map.

3.7. State Sequence Estimation

The set of weights $w_t^{[m]}$, paths $s^{t,[m]}$, maps $\Theta_t^{[m]}$, and actions $A_t^{[m]}$ approximates the S-SMART probability distribution $p(s^t, A_t, \Theta_t|\hat{u}^t, \hat{M}^t, n^t, s_0)$. The next step is to draw an optimal estimate $\{\bar{s}^t, \bar{A}_t, \bar{\Theta}_t\}$ of the actual state from this set of particles. For this purpose, S-SMART finds the best particle $[\bar{m}]$ according to the following experimentally derived rules: First, the algorithm selects all particles with the minimum number of objects in their map. In the example in Figure 3, the particle $[m_2]$ inserts only a single object and would therefore be selected above $[m_1]$. Next, S-SMART counts the number of objects per object type and discards all particles that have a different distribution of object types to the majority. Among the remaining particles, $[\bar{m}]$ is the particle with highest weight $w_t^{[m]}$. The S-SMART estimates of the person's path, his or her activities, and the semantic map then correspond to the path, map, and activity of this *best particle* $[\bar{m}]$.

Finally, S-SMART postcorrects earlier activity observations $A_\tau|_{\tau=1,\dots,t}$ and the state sequence Θ^t of the best particle's semantic map given the object interaction sequences. For this purpose, the framework takes all previous observations with $\hat{n}_\tau = \hat{n}_t, \tau = 1, \dots, t$ and applies the Viterbi algorithm [Forney Jr 1973] to the observation probabilities $p(A_\tau|\hat{M}_\tau)$, using the state transition model $p(\alpha_{\hat{n}_t,\tau}|\alpha_{\hat{n}_t,\tau-1})$. During this process, S-SMART filters out actions with $p(A_\tau|\hat{M}_\tau) < p_1$, that is, actions for which the best guess of the Viterbi algorithm has a probability that is smaller than the minimal observation probability threshold. In that case, S-SMART assumes that this was an incorrect observation and that no action (and therefore also no state transition) occurred at time τ. The outputs of this postcorrection step are the likeliest activity sequence $\bar{A}^t$ and the semantic map as a function of time $\bar{\Theta}^t$. The sequence of object interactions also defines the states of all objects at each time. Algorithm 1 describes all the main steps of S-SMART.

ALGORITHM 1: S-SMART

```
Data: $p(A_t|\hat{M}_t)|_{t=1,\dots,T}$ , $\hat{u}^T$ , $s_0$
Result: $\bar{s}^T$ , $\bar{\Theta}_T$ , $\bar{A}_T$
$\{s_0^{[m]}, \Theta_0^{[m]}, w_0^{[m]}\} \Leftarrow$ initializeParticles($N_p$) ;                    // Initialization
for $t = 1, \dots, T$ do
    for $[m] = 1, \dots, N_p$ do
        $s_t^{[m]} = \{x_t^{[m]}, \phi_t^{[m]}\} \sim p(s_t^{[m]}|s_{t-1}^{[m]}, \hat{u}_t)$ ;                    // Motion Update
        if $\hat{M}_t \neq \emptyset$ then
            for $n = 1, \dots, N_{\theta,t-1}^{[m]}$ do                                      // Observation Update
                $w_n^{ac} = p(A_t, \alpha_{n_t,t}^{[m]}|\hat{M}^t, \hat{n}^{t-1,[m]}, n_t)$ ;                   // Action Type Factor
                $w_n^{pos} = p(\chi_{n_t,t-1}^{[m]}|s^t, \hat{n}^{t-1,[m]}, n_t)$ ;                        // Position Factor
            end
            $w_{N_{\theta,t-1}+1} = p_0$ ;                                      // Object Insertion Probability
            $\hat{n} \sim w_n = w_n^{pos} \cdot w_n^{ac}$ ;                            // Sampling of Object Association
            $N_{\theta,t}^{[m]} = \max(N_{\theta,t-1}^{[m]}, \hat{n})$ ;                     // Update Number of Objects in Map
            if $\hat{n} = N_{\theta,t-1} + 1$ then                                     // Initialize New Object
                $p(A_t, \alpha_{\hat{n}_t,t}|\hat{M}^t, \hat{n}^t) = p(A_t|\hat{M}_t)$;                      // Initial Action Type
                $p(\chi_{\hat{n}_t,t}^{[m]}|s^{t,[m]}, \hat{n}^t) = p(x_t|s_t^{[m]})$;                    // Initial Object Position
            else if $\hat{n} < N_{\theta,t-1} + 1$ then                               // Update Associated Object
                $p(A_t, \alpha_{\hat{n}_t,t}|\hat{M}^t, \hat{n}^t) = p(A_t|\hat{M}_t) \cdot \sum_{\alpha_{\hat{n}_t,t-1}} p(\alpha_{\hat{n}_t,t}|\alpha_{\hat{n}_t,t-1}) \cdot p(\alpha_{\hat{n}_t,t-1}|\hat{M}^{t-1}, \hat{n}^{t-1})$
                $p(\chi_{\hat{n}_t,t}^{[m]}|s^{t,[m]}, \hat{n}^t) = p(s_t^{[m]}|s^{t-1,[m]}, \chi_{\hat{n}_t,t-1}^{[m]}, \hat{n}^t) \cdot p(\chi_{\hat{n}_t,t-1}^{[m]}|s^{t-1,[m]}, \hat{n}^{t-1})$
            end
            $w_t^{[m]} = \eta \cdot w_{\hat{n}}$ ;                                 // Normalize Particle Weights
            $\{A^{t,[m]}, \Theta_t^{[m]}\} \Leftarrow$ mapMaintenance($s_t^{[m]}, A^{t,[m]}, \Theta_t^{[m]}$);      // Removal of False Objects
        end
    end
    $\{s^{t,[m]}, A_t^{[m]}, \Theta_t^{[m]}, w_t^{[m]}\} \Leftarrow$ resample($\{s^{t,[m]}, A_t^{[m]}, \Theta_t^{[m]}, w_t^{[m]}\}$) ;            // Resampling
    $[\bar{m}] \Leftarrow$ findBestParticle($\{\Theta_t^{[m]}, w^{[m]}\}$) ;                             // State Estimation
    $\bar{s}^t \Leftarrow s^{t,[\bar{m}]}$, $\{\bar{A}^t, \bar{\Theta}^t\} \Leftarrow$ Viterbi($\hat{n}^{t,[m]}, \{ p(A_\tau|\hat{M}_\tau\Big|_{\tau=1,\dots,t}\}), \Theta_t^{[\bar{m}]}$) ;        // Assign Results
end
```

4. SCENARIO-SPECIFIC IMPLEMENTATION

4.1. System Specification

The generic S-SMART particle-filter framework can be adapted to a specific scenario by implementing the appropriate *motion model* $p(s_t|s_{t-1}^{[m]}, \hat{u}_t)$, the *state transition model* $p(\alpha_{\hat{n}_t,t}|\alpha_{\hat{n}_t,t-1})$, and the *position observation model* $p(\chi_{\hat{n}_t,t}|s^t, \hat{n}^t)$. These models depend on the choice of sensors and algorithms for providing the open-loop tracking data $\hat{u}^t$ and the action recognition input $p(A_t|\hat{M}_t)$. In this section, we demonstrate the practical implementation of S-SMART for an at-home monitoring scenario: the fusion of tracking data from a foot-mounted IMU, hand action recognitions from a wrist-mounted IMU, and basic activity recognition from a hip-worn smartphone. The result is a fully wearable, stand-alone monitoring setup that can be deployed efficiently. We hereby focus on interactions with space-fixed objects, such as the opening of a door or the turning on of a water tap. We denote these actions by an *object type* (e.g., D for door) and an *interaction type* (e.g., OPEN for opening). To distinguish between multiple doors, we add an identifier to the object (e.g., D2_OPEN stands for opening door 2). We call this an *object-specific* action notation, as opposed to the *unspecific* notation (e.g., D_OPEN, which

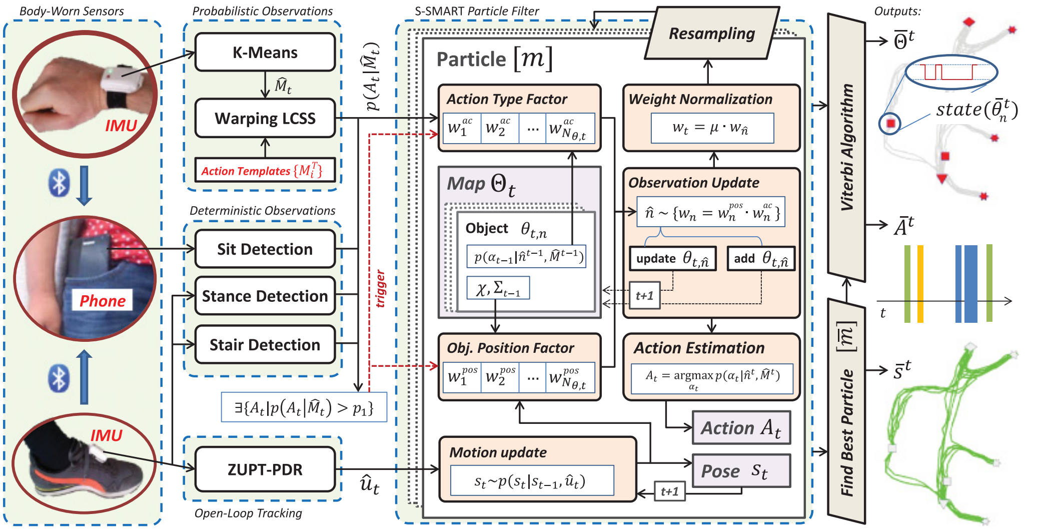
Body-Worn Sensors
IMU
Phone
IMU
Probabilistic Observations
K-Means
$\hat{M}_t$
Warping LCSS
Action Templates $\{M_i^T\}$
Deterministic Observations
Sit Detection
Stance Detection
Stair Detection
$\exists\{A_t|p(A_t|\hat{M}_t) > p_1\}$
ZUPT-PDR
Open-Loop Tracking
$p(A_t|\hat{M}_t)$
trigger
$\hat{u}_t$
S-SMART Particle Filter
Resampling
Particle $[m]$
Action Type Factor
w_1^{ac} w_2^{ac} … $w_{N_{\theta,t}}^{ac}$
Weight Normalization
$w_t = \mu \cdot w_{\hat{n}}$
Map Θ_t
Object $\theta_{t,n}$
$p(\alpha_{t-1}|\hat{n}^{t-1}, \hat{M}^{t-1})$
χ, Σ_{t-1}
Observation Update
$\hat{n} \sim \{w_n = w_n^{pos} \cdot w_n^{ac}\}$
update $\theta_{t,\hat{n}}$
add $\theta_{t,\hat{n}}$
t+1
Obj. Position Factor
w_1^{pos} w_2^{pos} … $w_{N_{\theta,t}}^{pos}$
Action Estimation
$A_t = \operatorname{argmax}_{\alpha_t} p(\alpha_t|\hat{n}^t, \hat{M}^t)$
Action A_t
Motion update
$s_t \sim p(s_t|s_{t-1}, \hat{u}_t)$
Pose s_t
t+1
Viterbi Algorithm
Find Best Particle $[\bar{m}]$
Outputs:
$\bar{\Theta}^t$
$state(\bar{\theta}_n^t)$
$\bar{A}^t$
t
$\bar{s}^t$

Fig. 4. Outline of the S-SMART implementation for fusion of hand action recognition and open-loop pedestrian tracking from a foot-mounted IMU. ZUPT-PDR detects a person's steps $\hat{u}_t$ from a foot-mounted IMU and triggers motion updates, while a template matching algorithm estimates action probabilities $p(A_t|\hat{M}_t)$ and triggers observation updates. A hip-worn smartphone detects whether a person sits or stands. The S-SMART particle filter then fuses these inputs to a dynamic map representation of the environment and returns the likeliest path and activity sequence of the user.

is the action of opening any door). Figure 4 depicts an overview of the proposed system architecture.

4.2. Open-Loop Tracking

For estimating open-loop location data $\hat{u}^t$, we apply the Zero-Velocity-Update Pedestrian Dead Reckoning (ZUPT-PDR) algorithm introduced by Foxlin [2005]. The output of ZUPT-PDR is a continuous track of the foot in 3D space. Stance detections cut this track into step observations $\hat{u}_t = \{\hat{l}_t, \delta\hat{\phi}_t\}$, each of them triggering an S-SMART motion update. To reduce heading drifts during long no-movement phases, our implementation applies measurements of the local magnetic field direction as proposed in Hardegger et al. [2015]. This drift correction uses the change in the magnetometer-based heading estimation with respect to the beginning of the stance phase as an additional observation in the extended Kalman filter of ZUPT-PDR. It only gets activated during long stance phases (at least 3s), as the drifts are negligible for shorter periods.

4.3. Wearable Action Recognition

For the set of hand actions that we investigate in this work, the template-matching algorithm Longest-Common-SubSequence (LCSS) [Vlachos et al. 2003] matching outperforms alternative approaches in reference datasets [Nguyen-Dinh et al. 2012]. In our implementation, we apply a variant of LCSS known as Warping LCSS [Roggen et al. 2015], which is computationally more efficient than standard LCSS while attaining similar recognition accuracy. Rather than applying LCSS matching in the 6-dimensional raw signal space spanned by triaxial gyroscope and accelerometer inputs, S-SMART first clusters the raw data to a one-dimensional symbol sequence (*"string"*) by applying the K-means algorithm.

For template matching, Warping LCSS requires a prerecorded training set with template strings $\{M_A^T\}$ for each action of interest. S-SMART compares these templates

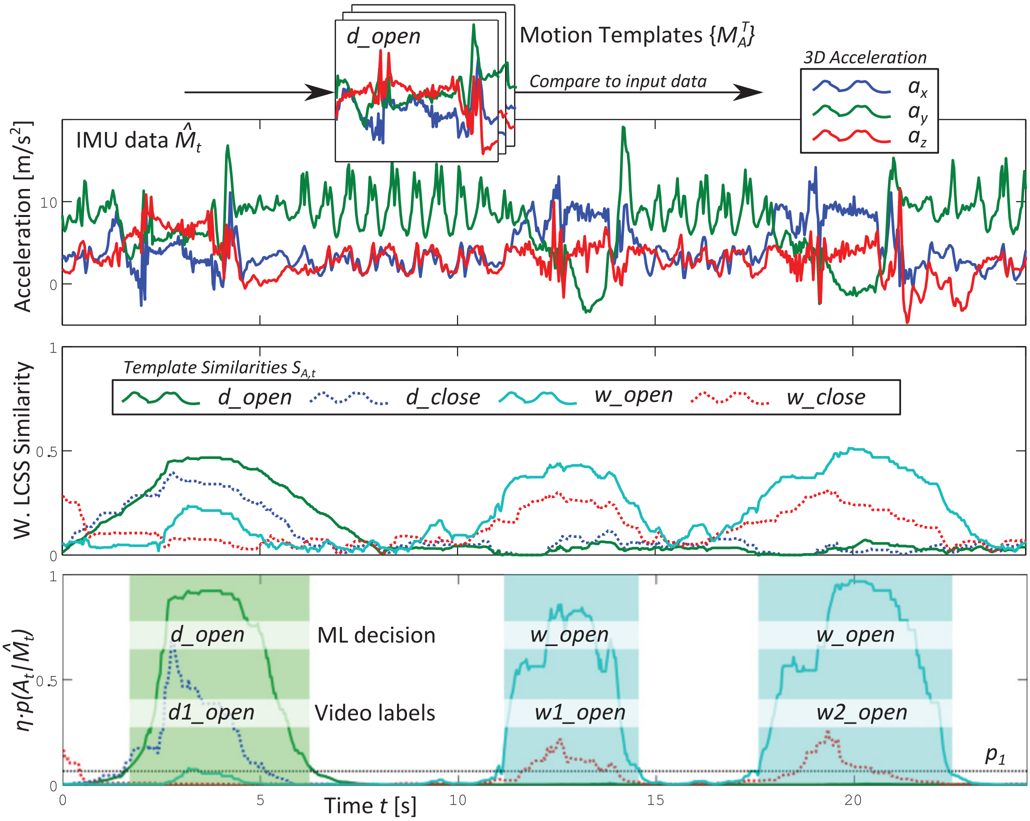

Fig. 5. Extract of an experimental recording (W1, see Table II) where the user first opens a door, walks to another location to open a window, and finally opens a second window elsewhere. Warping LCSS calculates the similarity scores $S_{A,t}$ by comparing pretrained templates (D_OPEN) with the input IMU data. Next, a sigmoid function maps the similarities to observation probabilities $p(A_t|\hat{M}_t)$, triggering an observation update if the probability for at least one action is above the minimal observation probability p_1.

to the input signal $\hat{M}_t$ of the wrist-worn IMU and calculates the LCSS similarity scores $S_{A,t}(\hat{M}_t, M_A^T)$. These similarities can be mapped to the observation probabilities $p(A_t|\hat{M}_t) = f(S_{A,t})$ through a sigmoid function, as previously proposed for linking visual-object-type recognitions to observation probabilities [Meyer et al. 2011]. When $p(A_t|\hat{M}_t)$ is smaller than the minimal observation probability p_1 for all actions in the training set, S-SMART assumes that no action took place and consequently does not trigger an observation update. However, if $p(A_t|\hat{M}_t) \geq p_1$ for at least one action A_t, S-SMART triggers an observation update, normalizing the individual probabilities for each action to sum up to 1. Figure 5 outlines the processing steps from the raw motion data to observation probabilities $p(A_t|\hat{M}_t)$.

In addition, our setup recognizes the basic activities *sitting*, *standing still*, and *stair climbing* from the foot-mounted IMU and the orientation of the hip-worn smartphone. Given that the confidence in these observations is high, S-SMART fixes $p(A_t|\hat{M}_t)$ to 1 for these actions. The activity recognition evaluation in the next section does not include these basic activities in the performance measures.

4.4. S-SMART Model Implementations

- **Motion Model** $p(s_t|s_{t-1}^{[m]}, \hat{u}_t)$: Previous research proposed various error models for ZUPT-PDR (e.g., Robertson et al. [2013]). In this work, we use the motion model from Hardegger et al. [2015], which accounts for the exponential error growth as a function of the time between stance observations in ZUPT-PDR.
- **State Transition Model** $p(\alpha_{\hat{n}_t,t}|\alpha_{\hat{n}_t,t-1})$: S-SMART models the state transition for an object as a stochastic process. In this way, it considers that state transition observations may have been erroneous or skipped. For each object with two or more possible states, the probability of a state change between two object interactions is modeled to be equal to the state transition probability c_0, and the probability of not changing the state is $1-c_0$. This, for example, accounts for the possibility that another person might close a window that the user previously opened.
- **Position Observation Model** $p(\chi_{\hat{n}_t,t}|s^t, \hat{n}^t)$: For updating an object's position estimate in response to an observation, S-SMART takes the same approach as ActionSLAM [Hardegger et al. 2015]. At time $t-1$, every particle in S-SMART represents the position of the observed object $p(\chi_{\hat{n}_t,t-1}|s^{t-1}, \hat{n}^{t-1})$ as a Gaussian with mean $\chi_{\hat{n},t-1}$ and covariance $\Sigma_{\hat{n},t-1}$. We approximate objects to have a disc shape with radius r_0, as previously proposed in Hardegger et al. [2015]. The new observation of the object position is equal to the person's position x_t and obstructed by Gaussian noise with *measurement covariance* R_t:

$$p(s_t|s^{t-1}, \chi_{\hat{n}_t,t-1}, \hat{n}^t) = \max(0, x_t - \chi_{\hat{n}_t,t-1} - r_0) + R_t. \tag{13}$$

  Given this Gaussian observation, the posterior of the object position can be calculated with a Kalman filter:

$$K_t = \Sigma_{\hat{n}_t,t-1}(\Sigma_{\hat{n}_t,t-1} + R_t)^{-1} \tag{14}$$

$$\chi_{\hat{n}_t,t} = \chi_{\hat{n}_t,t-1} + K_t(\max(0, x_t - \chi_{\hat{n}_t,t-1} - r_0))^T \tag{15}$$

$$\Sigma_{\hat{n}_t,t} = (I - K_t)\Sigma_{\hat{n}_t,t-1}. \tag{16}$$

  The corresponding position update factor $w_{\hat{n}}^{pos}$ is the result of a convolution between two Gaussians (see Thrun et al. [2004] for derivation):

$$w_{\hat{n}}^{pos} = |2\pi(\Sigma_{\hat{n}_t,t-1} + R_t)|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(\hat{z}_n^T(\Sigma_{\hat{n}_t,t-1} + R_t)^{-1}\hat{z}_n\right). \tag{17}$$

5. EVALUATION

5.1. Dataset

For evaluation, we recorded two datasets with people wearing the S-SMART sensor setup while interacting with their environment. In all the data collections, people wore a foot-mounted EXLs3[1] IMU for ZUPT-PDR, a second EXLs3 sensor attached to the right wrist for hand action spotting, and a smartphone in the hip pocket for sit detection and data logging. Both IMUs sampled acceleration, rotation, and magnetic field data at 100Hz and streamed the data to the phone for offline analysis. In addition, the participants wore a GoPro HD2 camera on the head. We postannotated the videos for ground-truth activity labels and acquired building layouts that visually confirm the correctness of the S-SMART output tracks. Figure 6 depicts the sensor setup and a part of the environment arrangement for an example recording, and Table II summarizes the dataset characteristics.

[1] www.exelmicroel.com/products_medical_exls3.html.

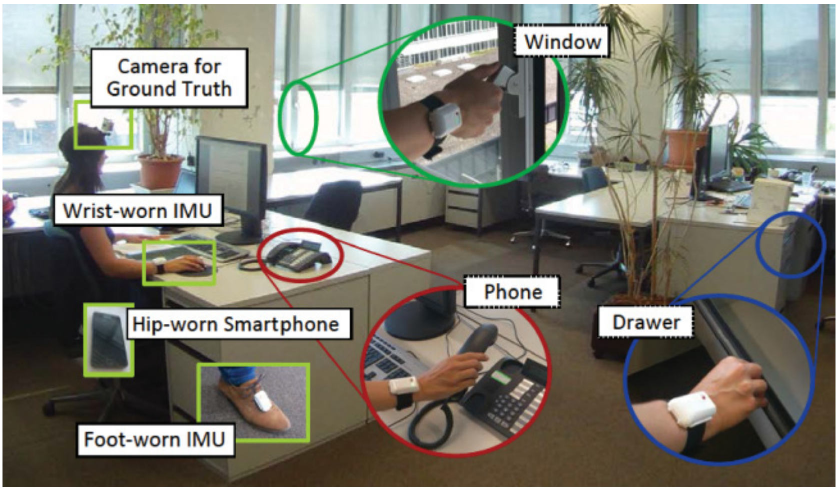

Fig. 6. This picture shows the fully wearable sensor setup in the experimental recordings, and some objects the person interacted with in experiment W8.

In dataset D_1, the 10 participants performed a sequence of predefined object interactions once during an initial training phase. These recordings were later used as templates $\{M_A^T\}$ for Warping LCSS. During the remainder of the experiment, the participants were free in the order they performed actions, as long as they interacted with each object at least four times. While they performed some unanticipated hand motions (e.g., scratching the neck, carrying something), they mostly conformed to the prelearned actions, completing the task within 5 to 20 minutes. Experiments took place in workplace (W1–6) and home (E1–4) settings. In total, the participants performed 554 labeled interactions with up to 11 objects per recording, and walked for 3.80km. The number of object-related activities varied across the experiments, from seven classes in W1 (including the null-class) up to 23 classes in E1.

The second set of experiments D_2 involved scenarios that more closely resemble daily-life behavior. Again, we asked 10 people to wear the sensor setup, but this time during 35 to 54 minutes at their workplace (W7–12) or while cooking, cleaning, and eating at home (E5–8). In a discussion before the recording, we identified objects with which they were likely to interact during that time. We then recorded training data with each participant performing the identified interactions five times. From the corresponding templates, S-SMART chose the best action representation by calculating the intertemplate LCSS distances and selecting the template with minimum mean distance to the others. Subsequently, the main experiment started and the participants were free in their activities, except that we asked them to always use the sensor-equipped hand when interacting with one of the previously selected objects. To induce activity, we asked the experiment participants to complete higher-level tasks during the recording, for example, having breakfast at home or printing papers and bringing them to a drawer in the office. In between these instructed routines, the participants performed various other activities, such as brushing the teeth, searching for a lost phone, or meeting colleagues for a coffee break. In total, the participants walked for 4.62km, performed 386 labeled actions, and interacted with up to 10 preselected objects per recording. Due to the less constrained setup than in dataset D_1, D_2 contains many unanticipated hand motions that were not included in the training dataset.

Table II. Experimental Dataset Statistics, Averaged for 10 Repeated Algorithm Executions

	Experiment Characteristics			Warping LCSS			S-SMART		
Name	L [m]	N_A	Description of Recording	F_1	PR	REC	F_1	PR	REC
W1	127	18	Open/close 1 door and 2 windows	0.97	0.97	0.96	0.99	1	0.98
W2	300	72	Open/close 1 door, 3 drawers, 4 windows, and a fridge	0.84	0.87	0.82	0.95	0.96	0.93
W3	389	50	Open/close 2 doors, 2 drawers, and 3 windows	0.77	0.78	0.75	0.86	0.91	0.81
W4	388	39	Open/close 1 window, use 3 water taps, water plant, manipulate lever, turn book pages	0.79	0.78	0.81	0.97	0.99	0.95
W5	435	55	Open/close 1 door, 1 drawer, and 2 windows; use 1 water tap; water plant; manipulate lever	0.79	0.83	0.75	0.92	0.99	0.87
W6	549	65	Open/close 2 doors, 2 drawers, and 2 windows; use 2 water taps; water plant; drink at table	0.73	0.75	0.72	0.86	0.91	0.81
E1	361	60	Open/close 3 doors, 2 drawers, and 6 windows	0.76	0.78	0.74	0.88	0.89	0.86
E2	330	66	Open/close 2 doors and 4 windows, use 2 water taps, drink at table	0.62	0.68	0.58	0.83	0.9	0.77
E3	416	58	Open/close 4 drawers and 3 windows, use 1 water tap	0.72	0.74	0.69	0.83	0.88	0.8
E4	507	71	Open/close 2 doors, 1 drawer, and 3 windows; use 2 water taps	0.78	0.81	0.76	0.92	0.95	0.89
W7	424	40	Working in office, getting water, drinking, answering calls, printing, etc.	0.67	0.74	0.61	0.8	0.88	0.74
W8	520	42	As W7, but different person	0.57	0.69	0.49	0.68	0.92	0.54
W9	571	42	As W7, but different person	0.6	0.61	0.58	0.65	0.7	0.6
W10	259	20	As W7, but different person	0.42	0.43	0.41	0.54	0.62	0.49
W11	856	44	Lab and computer work, including hand cleaning, visits to colleagues, printing, etc.	0.49	0.52	0.48	0.69	0.75	0.63
W12	299	34	Cutting papers with machine, working at computer, cleaning lab kitchen	0.53	0.69	0.44	0.84	0.86	0.82
E5	358	49	Going to the kitchen for breakfast, brushing teeth, working	0.48	0.45	0.52	0.65	0.7	0.61
E6	258	34	Socializing, getting tea for guests from kitchen, opening/closing windows	0.76	0.78	0.74	0.79	0.86	0.74
E7	358	37	Getting food from fridge, eating in front of TV, opening/closing windows	0.47	0.66	0.36	0.64	0.77	0.55
E8	716	44	Watering plants, opening/closing windows, looking for phone, getting water, etc.	0.33	0.32	0.33	0.53	0.56	0.49

Name is the abbreviation with which we refer to single recordings in this text, L is the distance walked by the user, and N_A the number of his or her labeled activities. The Warping LCSS results are for a parameter set that is specifically optimized for Warping LCSS without S-SMART fusion. Therefore, these numbers can be considered as state-of-the-art performances of a motion-only action recognition system. W1–12 were performed in a workplace environment, and E1–8 in home-like settings. While the workplace experiments took place in single-floor environments, all the at-home recordings except for E4 included multifloor walking. The listed evaluation results are for user-dependent action recognition: that is, the training templates are from the same person who later performed the actions.

5.2. Parameter Optimization

The foot-mounted sensor, the ZUPT-PDR algorithm for open-loop tracking, and the motion model that we apply to our experimental data are all identical to the work on ActionSLAM in Hardegger et al. [2015]. Therefore, we reused the same motion model and object position parameters in the evaluation of S-SMART. For the action recognition system Warping LCSS, we apply the parameters that Nguyen-Dinh et al. [2012] propose. The remaining parameters of S-SMART are the *state transition probability* c_0, the *object insertion probability* p_0, the *minimal observation probability* p_1, and the *map maintenance threshold* V_0. We optimized these parameters with

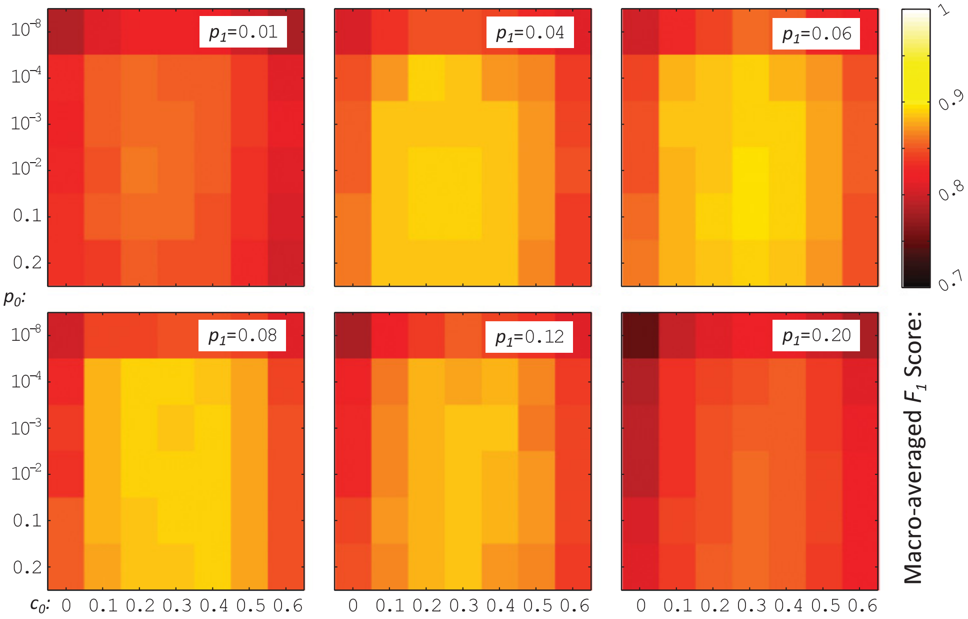

Fig. 7. Averaged F_1 scores for the S-SMART output in all 10 recordings of D_1, with 10 repeated runs per parameter combination. Best results are achieved with $c_0 = 0.3$, $p_0 = 0.01$, and $p_1 = 0.06$. The figure shows that S-SMART is not very sensitive to the choice of parameters.

regard to the macro-averaged F_1 score [Sokolova and Lapalme 2009] in recognizing object-specific actions such as D1_OPEN or W5_CLOSE. For the definition of True Positive (TP) recognitions, False Positives (FPs), and False Negatives (FNs), we used the event-based performance measures proposed in Ward et al. [2011]. The object-specific F_1 score is also a measure for the algorithm's localization robustness, given that incorrect mappings lead to confusion in the object identifier (e.g., adding a door D3 even though in reality, the person revisited D2) and thus reduced the F_1 score.

To avoid overfitting, the parameter optimization was done in dataset D_1 only. In preliminary tests, we fixed the map maintenance threshold to $V_0 = 7$, meaning that S-SMART removes object landmarks if the ratio between visits with object interaction and without interaction is lower than 1 : 7. For the remaining parameters c_0, p_0, and p_1, we performed extensive parameter sweeps. Due to the probabilistic nature of S-SMART, we always averaged outcomes of 10 repeated runs with identical settings. Figure 7 depicts the average results for these sweeps over all 10 recordings of dataset D_1. The best results are achieved with the state transition probability $c_0 = 0.3$, object insertion probability $p_0 = 0.01$, and minimal observation probability $p_1 = 0.06$. Increasing p_0 and p_1 reduces the algorithm's precision at the cost of more missed action observations. Changes in c_0 lead to more substitution errors (e.g., confusion between D_OPEN and D_CLOSE). Overall, S-SMART is not very sensitive to the choice of parameters: within the ranges $c_0 \in [0.1, 0.4]$, $p_0 \in [0.0001, 0.2]$, and $p_1 \in [0.04, 0.12]$, the performance stays within 5% of the optimum. Scenario-independent parameter sets can therefore be used without a significant impact on performance. An additional parameter sweep for the number of particles N_p showed that for $N_p \geq 500$ particles, the algorithm's averaged performance does not further improve. All results that we report in this article are for $N_p = 1{,}000$ particles.

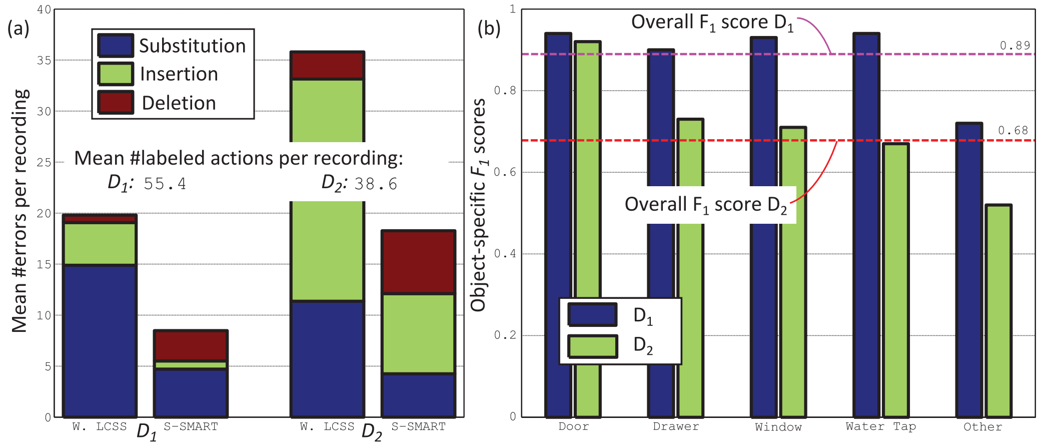

Fig. 8. Part (a) shows the error type distributions for Warping LCSS, and after S-SMART fusion. In D_1, the majority are substitution errors, which S-SMART partly corrects. In D_2, which includes a variety of unexpected movements, more insertions occur, which S-SMART again partly removes. Part (b) depicts the recognition performances for interactions with different objects. Generally, S-SMART performs best for door interactions and much worse for some other movements (e.g., answering phone, watering plants). Door, window, and similar interactions correspond to characteristic motion patterns that Warping LCSS can reliably spot, whereas template matching is not successful with detecting, for example, the hand motions involved in watering a plant.

5.3. Performance Analysis

For performance analysis, we used the optimized parameter set from the previous section. As this optimization was done with dataset D_1, the results obtained for D_1 may be biased, although the parameter optimization indicates that S-SMART is insensitive to the choice of parameters. With two subsequently noted exceptions, we used the same settings in dataset D_2, thus ensuring that the results are not affected by parameter overfitting. As a baseline for action recognition performance, we use Warping LCSS. On the other hand, we compare localization and mapping results to open-loop tracking with ZUPT-PDR and closed-loop ActionSLAM.

5.3.1. User-Dependent Action Recognition. Table II lists the precision ($PR = \frac{TP}{TP+FP}$) and recall rates ($REC = \frac{TP}{TP+FN}$) for all recordings, together with the F_1 scores ($F_1 = 2 \cdot \frac{PR*REC}{PR+REC}$). Overall, the results show a significant ($p < 0.01$) improvement for S-SMART over Warping LCSS (in D_1, the mean F_1 score improves from 78% to 89%, and in D_2 from 53% to 68%). This confirms that adding location awareness and state attributes assists the recognition of actions from body-worn sensors. S-SMART results for the object-unspecific and object-specific analyses are identical, except for when the location tracking fails. Here, we report the object-specific performance results for S-SMART only. On the other hand, the listed Warping LCSS results are for object-unspecific action recognitions (e.g., the actions D1_OPEN and D2_OPEN in the S-SMART evaluation are both D_OPEN activities in the Warping LCSS evaluation). A location-agnostic algorithm such as Warping LCSS performs badly in object-specific recognition tasks as it cannot differentiate objects of the same type.

Figure 8(a) shows the distribution of error types in D_1 and D_2, confirming the following characteristics of S-SMART: First of all, the accumulation of information about a place, the introduction of state attributes to objects, and the application of the Viterbi algorithm to interaction sequences result in a strong reduction of substitution errors (60%–70% compared to Warping LCSS). The framework resolves object type confusions (e.g., between W_OPEN and D_OPEN by taking into account the prior likelihood for a window or door being at this place) and interaction confusion (e.g., between W_OPEN and W_CLOSE). Second, the map maintenance step in S-SMART removes most of the insertion errors due to hand motions at unexpected places. This further improves the algorithm's precision, but comes at the cost of additional deletion errors. This is because map maintenance in S-SMART removes action recognitions that correspond to infrequently observed objects (deleting the object from the map). The tradeoff between insertion and deletion errors is not very relevant in dataset D_1, which anyway includes only few unanticipated motions. However, it has a stronger effect on the performance in the real-world scenarios of D_2.

Table II shows that the S-SMART action recognition performance varies across the recordings of D_1 and D_2. The first reason for this variance is that the number and type of objects with which the users interacted was different in every single recording. As Figure 8(b) depicts, interactions with some objects can be recognized with much higher accuracy than others. The second reason is that recordings took place in different buildings, with different window and door handles, water taps, and more or less complex building layouts. For example, the window handles in the workplace environment (W7–12) were very different from the door handles in the same building, and action recognition had no difficulty in separating interactions with these two object categories. On the other hand, in some of the home environments, window and door handles were similar, and more confusion errors occurred. Third, even in the same building and when interacting with the same objects, people performed some of the movements in different ways, adding further variance to the performance outcomes. Finally, the number of insertion errors varied, depending on the person's behavior. For example, the person in E8 was looking for his lost phone and while doing so he opened several drawers. The drawer-opening action was not in the training set for E8, and many of the corresponding motions were confused with window-opening and other gestures.

Even with all these variations across individual recordings, S-SMART consistently outperforms Warping LCSS in both datasets, showing that the unsupervised learning of location-to-activity mapping with S-SMART works, that the framework improves the overall recognition accuracy, and that it can distinguish interactions with multiple objects without deterioration of the recognition performance. All of these statements could be confirmed in real-world settings. The introduction of states to objects furthermore leads to a performance increase compared to LocAFusion [Hardegger et al. 2014], which also uses semantic maps for action recognition but does not keep track of object states. In D_1, LocAFusion achieves an averaged F_1 score of 82%, which is 7% below the reported S-SMART performance.

5.3.2. Mapping and Localization Performance. In both datasets, S-SMART robustly converges to accurate depictions of the person's path and the map of the building in which he or she moves. With exception of W9 and E6, 100% of the visually analyzed S-SMART output paths $\bar{s}^t$ and maps $\bar{\Theta}_t$ were topologically correct. In W9 and E6, the open-loop tracking estimates $\hat{u}^t$ are affected by large errors that cause S-SMART with standard parameter settings to fail (in both cases, this happened when the person moved on a swivel chair). After increasing the motion model noise parameters, 70% of the runs in these two scenarios converged correctly. Figure 9 depicts three example semantic maps created by S-SMART, together with the open-loop path $\hat{s}^t$ and the algorithm's posterior path estimate $\bar{s}^t$. While the state-of-the-art open-loop tracking algorithm ZUPT-PDR accumulates large heading errors during the time of the recording, S-SMART can correct these and provide a robust estimate of the person's path. As a reference algorithm for closed-loop tracking from wearable IMUs, we applied ActionSLAM to

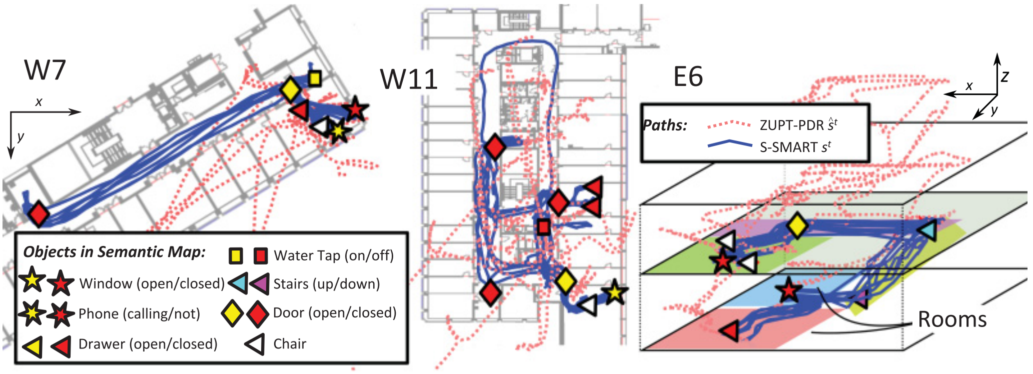

Fig. 9. Open-loop tracking paths $\hat{s}^t$ and S-SMART posterior paths $\bar{s}^t$ for example runs (single-floor W7, W11; multifloor E6), together with the objects in the semantic map and their state at the end of the recording. The object type is indicated by the shape of the landmark, its state by the color. The plots show that S-SMART successfully compensates for the accumulated errors of ZUPT-PDR.

the same tracks, using only basic action observations for mapping. With standard settings, ActionSLAM also failed in W9 and E6, and it performed slightly worse in the remaining scenarios (100% robustness in D_1, 90% in D_2), mainly because it sometimes confuses nearby sitting locations, leading to a folding of the map.

5.3.3. User-Independent Performance. For the experiments W7 through W10, which all took place in the same building, we performed a user-independent performance analysis. The participants in these experiments interacted with different windows, doors, and so forth, but the handles were all similar. For analysis, we applied S-SMART with the prerecorded templates of one subject to the test data of a second subject. While the F_1 score decreased overall from 68% to 42%, the deterioration was minor in some of the combinations. For example, training templates from W8 worked well with W7 (79%) and vice versa (55%), even though these were two different people walking around at different times in different parts of the building. The ground-truth videos confirm that these two experiment participants used the handles in a similar way. On the other hand, the performances for W9 are below 36% with training templates from any other subject, and indeed the person manipulated window and door handles differently from everyone else. This indicates that user-independent action templates cannot cover the full interpersonal variability in human motion. Therefore, larger training datasets or algorithms that adapt the training template set at runtime are necessary for user-independent S-SMART. In a preliminary test, we performed a leave-one-out cross-validation with W7 through W10, in which we used templates of three recordings in the analysis of the fourth experiment. The average F_1 score in this case improves to 51%, but the additional template matching runs multiply the computational effort for action recognition. We propose more elegant approaches to this issue in Section 6.

5.4. Enhanced Experience Sampling Through S-SMART

So far, we focused on fusing open-loop positioning from a foot-worn motion sensor with action recognitions from a wrist-mounted IMU. However, by adapting the S-SMART motion and observation models, the framework can fuse any comparable location and activity sources. Another example application that may benefit from S-SMART is the correction of user-provided activity labels in experience sampling tasks. These user labels often contain substitution errors [Klasnja et al. 2008], which fusion with location data may eliminate. Such a system could consist of a smartphone only, with background

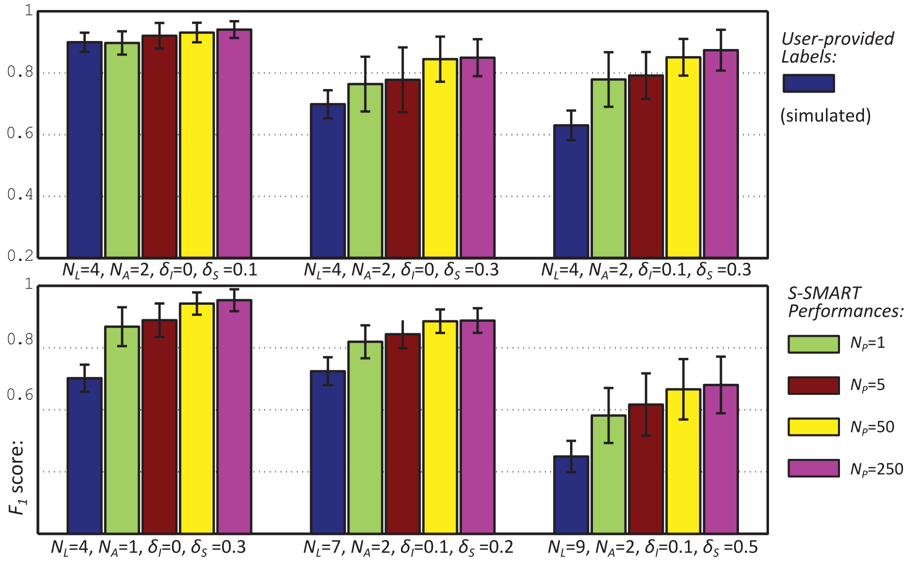

Fig. 10. These bar plots show averaged action recognition accuracies for 100 repeated runs, all of which consist of $N_a = 100$ ground-truth actions. The bars represent the outputs for simulated user labeling and after S-SMART fusion with location in function of the number of particles N_p. For all settings, the accuracy of the user labels significantly improved with S-SMART, and got better the more particles N_p were used in filtering. For the same recordings, the location estimation error per action typically decreased to around $0.5 \cdot \sigma_{GPS}$.

GPS logging and a pop-up questionnaire in which the user selects the activity he or she currently performs.

We simulated this scenario by creating artificial datasets with N_a activity labels from N_L locations, assuming that at each location maximally N_A activity types occur. For example, a person may visit $N_L = 4$ separate places during a regular day: home, an office building where he or she works and which has a restaurant for lunch, the fitness studio, and a mall. The corresponding activities could be sleeping, relaxing, and eating at home ($N_A = 3$); working and eating at the office building ($N_A = 2$); exercising in the fitness studio ($N_A = 1$); and shopping in the mall ($N_A = 1$). We model the GPS scans to be affected by Gaussian noise with standard deviation σ_{GPS}, and the user labels as obstructed with substitution errors (rate δ_S) and insertion errors (rate δ_I). We ignore deletion errors, since S-SMART cannot correct them anyway. An additional proportion δ_E of the activity labels corresponds to exceptional activities that the user performs seldomly and at alternative places, such as going to the hairdresser. We implemented the corresponding motion and observation models in S-SMART and investigated the outcomes for various settings; see Figure 10. S-SMART improves the activity labeling in all these scenarios and at the same time reduces the positioning error, although the extent of both effects depends on the detailed problem configuration. The simultaneously created semantic maps reflect the purpose of each visited building.

6. DISCUSSION

The analysis in the previous section confirms that the semantic maps S-SMART builds accurately reflect the environment state and can therefore be used to enhance activity recognition and location tracking. With the proposed sensor setup made up of just two wearable IMUs and a smartphone, we target a wearable, unobtrusive, and user-friendly system that achieves adequate performance for specific applications, rather than maximal accuracy at the cost of low usability. Most of the remaining action recognition errors in the experiments are due to a subset of motions that seem to be particularly hard to spot from a single wrist-worn IMU. Figure 8 shows that the accuracy varies across the action types, with door and window interactions being recognized more robustly than, for example, the shorter water-tap interactions. Applications that focus on the recognition of a few robustly detectable actions are already possible with the presented setup, while more advanced applications and higher robustness will require additional sensors. For example, a second IMU on the upper arm would enable more precise motion tracking. Also, complementary modalities such as the user's heading when performing an action, sounds that a motion causes, or Wifi and magnetic field fingerprints may improve the performance. Such additional measurements can easily be integrated in the framework by extending the dimensionality of $\hat{M}_t$ and adapting the corresponding models $p(A_t|\hat{M}_t)$. It is also possible to use alternative action recognition methods instead of Warping LCSS, in particular feature-based classification as in Bao and Intille [2004] if this method is better suited for recognizing the activities of interest.

While the location-to-action mapping in S-SMART does not require supervised training at the experiment location, a remaining issue of the presented implementation is the need for pretrained motion templates $\{M_A^T\}$. This means that the user still needs to provide training templates for each action of interest, even though these templates do not need to be collected in the target environment. The user-independent analysis in Section 5.3.3 shows that with action templates recorded by other people, the performance decreases. Also, we observed large differences in the way people perform actions at the time of model training and during the main run of experiments. This further limits the recognition accuracy of both Warping LCSS and S-SMART. To improve the template set without the need for complex model training, an adaptive implementation of S-SMART could modify the templates at runtime. Starting from an initial, user-independent set, this adaptive S-SMART system would detect actions that repeatedly occur at a specific place. It could then replace the initial templates with novel movement data that better represents how the user actually performs this activity. Even a semisupervised approach is possible, in which S-SMART autonomously learns motion patterns that repeatedly occur at the same place and then assigns object labels from voluntary user inputs.

With S-SMART calculating user paths and activity sequences from fully wearable sensor setups, the framework is ideally suited for daily-life monitoring applications. For example, S-SMART could measure for how long a specific window is left open per day, identify frequently and infrequently used drawers, or estimate the amount of water taken from a water tap (by measuring the time for which it is turned on). The sequence of object interactions may furthermore reveal higher-level activity routines, for example, *preparing lunch* or *taking medication*, as previously seen in smart homes [Philipose et al. 2004; Van Kasteren et al. 2008; Rashidi and Cook 2013]. A fully standalone system that does not require preinstalled infrastructure or prior mapping would strongly simplify the deployment of such systems to private homes or workplaces.

In addition to monitoring applications, S-SMART could be used for real-time context-aware assistance, which requires the implementation on wearable devices such as smartphones. The computational requirements in that case largely depend on the individual components for activity recognition and open-loop tracking. The Warping LCSS spotting in the presented system was specifically developed for efficient online action recognition [Roggen et al. 2015]. ZUPT-PDR and the ActionSLAM algorithm, which has a similar computational complexity to S-SMART, were previously implemented for real-time tracking on Android phones [Hardegger et al. 2015]. Our nonoptimized Matlab implementation of S-SMART typically processes experimental data in $\sim 50\%$

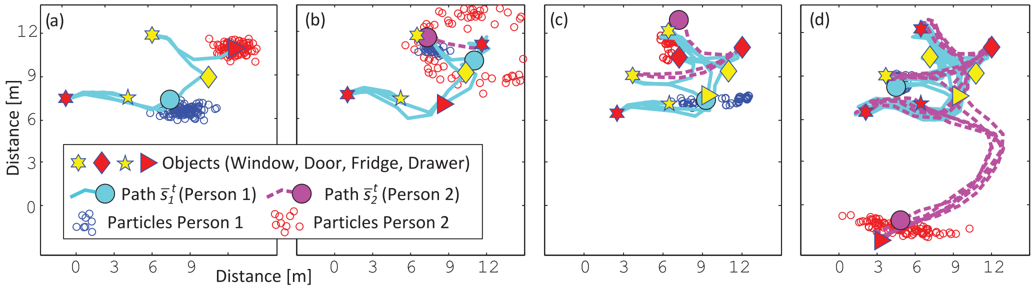

Fig. 11. Map $\bar{\Theta}_t$ and paths $\bar{s}_1^t$ and $\bar{s}_2^t$ of two people as found by Collaborative S-SMART. In this recording, person 2 started at the same place as person 1, but with unknown heading. Therefore, the relative orientation between the two is unknown in the beginning (see (a)), resulting in an initial spreading of the particle cloud for $s_2^{t,[m]}$ (see b)). After a while, the particle filter learns the orientation difference and accurately estimates the full state (see (c) and (d)). N_p was set to 5, 000 in this analysis.

of the recording's duration. All of this suggests that S-SMART can run on wearable assistants and in the future provide real-time context awareness.

7. COLLABORATIVE S-SMART

Usually, more than one person inhabits an environment, and they potentially interact with the same objects. S-SMART can easily be extended to simultaneously localize more than one person moving within the same map. In that case, the particle filter has to sample multiple paths $s_1^t, s_2^t, \ldots$ and actions $A_{1,t}, A_{2,t}, \ldots$ but still update only a single semantic map Θ_t. For the case of two people, the following factorization applies:

$$\begin{aligned} P(s_1^t, \quad & A_{1,t}, s_2^t, A_{2,t}, \Theta_t | \hat{u}_1^t, \hat{M}_1^t, \hat{u}_2^t, \hat{M}_2^t) \\ & = p(s_1^t|\hat{u}_1^t, s_{1,t=0}) p(s_2^t|\hat{u}_2^t, s_{2,t=0}) \prod_{n=1}^{N_{\theta,t}} p(A_{1,t}, A_{2,t}, \theta_{n,t}|s_1^t, s_2^t, \hat{M}^t, \hat{n}^t). \qquad (18) \end{aligned}$$

We implemented this collaborative algorithm as an offline extension to the S-SMART implementation in Section 4. For testing, we recorded data with the identical sensor setup as in the single-person S-SMART evaluation, but with multiple people walking around at the same time in a single area. Figure 11 shows an example run of collaborative S-SMART with two people who started walking at the same place. Due to the higher number of paths that the filter must track, the required number of particles N_p increases for multiperson S-SMART.

8. CONCLUSION

In this article, we introduced the generic Bayesian framework S-SMART that fuses open-loop tracking with activity recognitions to build and update a dynamic semantic map. It does so without the need for supervised training of the location-activity relationship. The map then resets accumulated position errors and simultaneously acts as a prior on new observations. The algorithm links locations to objects that the person modifies, thus making it possible to distinguish identical interactions with different objects.

We implemented this algorithm as a fully wearable system and experimentally showed its potential in terms of improving state-free, location-agnostic activity recognition, as well as in long-term position tracking. In real-world experiments, S-SMART achieved an action recognition F_1 score of 68% in problems with up to 23 action classes. Its mapping and localization robustness was 100% for eight out of 10 recordings. The framework is modular and can benefit from improvements in individual components, such as the template matching or the open-loop tracker. Since S-SMART does not require predeployed infrastructure and only little training data, potentially acquired by other users, it supports efficient deployment of complex monitoring and context-aware feedback applications.

With its combined, fully wearable approach to location and activity tracking by means of dynamically updated semantic maps, S-SMART provides the fundamentals for the next generation of wearable assistance scenarios. It is an important step toward fine-grained life logging and cognitive prostheses, while decreasing (or avoiding altogether, as in this article) the need for ambient sensing. As a self-contained system that senses and models on the wearables, S-SMART also addresses privacy concerns often raised with ambient instrumentation.

REFERENCES

Amin Ahmadi, Edmond Mitchell, Francois Destelle, Marc Gowing, Chris Richter, Noel E. O'Connor, and Kieran Moran. 2014. Automatic activity classification and movement assessment during a sports training session using wearable inertial sensors. In *Proceedings of the 11th International Conference on Wearable and Implantable Body Sensor Networks*. IEEE.

Alberto Alvarez-Alvarez, José M. Alonso, Gracian Trivino, Noelia Hernández, Fernando Herranz, Ángel Llamazares, and Manuel Ocana. 2010. Human activity recognition applying computational intelligence techniques for fusing information related to WiFi positioning and body posture. In *Proceedings of the International Conference on Fuzzy Systems*. IEEE.

Michael Angermann and Patrick Robertson. 2012. FootSLAM: Pedestrian simultaneous localization and mapping without exteroceptive sensors - hitchhiking on human perception and cognition. *Proceedings of the IEEE* 100 (2012), 1840–1848.

Tim Bajarin. 2014. Where Wearable Health Gadgets Are Headed. Retrieved from http://time.com/2938202/health-fitness-gadgets/.

Ling Bao and Stephen S. Intille. 2004. Activity recognition from user-annotated acceleration data. In *Pervasive Computing*. Springer, 1–17.

Donald J. Berndt and James Clifford. 1994. Using dynamic time warping to find patterns in time series. In *KDD Workshop*, Vol. 10. 359–370.

Andreas Bulling, Ulf Blanke, and Bernt Schiele. 2014. A tutorial on human activity recognition using body-worn inertial sensors. *Computer Surveys* 46, 3 (2014), 33.

Ricardo Chavarriaga, Hesam Sagha, Alberto Calatroni, Sundara Tejaswi Digumarti, Gerhard Tröster, José del R. Millán, and Daniel Roggen. 2013. The opportunity challenge: A benchmark database for on-body sensor-based activity recognition. *Pattern Recognition Letters* 34, 15 (2013), 2033–2042.

Liming Chen, Chris D. Nugent, and Hui Wang. 2012. A knowledge-driven approach to activity recognition in smart homes. *IEEE Transactions on Knowledge and Data Engineering* 24, 6 (2012), 961–974.

Burcu Cinaz and Holger Kenn. 2008. HeadSLAM-simultaneous localization and mapping with head-mounted inertial and laser range sensors. In *Proceedings of the 12th International Symposium on Wearable Computers*. IEEE, 3–10.

M. W. M. Gamini Dissanayake, Paul Newman, Steven Clark, Hugh F. Durrant-Whyte, and Michael Csorba. 2001. A solution to the simultaneous localization and map building (SLAM) problem. *IEEE Transactions on Robotics and Automation* 17, 3 (2001), 229–241.

Randal Douc and Olivier Cappé. 2005. Comparison of resampling schemes for particle filtering. In *Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis*. IEEE, 64–69.

Hugh F. Durrant-Whyte and Tim Bailey. 2006. Simultaneous localization and mapping: Part I. *IEEE Robotics & Automation Magazine* 13, 2 (2006), 99–110.

Nathan Eagle and Alex Pentland. 2006. Reality mining: Sensing complex social systems. *Personal and Ubiquitous Computing* 10, 4 (2006), 255–268.

George David Forney Jr. 1973. The viterbi algorithm. *Proceedings of the IEEE* 61, 3 (1973), 268–278.

Eric Foxlin. 2005. Pedestrian tracking with shoe-mounted inertial sensors. *Computer Graphics and Applications* 25, 6 (2005), 38–46.

Caroline Free, Gemma Phillips, Leandro Galli, Louise Watson, Lambert Felix, Phil Edwards, Vikram Patel, and Andy Haines. 2013. The effectiveness of mobile-health technology-based health behaviour change or disease management interventions for health care consumers: A systematic review. *PLoS Medicine* 10, 1 (2013).

Cipriano Galindo, Juan-Antonio Fernández-Madrigal, Javier González, and Alessandro Saffiotti. 2008. Robot task planning using semantic maps. *Robotics and Autonomous Systems* 56, 11 (2008), 955–966.

Slawomir Grzonka, Andreas Karwath, Frederic Dijoux, and Wolfram Burgard. 2012. Activity-based estimation of human trajectories. *IEEE Transactions on Robotics* 28, 1 (2012), 234–245.

Michael Hardegger, Long-Van Nguyen-Dinh, Alberto Calatroni, Gerhard Tröster, and Daniel Roggen. 2014. Enhancing action recognition through simultaneous semantic mapping from body-worn motion sensors. In *Proceedings of the 13th International Symposium on Wearable Computers*. ACM, 99–106.

Michael Hardegger, Daniel Roggen, Sinziana Mazilu, and Gerhard Tröster. 2012. ActionSLAM: Using location-related actions as landmarks in pedestrian SLAM. In *2012 International Conference on Indoor Positioning and Indoor Navigation*. IEEE, 1–10.

Michael Hardegger, Daniel Roggen, and Gerhard Tröster. 2015. 3D ActionSLAM: Wearable person tracking in multi-floor environments. *Personal and Ubiquitous Computing* 19, 1 (2015), 123–141.

Robert Harle. 2013. A survey of indoor inertial positioning systems for pedestrians. *IEEE Communications Surveys & Tutorials* 15, 3 (2013), 1281–1293.

Ville Honkavirta, Tommi Perala, Simo Ali-Loytty, and Robert Piché. 2009. A comparative survey of WLAN location fingerprinting methods. In *Proceedings of the 6th Workshop on Positioning, Navigation and Communication*. IEEE, 243–251.

Chih-Ning Huang, Chih-Yen Chiang, Jui-Sheng Chang, Yi-Chieh Chou, Ya-Xuan Hong, Steen J. Hsu, Woei-Chyn Chu, and Chia-Tai Chan. 2009. Location-aware fall detection system for medical care quality improvement. In *Proceedings of the 3rd International Conference on Multimedia and Ubiquitous Engineering*. IEEE, 477–480.

Predrag Klasnja, Beverly L. Harrison, Louis LeGrand, Anthony LaMarca, Jon Froehlich, and Scott E. Hudson. 2008. Using wearable sensors and real time inference to understand human recall of routine activities. In *Proceedings of the 10th International Conference on Ubiquitous Computing*. ACM, 154–163.

Masakatsu Kourogi and Takeshi Kurata. 2003. Personal positioning based on walking locomotion analysis with self-contained sensors and a wearable camera. In *Proceedings of the 2nd International Symposium on Mixed and Augmented Reality*. IEEE, 103.

Lin Liao, Donald J. Patterson, Dieter Fox, and Henry Kautz. 2007. Learning and inferring transportation routines. *Artificial Intelligence* 171, 5 (2007), 311–331.

Hui Liu, Houshang Darabi, Pat Banerjee, and Jing Liu. 2007. Survey of wireless indoor positioning techniques and systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews* 37, 6 (2007), 1067–1080.

Jun S. Liu and Rong Chen. 1998. Sequential monte carlo methods for dynamic systems. *Journal of the American Statistics Association* 93, 443 (1998), 1032–1044.

Ching-Hu Lu and Li-Chen Fu. 2009. Robust location-aware activity recognition using wireless sensor network in an attentive home. *IEEE Transactions on Automation Science and Engineering* 6, 4 (2009), 598–609.

Stéphane Magnenat, Roland Philippsen, and Francesco Mondada. 2012. Autonomous construction using scarce resources in unknown environments. *Autonomous Robots* 33, 4 (2012), 467–485.

Sinziana Mazilu, Michael Hardegger, Zack Zhu, Daniel Roggen, Gerhard Tröster, Meir Plotnik, and Jeffrey Hausdorff. 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*. IEEE, 123–130.

Johannes Meyer, Paul Schnitzspan, Stefan Kohlbrecher, Karen Petersen, Mykhaylo Andriluka, Oliver Schwahn, Uwe Klingauf, Stefan Roth, Bernt Schiele, and Oskar von Stryk. 2011. A semantic world model for urban search and rescue based on heterogeneous sensors. In *Robot Soccer World Cup XIV*. Springer, 180–193.

Michael Montemerlo, Sebastian Thrun, Daphne Koller, and Ben Wegbreit. 2002. FastSLAM: A factored solution to the simultaneous localization and mapping problem. In *Proceedings of the 18th National Conference on Artificial Intelligence*. American Association for Artificial Intelligence, 593–598.

Long-Van Nguyen-Dinh, Daniel Roggen, Alberto Calatroni, and Gerhard Tröster. 2012. Improving online gesture recognition with template matching methods in accelerometer data. In *Proceedings of the 12th International Conference on Intelligent Systems Design and Applications*. IEEE, 831–836.

Andreas Nüchter and Joachim Hertzberg. 2008. Towards semantic maps for mobile robots. *Robotics and Autonomous Systems* 56, 11 (2008), 915–926.

Kwanghyo Park, Hyojeong Shin, and Hojung Cha. 2013. Smartphone-based pedestrian tracking in indoor corridor environments. *Personal and Ubiquitous Computing* 17, 2 (2013), 359–370.

Donald J. Patterson, Dieter Fox, Henry Kautz, and Matthai Philipose. 2005. Fine-grained activity recognition by aggregating abstract object usage. In *Proceedings of the 9th International Symposium on Wearable Computers*. IEEE, 44–51.

Matthai Philipose, Kenneth P. Fishkin, Mike Perkowitz, Donald J. Patterson, Dieter Fox, Henry Kautz, and Dirk Hahnel. 2004. Inferring activities from interactions with objects. *IEEE Pervasive Computing* 3, 4 (2004), 50–57.

Parisa Rashidi and Diane J. Cook. 2013. COM: A method for mining and monitoring human activity patterns in home-based health monitoring systems. *ACM Transactions on Intelligent Systems and Technology* 4, 4 (2013), 64.

Patrick Robertson, Martin Frassl, Michael Angermann, Marek Doniec, Briann J. Julian, Maria Garcia Puyol, Mohammed Khider, Michael Lichtenstern, and Luigi Bruno. 2013. Simultaneous localization and mapping for pedestrians using distortions of the local magnetic field intensity in large indoor environments. In *Proceedings of the 4th International Conference on Indoor Positioning and Indoor Navigation*. IEEE, 925–934.

Daniel Roggen, Luis Ponce Cuspinera, Guilherme Pombo, Falah Ali, and Long-Van Nguyen-Dinh. 2015. Limited-memory warping LCSS for real-time low-power pattern recognition in wireless nodes. In *Wireless Sensor Networks*. Springer, 151–167.

Antonio Ramón Jiménez Ruiz, Fernando Seco Granja, Jose Carlos Prieto Honorato, and Jorge I. Guevara Rosas. 2012. Accurate pedestrian indoor navigation by tightly coupling foot-mounted IMU and RFID measurements. *IEEE Transactions on Instrumentation and Measurement* 61, 1 (2012), 178–189.

Philipp M. Scholl, Nagihan Kücükyildiz, and Kristof Van Laerhoven. 2013. When do you light a fire: Capturing tobacco use with situated, wearable sensors. In *Adjunct Proceedings of the International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 1295–1304.

Julia Seiter, Sebastian Feese, Bert Arnrich, Gerhard Tröster, Oliver Amft, Lucian Macrea, and Konrad Maurer. 2013. Evaluating daily life activity using smartphones as novel outcome measure for surgical pain therapy. In *Proceedings of the 8th International Conference on Body Area Networks*. 153–156.

Marina Sokolova and Guy Lapalme. 2009. A systematic analysis of performance measures for classification tasks. *Information Processing & Management* 45, 4 (2009), 427–437.

Thomas Stiefmeier, Daniel Roggen, Georg Ogris, Paul Lukowicz, and Gerhard Tröster. 2008. Wearable activity tracking in car manufacturing. *IEEE Pervasive Computing* 7, 2 (2008), 42–50.

Sebastian Thrun, Michael Montemerlo, Daphne Koller, Ben Wegbreit, Juan Nieto, and Eduardo Nebot. 2004. FastSLAM: An efficient solution to the simultaneous localization and mapping problem with unknown data association. *Journal of Machine Learning Research* 4, 3 (2004), 380–407.

Tim Van Kasteren, Athanasios Noulas, Gwenn Englebienne, and Ben Kröse. 2008. Accurate activity recognition in a home setting. In *Proceedings of the 10th International Conference on Ubiquitous Computing*. ACM.

Michail Vlachos, Marios Hadjieleftheriou, Dimitrios Gunopulos, and Eamonn Keogh. 2003. Indexing multidimensional time-series with support for multiple distance measures. In *Proceedings of the 9th International Conference on Knowledge Discovery and Data Mining*. ACM, 216–225.

Chieh-Chih Wang, Charles Thorpe, Sebastian Thrun, Martial Hebert, and Hugh Durrant-Whyte. 2007. Simultaneous localization, mapping and moving object tracking. *International Journal of Robotics Research* 26, 9 (2007), 889–916.

He Wang, Souvik Sen, Ahmed Elgohary, Moustafa Farid, Moustafa Youssef, and Romit Roy Choudhury. 2012b. No need to war-drive: Unsupervised indoor localization. In *Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services*. ACM, 197–210.

Liang Wang, Tao Gu, Xianping Tao, and Jian Lu. 2012a. A hierarchical approach to real-time activity recognition in body sensor networks. *Pervasive and Mobile Computing* 8, 1 (2012), 115–130.

Jamie A. Ward, Paul Lukowicz, and Hans W. Gellersen. 2011. Performance metrics for activity recognition. *ACM Transactions on Intelligent Systems and Technology* 2, 1 (2011), 6.

Oliver Woodman and Robert Harle. 2008. Pedestrian localisation for indoor environments. In *Proceedings of the 10th International Conference on Ubiquitous Computing*. ACM, 114–123.

Vincent W. Zheng, Yu Zheng, Xing Xie, and Qiang Yang. 2010. Collaborative location and activity recommendations with GPS history data. In *Proceedings of the 19th International Conference on World Wide Web*. ACM, 1029–1038.

Received September 2014; revised June 2015; accepted September 2015