Enhancing Action Recognition through Simultaneous Semantic Mapping from Body-Worn Motion Sensors

Michael Hardegger*, Long-Van Nguyen-Dinh, Alberto Calatroni, Gerhard Tröster
Wearable Computing Laboratory
ETH Zürich, Switzerland
*michael.hardegger@ife.ee.ethz.ch

Daniel Roggen*
University of Sussex
Brighton, UK
*daniel.roggen@ieee.org

ABSTRACT

Locations and actions are interrelated: some activities tend to occur at specific places, for example a person is more likely to twist his wrist when he is close to a door (to turn the knob). We present an unsupervised fusion method that takes advantage of this characteristic to enhance the recognition of location-related actions (e.g., open, close, switch, etc.). The proposed LocAFusion algorithm acts as a post-processing filter: At run-time, it constructs a semantic map of the environment by tagging action recognitions to Cartesian coordinates. It then uses the accumulated information about a location i) to discriminate between identical actions performed at different places and ii) to correct recognitions that are unlikely, given the other observations at the same location. LocAFusion does not require prior statistics about where activities occur, which allows for seamless deployment to new environments. The fusion approach is agnostic to the sensor modalities and methods used for action recognition and localization.

For evaluation, we implemented a fully wearable setup that tracks the user with a foot-mounted motion sensor and the ActionSLAM algorithm. Simultaneously, we recognize hand actions through template matching on the data of a wrist-worn inertial measurement unit. In 10 recordings with 554 performed object interactions, LocAFusion consistently outperformed location-independent action recognition (8-31% increase in F1 score), identified 96% of the objects in the semantic map and overall correctly labeled 82% of the actions in problems with up to 23 classes.

Author Keywords

Activity Recognition; Tracking; Wearable; SLAM; LCSS;

ACM Classification Keywords

H.1.2 User/Machine Systems: Miscellaneous.

INTRODUCTION

In daily life, we tend to perform activities at specific locations: In the kitchen we cook, on the couch we relax, in the office we work. Location-based activity recognition aims at exploiting this relationship, either by inferring the person's activities from his location alone, or through fusion with complementary modalities (e.g., motion sensing). An issue for many applications, such as wearable assisted living and memory assistants, is the supervised learning of the probability density function that links motion and location to activities (e.g., [12, 23]). Supervised learning requires users to perform training activities in the target environment prior to application, which is time-consuming and inconvenient for large-scale deployment.

In this paper, we propose a novel, unsupervised method called *LocAFusion* that learns at run-time where a user performs which activities, rather than in a pre-deployment training phase. For this purpose, we introduce the concept of semantic mapping to human activity recognition. These maps tag the location of relevant objects in a local coordinate system. By combining activity recognition with location tracking, LocAFusion builds semantic maps in an unsupervised manner. In post-processing, these semantic maps then help i) to distinguish between identical activities performed at different locations, and ii) to resolve recognition confusions by using the accumulated information about what a person usually does at a certain place.

We evaluate the LocAFusion approach in a practical setup representative for assisted daily-living scenarios: The recognition of hand actions. Our hands are the most important instrument for environment interactions, and automatic recognition of these interactions from wearable sensors enables various applications. For instance, after opening a window, a wearable assistant may remind an elderly person of closing it again before leaving the house. Such systems have been proposed for people at the onset of dementia [8]. Location-awareness is crucial in this case: we can only say whether a specific window is open if we know the user's location when he performs the hand action *open window*.

With such applications in mind, we here introduce the LocAFusion-based *Loc-Action System* that recognizes a set of hand actions (opening and closing windows/doors/drawers, using water taps, watering plants, etc.) from a wrist-worn Inertial Measurement Unit (IMU), and in parallel obtains location from a foot-mounted motion sensor through the Simultaneous Localization And Mapping algorithm ActionSLAM [6]. This positioning algorithm accurately tracks a person solely from the body-worn IMU without relying on pre-installed infrastructure. The system then applies LocAFusion to learn a model of the location-action relationship from data

ISWC'14, September 13-17 2014, Seattle, WA, USA.

ACM 978-1-4503-2969-9/14/09$15.00.
http://dx.doi.org/10.1145/2634317.2634323

collected at run-time, thus avoiding the effort of making people perform each action prior to deployment in their home for model training.

We evaluated the algorithm's performance in 10 recordings with people performing 554 object-related hand actions, as well as in the Opportunity reference dataset [16] with 1485 labeled hand motions. In these scenarios, the Loc-Action System consistently provides better action recognition accuracy than location-agnostic systems. The semantic maps built by LocAFusion accurately reflect the spatial arrangement of objects in the environment. Furthermore, our evaluation indicates that LocAFusion outcomes are similar to those achieved with supervised learning, but at strongly reduced pre-deployment effort, given that no training in the target environment is necessary.

RELATED WORK

Some location-specific user activities may be inferred from location alone, which is for example exploited outdoors with GPS [11, 22], or in smart homes where binary ambient sensors indicate in which room the inhabitants are [21]. Location-based activity recognition is necessarily coarse and cannot distinguish between multiple activities occurring at the same place. For instance, being in the kitchen indicates *cooking*, but does not discriminate *cutting* from *cleaning dishes*. This limitation may be addressed through sensor fusion, taking advantage of two (or more) modalities related to the users activities. A common approach is to combine location with motion sensing, as many activities are characterized by specific body movement patterns (a review of wearable activity recognition is outside of the scope of this paper, see [4]). In that case, location can be a prior on the activity classes detected from the motion sensor, or pre-select a motion classifier that is optimized for the typical activities occurring at a place [18].

Most approaches that fuse location with other sensors follow a supervised learning paradigm. They train joint activity models for all modalities (e.g., location, motion) prior to deployment, and then at run-time check whether sensor readings fulfill the model assumptions for one or multiple trained activities [12]. In well-defined environments, prior knowledge about where specific activities take place may even be preprogrammed. This was for example done in manufacturing scenarios, where users perform activities always at defined locations [18, 19, 23]. Similarly, [20] pre-deployed RFID tags to identify the room in which a person is before recognizing room-specific activities.

However, supervised learning is a significant hindrance for many activity-monitoring scenarios, as learning the location-action relationship requires labeled training data with the users performing activities in the target environment. To our knowledge, few works attempted to build a model of the location-action relationship at run-time. In [10], a fully unsupervised activity discovery process is applied to GPS traces, but the method relies solely on location data and thus does not address the issue of finer activity recognition.

In the next sections, we introduce a post-processing method that combines the outcomes of two independent systems for location tracking and action recognition through the unsupervised construction of a semantic map. Such maps are used in robotics for assisting the navigation planning of mobile agents (e.g., [14]). Here, we apply semantic maps to accumulate information about the activities performed at a location, and thus infer the location-activity model at run-time.

TERMINOLOGY

In the following, we refer to *objects* in the user's environment through the acronym `Object` indicating their *object type* (listed in Table 1), and an `ID`. For example, we abbreviate window 3 with `w3`. Additionally, objects have a state V that a person may change through *interactions* with the object. Most objects in this paper have two states, e.g., $V_{\mathtt{d}}(t) \in \{\text{open,closed}\}$. We group object types in *object categories* and assume that only one object of a category can be at a given location (e.g., doors and windows cannot be at the same place, while a book and a cup may be).

Furthermore, we use the term *action* for short, voluntary hand movements. Object-related actions are actions that only occur when the person is close to an object (e.g., door opening when the person is in reach of a door). For each object-related action, we either report the *object-specific* action type with the acronym `ObjectID_Interaction` (e.g., `w3_open` for opening window 3), or the *object-unspecific* action type by leaving away the ID (e.g., `w_open` stands for the unspecific action of opening any window in the environment).

Table 1. List of objects and interactions referred to in the text.

Object Type	Acronym	Interactions	Category
Window	`w`	open/close	I
Door	`d`	open/close	I
Drawer	`r`	open/close	I
Water tap	`t`	turn on/off	II
Plant	`p`	water	III
Book	`b`	turn page	IV
Switch	`s`	toggle	V
Cup	`c`	drink	VI

THE LOCAFUSION ALGORITHM

In this section we introduce the LocAFusion algorithm for posterior fusion of location with action recognitions. The proposed approach is agnostic to the source of the two input measurements: It only requires the person's track $\vec{x}(t)$ in a local Cartesian coordinate system, and $N_{\hat{\alpha}}$ unspecific action observations $\hat{\alpha}_i = \{t_i^0, t_i^1, \hat{A}_i, \hat{s}_i\}$ as input. Here, t_i^0 and t_i^1 are the start and end time of the observed action, and $\hat{s}_i \in [0, 1]$ a score that describes the certainty about the recognition, as provided by most action recognition systems. Alternatively, $\hat{s}_i$ may be fixed to 1. $\hat{A}_i = \{\hat{O}_i, \hat{I}_i\}$ is the unspecific action type (e.g., `d_open`, `w_close`, etc.), with $\hat{O}_i$ being the type of the object interacted with, and $\hat{I}_i$ the interaction. Table 2 lists the variable names we use in this paper. The outputs of LocAFusion are a corrected list of unspecific actions $\bar{\alpha}$, object-specific actions $\tilde{\bar{\alpha}}$ and the semantic map Θ.

Table 2. List of variable names.

Name	Symbol	Name	Symbol
Action	α_i	Action Location	$\vec{x}_i$
#Actions	N_α	Recognition Score	s_i
Action Type	A_i	Object Type	O_i
		Interaction Type	I_i
		Object Association	n_i
Object	$\theta_{[j]}$	Object Location	$\vec{c}_{[j]}$
#Objects	N_θ	Object Category	$O_{[j]}$
		Object State	$V_{[j]}(t)$

1. Location-Based Action Clustering: First, LocAFusion assigns to each action $\hat{\alpha}_i$ the user's position at the time of the action $\vec{x}_i = \frac{1}{t_i^1 - t_i^0} \int_{t_i^0}^{t_i^1} \vec{x}(t)dt$. Complete linkage clustering [1] with maximal Euclidean distance d_0 then groups the locations $\vec{x}_i$ to a list of action clusters with locations $\vec{c}_{[j]}$. Figure 1 illustrates the outcome of location-based clustering for an artificial example. This clustering is performed for each object category individually, so that multiple clusters can be at a single place, as long as the corresponding interactions are with objects in different categories (e.g., drawer and switch). Cluster associations $n_i = \text{argmin}_n \|\vec{x}_i - \vec{c}_n\|$ associate each action $\hat{\alpha}_i$ to the closest cluster.

2. Semantic Mapping: Next, LocAFusion estimates the most likely object type $\bar{O}_{[j]}$ for each action cluster. For this purpose, the recognition scores $\hat{s}_i$ of all actions with $n_i = [j]$ and $\hat{O}_i = k$ are summed up to a total score $\bar{s}_{[j],k}$, with $[j]$ identifying the cluster, and k the object type. If $\bar{s}_{[j],k} < Th_{score}$ for all object types k, the cluster is deleted from the list, and we discard any action with $n_i = [j]$ as false positive observation of the action recognition system. This eliminates a cluster if the cumulative confidence in the detection of any activity at this location is low. If the total score for at least one object type at the location $\vec{c}_{[j]}$ is above Th_{score}, LocAFusion assigns $\bar{O}_{[j]} = \bar{k}$, with $\bar{k}$ being the object type with highest total score $\bar{s}_{[j],\bar{k}}$. In the example in Figure 1 with $\hat{s}_i = 1 \; \forall \; i \in \{1, \ldots, 11\}$, the scores for the cluster at $\vec{c}_2$ would be $\bar{s}_{2,\mathtt{d}} = 2$ in favor of $\mathtt{d}$ and $\bar{s}_{2,\mathtt{w}} = 1$ for $\mathtt{w}$. The final decision is consequently that $\bar{O}_2 = \mathtt{d}$. Similarly, $\bar{O}_1 = \mathtt{w}$ with $\bar{s}_{1,\mathtt{w}} = 4$ and $\bar{O}_3 = \mathtt{d}$ with $\bar{s}_{3,\mathtt{d}} = 2$ and $\bar{s}_{2,\mathtt{w}} = 1$.

The semantic map Θ combines all this information to a list of N_θ objects $\theta_{[j]} = \{\vec{c}_{[j]}, \bar{O}_{[j]}, \bar{V}_{[j]}(t)\}$, where $\vec{c}_{[j]}$ is the object's position, $\bar{O}_{[j]}$ its type, and $\bar{V}_{[j]}(t)$ the state, which is the result of the last observed interaction $\hat{I}_i$ with the object before the time t, given a set of pre-defined, object-type-specific rules (e.g., after the action `d2_open`, the state of object `d2` is `open`). Figure 2 depicts the semantic map and the corresponding object states as a function of time for an example LocAFusion run.

3. Action Correction: The semantic map specifies what object is most likely at the location $\vec{c}_{[j]}$, given all available information about the person's activities at this place. The action correction step now changes for all actions $\hat{\alpha}_i$ the object type to $\bar{O}_i = \bar{O}_{[n_i]}$. Furthermore, it chooses the interaction $\bar{I}_i$ as the interaction with highest score $\bar{s}_{i,\bar{O}_i}$ in the subset of interactions that are possible with the object $\bar{O}_i$, and discards the action if this score is below the null-class rejection threshold Th_{null}. The final result of this step is a corrected list of object-unspecific actions $\bar{\alpha}_i = \{t_i^0, t_i^1, \bar{A}_i, \bar{s}_i\}$ with $\bar{A}_i = \{\bar{O}_i, \bar{I}_i\}$. In addition, the list of object-specific actions $\tilde{\bar{\alpha}}_i$ can be derived by adding the object associations n_i to the action list, i.e., $\tilde{\bar{\alpha}}_i = \{t_i^0, t_i^1, \bar{A}_i, \bar{s}_i, n_i\}$.

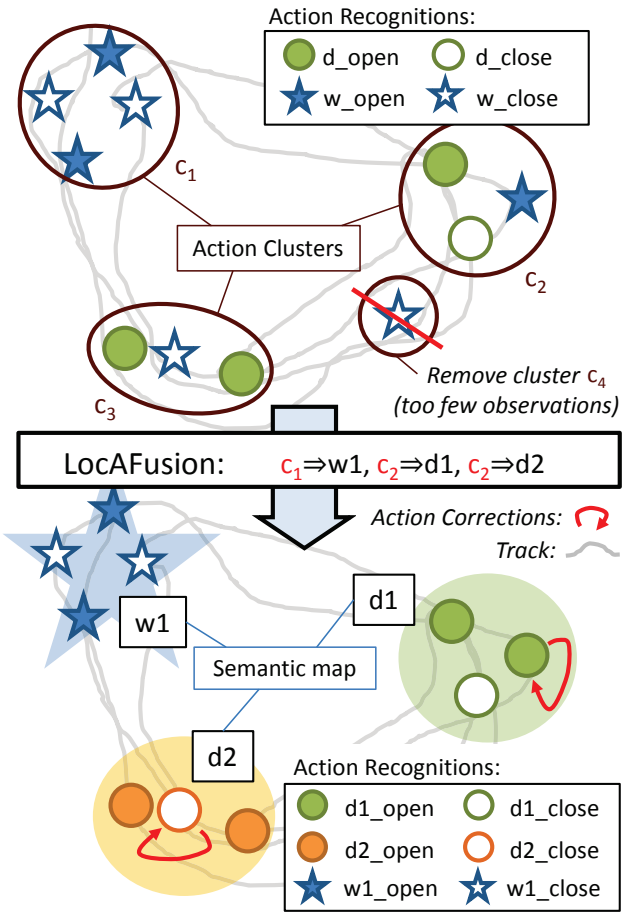

Figure 1. LocAFusion processing for an example walk with door and window interactions.

IMPLEMENTATION

For analysis and evaluation of the proposed LocAFusion algorithm, we implemented the *Loc-Action System* that simultaneously tracks the location and recognizes actions from wearable sensors only, before applying LocAFusion in post-processing. The system is composed of an EXL-S3 IMU[1] attached to the wrist of the right hand for action recognition, and a second IMU at the foot for location tracking. The IMUs stream calibrated acceleration and rotation velocity data at a rate of 100 Hz to a hip-worn smartphone, which synchronizes and logs the data for offline analysis. Figure 3 depicts the

[1] `http://www.exelmicroel.com/products_medical_exls3.html`

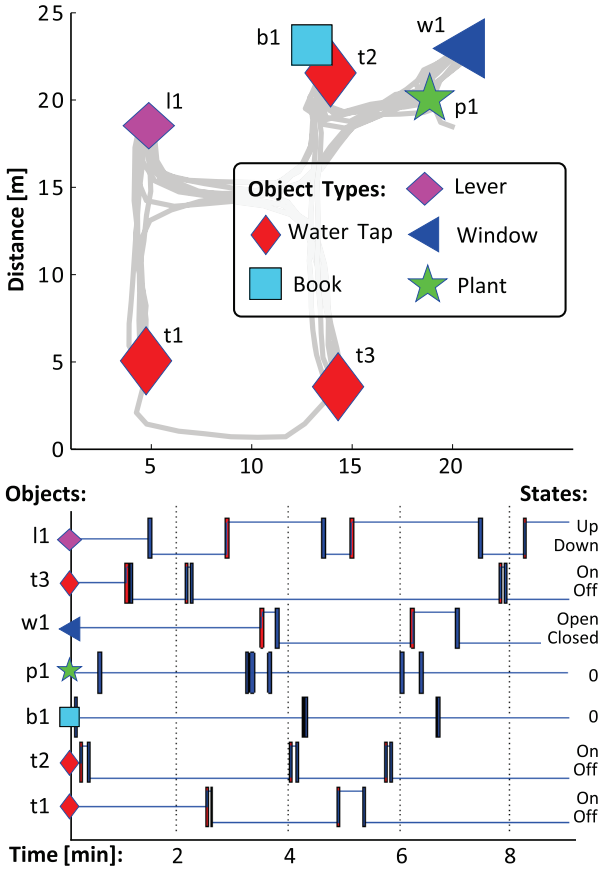

Figure 2. Semantic map Θ for an experimental recording (W4), with the person's track and the estimated object locations and types marked. In addition, the figure shows the object states $V_j(t)$ as a function of time, with the interaction times indicated.

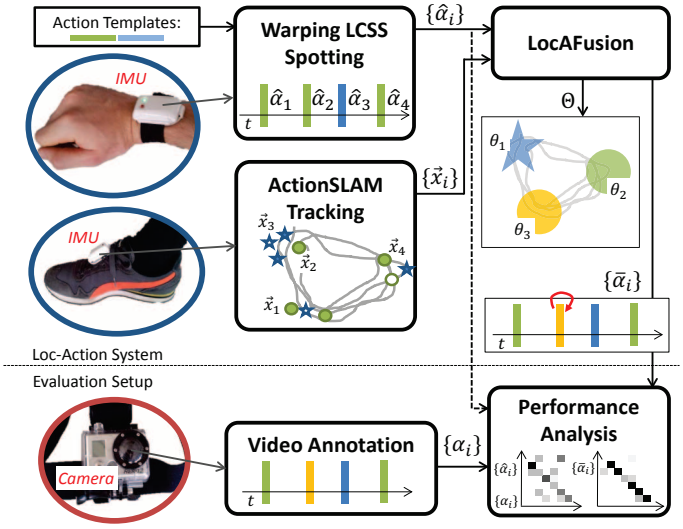

Figure 3. Outline of the proposed Loc-Action System for combined action recognition and location tracking, as well as the camera-based evaluation setup that we use in the next section.

overall system architecture. We selected the individual system components taking into account their accuracy, fast deployment, and low computational effort.

Location Tracking

For indoor location tracking, we use the ActionSLAM system introduced in [6]. ActionSLAM fuses the measurements of the foot-mounted IMU with the recognition of sitting, standing still and stair climbing to build and update a local 3D map of the building. Simultaneously, ActionSLAM uses the map to correct for the error accumulation of open-loop motion integration. The main reason for choosing ActionSLAM in this scenario is the straight-forward deployment, not requiring any pre-installed infrastructure or prior maps of the building. Instead, the algorithm builds the map of the environment by itself, in a fully unsupervised manner. At the same time, ActionSLAM provides high positioning accuracy (mean tracking error ≈ 1.2 m) for walks in constrained indoor areas such as homes or office buildings.

Action Recognition

Template matching approaches were shown to be versatile and capable of spotting specific movements, such as HCI gestures [13] or car manufacturing motions [19], in the signals of body-worn motion sensors. They typically translate the raw IMU data to a sequence of symbols S^m, then compare extracts $S^m_{[W]}$ to pre-trained string templates S^t, and decide based on the similarity $\text{sim}(S^m_{[W]}, S^t)$ whether the motion in window $[W]$ corresponds to the template S^t or not. There are various distance metrics for calculating the similarity between two strings, the best-known being Dynamic Time Warping (DTW) [3]. DTW was for example adapted for the detection of food preparation activities [15] or gestures in a cricket umpire [9]. An alternative to DTW is Longest Common SubSequence (LCSS) matching [7], which is less sensitive to noise than DTW and therefore better suited for real-world deployment. Warping LCSS is a LCSS variant that outperformed DTW in reference datasets while being computationally faster than standard LCSS implementations [13]. In the Loc-Action System, we therefore apply Warping LCSS for recognizing actions from the wrist-mounted IMU. We furthermore use location-agnostic Warping LCSS as reference algorithm for comparison with the Loc-Action System.

We implemented the following processing chain: First of all, we downsample the acceleration and rotation velocity data of the wrist-worn IMU to $f_s = 32$ Hz and translate the raw signals to the string S^m using the k-means algorithm with the alphabet size K as parameter, thus reducing the 6D input data to a 1D symbol sequence. Given the string S^m and a set of templates S^{t_u} with $u = 1, \ldots, N_t$, Warping LCSS identifies all subsequences $S^m_{[W]}$ with $\hat{s}_i = \eta \cdot \text{sim}(S^m_{[W]}, S^{t_u}) > Th_{null}$ as actions $\hat{\alpha}_i$. Th_{null} is the null-class rejection threshold, and multiplying with the normalization factor η assures that the score $\hat{s}_i$ is always in $[0, 1]$. The action type $\hat{A}_i$ is the type of the corresponding template S^{t_u}. If the set of pre-trained templates contains multiple templates for an object category, the system adds observations $\hat{\alpha}_i$ for all actions of this category, even if only one of them has a similarity above the null-class rejection threshold. In this way, the calculated total scores $\bar{s}_{[j],k}$ in LocAFusion allow for a fair comparison.

Overall, the action recognition parameters of the Loc-Action System are the alphabet size K for transformation of raw signals to strings, the null-class rejection threshold Th_{null}, and the penalty factor F_{pen}, which Warping LCSS uses internally to penalize dissimilar symbols in the sequence matching. F_{pen} depends on the scaling of the input signals (here: acceleration in $\frac{m}{s^2}$ and rotation velocity in $\frac{rad}{s}$).

EVALUATION

Dataset

For preliminary analysis of the system's characteristics, we performed ten recordings with volunteers walking and interacting with objects in office and home environments. All recordings consisted of a training run, where the person performed each possible object interaction once. We used the recorded motion data as action templates in Warping LCSS. The training run was followed by the main experiment, during which the person repeatedly performed the same interactions in a self-chosen sequence. Table 3 summarizes the characteristics of the individual recordings. W1-6 were all in the same office building, with similar window and door handles, while E1-4 took place in different private houses. The participants in recordings W1-4 and E1 were the authors of this paper, whereas the participants in W5-6 and E2-4 were neither familiar with action recognition nor indoor tracking from inertial sensors. We purposely tested the system in multiple environments (rather than repeating the experiment in the same environment, as common in activity recognition), because the topology of the environment influences the task complexity and generalization capacity of LocAFusion.

During the experiments, the participants had a GoPro HD2 camera mounted on their head. We annotated the recorded videos by hand with the annotation tool ANVIL[2], identifying a total of 554 object-specific actions (62 `d_open`, 61 `d_close`, 89 `w_open`, 89 `w_close`, 57 `r_open`, 57 `r_close`, 44 `t_on`, 44 `t_off`, 13 `p_water`, 9 `c_drink`, 29 others). In addition to labeled actions, the recordings contain many null-class hand activities, such as arm swing while walking, switching on and off the light, carrying objects, answering phone calls, adjusting the camera, etc. The summed-up walking distances of all experiments was $L = 3802$ m.

Performance Measures

To evaluate the performance of LocAFusion, we compare the output $\bar{\alpha}_i$ of the algorithm with the ground truth labels α_i, and check how the accuracy relates to the location-agnostic action recognition estimate $\hat{\alpha}_i$. The analysis is event-based, i.e., we check for each observed action $\bar{\alpha}_i$ whether a ground truth action α_j with $A_j = \bar{A}_i$ overlaps in time. If no action α_j overlaps, we count $\hat{\alpha}_i$ as an *insertion error*. In a similar way, if LocAFusion does not observe an action $\bar{\alpha}_i$ at the time of a ground-truth action, we consider it a *deletion error*. If a ground-truth and an observed action take place at the same time, but have different type, this is a *substitution error*. Figure 4a depicts the definitions graphically.

[2] `http://www.anvil-software.org/`

The action lists $\{\bar{\alpha}_i\}$ and $\{\alpha_i\}$ define the confusion matrix $\mathrm{CF}(\{\bar{\alpha}_i\}, \{\alpha_i\})$, from which standard multi-class comparison measures such as micro- and macro-averaged F1 scores F_1^{μ} and F_1^{M} can be extracted [17]. We apply the same procedure to object-unspecific and specific actions α_i and $\tilde{\alpha}_i$, and mark the performance measures of the object-specific analysis by adding a tilde (e.g., $\tilde{\tilde{F}}_1^{M}$ is the macro-averaged F1 score for the object-specific LocAFusion outcomes). We also analyze the semantic maps that LocAFusion creates. Here, we use the same measures as for action recognition, but check for location proximity to the ground-truth position of objects instead of time overlap (see Figure 4b).

Due to the randomized initialization of k-means clustering, every execution of Warping LCSS results in different outputs $\hat{\alpha}_i$, and therefore also LocAFusion results $\bar{\alpha}_i$. To account for the probabilistic nature of the system, we report the F1 scores as average values from 20 repeated runs with identical parameter settings.

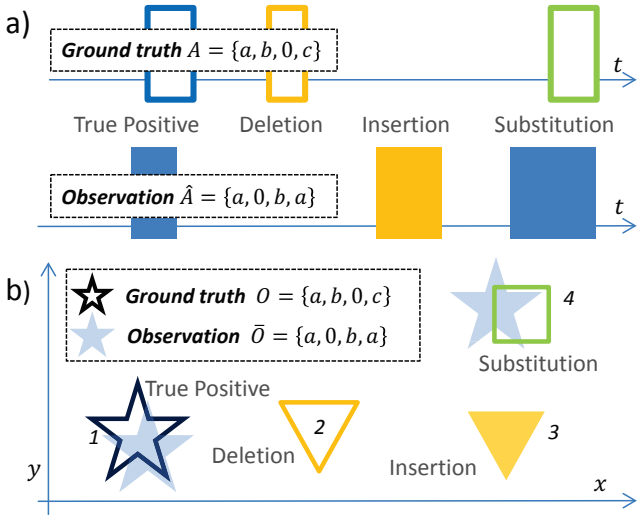

Figure 4. Part a) shows the event classifications for action recognition results, and b) the analogue rules for the semantic map output.

Parameter Optimization

In preliminary experiments, we fixed the parameters of LocAFusion to $Th_{score} = 1$ and $d_0 = 1.5$ m. With this choice of Th_{score}, objects that are only observed once, or multiple times but with very low similarity score, are discarded from the map. The clustering cut-off distance d_0 was set to be smaller than the typical distance between objects in our experimental dataset, but to still allow for minor inaccuracies in the location tracking. We furthermore extracted an optimal set of parameters for the action recognition chain (K, F_{pen}, Th_{null}) through extensive parameters sweeps applied to the data of recordings W1-4. As optimization target we used the weighted, macro-averaged F1 score in recognizing object-specific actions ($\tilde{\tilde{F}}_1^{M}$). The following settings provide close-to-optimal performance for each of the four recordings: $Th_{null} = 0.1$, $F_{pen} = 0.4$, $K = 100$. We continued to use the same parameter set for all further analyses with Warping LCSS and the Loc-Action System. As a consequence, the results we report for experiments W1-4 correspond to

Table 3. Experimental dataset statistics, averaged for 20 repeated executions (mean and standard deviation). L is the distance walked by the user.

	Experiment Characteristics			Warping LCSS Results		LocAFusion Results		
Name	L [m]	N_α	**Description**	$\hat{F}_1^M$	$\tilde{\hat{F}}_1^M$	$\bar{F}_1^M$	$\tilde{\bar{F}}_1^M$	$F_{1,\Theta}^M$
W1	127	18	Open/close 1 door and 2 windows	0.97 ± 0.01	0.68 ± 0.08	0.97 ± 0.01	0.97 ± 0.01	1 ± 0
W2	300	72	Open/close 1 door, 3 drawers, 4 windows and a fridge	0.86 ± 0.03	0.71 ± 0.04	0.87 ± 0.04	0.88 ± 0.03	1 ± 0
W3	389	50	Open/close 2 doors, 2 drawers and 3 windows	0.74 ± 0.4	0.67 ± 0.04	0.74 ± 0.04	0.75 ± 0.00	1 ± 0
W4	388	39	Open/close 1 window, use 3 water taps, water plant, manipulate lever, turn book pages	0.89 ± 0.02	0.66 ± 0.04	0.97 ± 0.02	0.97 ± 0.02	1 ± 0
W5	435	55	Open/close 1 door, 1 drawer and 2 windows, use 1 water tap, water plant, manipulate lever	0.83 ± 0.04	0.75 ± 0.04	0.85 ± 0.03	0.85 ± 0.03	0.91 ± 0.03
W6	549	65	Open/close 2 doors, 2 drawers and 2 windows, use 2 water taps, water plant, drink at table	0.68 ± 0.03	0.66 ± 0.02	0.75 ± 0.03	0.77 ± 0.03	0.91 ± 0.04
E1	361	60	Open/close 3 doors, 2 drawers and 6 windows	0.70 ± 0.04	0.50 ± 0.05	0.70 ± 0.03	0.74 ± 0.03	0.89 ± 0.04
E2	330	66	Open/close 2 doors and 4 windows, use 2 water taps, drink at table	0.67 ± 0.06	0.42 ± 0.07	0.71 ± 0.06	0.73 ± 0.06	0.91 ± 0.02
E3	416	58	Open/close 4 drawers and 3 windows, use 1 water tap	0.71 ± 0.06	0.66 ± 0.06	0.76 ± 0.05	0.77 ± 0.05	1 ± 0
E4	507	71	Open/close 2 doors, 1 drawer and 3 windows, use 2 water taps	0.67 ± 0.03	0.44 ± 0.05	0.74 ± 0.05	0.75 ± 0.05	0.95 ± 0.03

experiment-specific parameter optimization and they may be affected by overfitting, while the analyses for the remaining recordings are with experiment-unspecific parameters.

Performance Analysis

Table 3 summarizes the F1 scores for all 10 recordings, averaged in 20 repeated executions. Due to the varying number of action types and the diverse spatial arrangement of objects, the results from different recordings are not directly comparable. However, we observe that for all experiments, the object-specific (improvement of $8-31\%$) and the object-unspecific F1 scores (improvement of $0-8\%$) increase with LocAFusion. The average F1 scores over all 10 recordings are $\bar{F}_1^M = 0.81$ and $\tilde{\bar{F}}_1^M = 0.82$. For some experiments, the object-specific score $\tilde{\bar{F}}_1^M$ is higher than the object-unspecific score $\bar{F}_1^M$, which is due to the changed weighting and number of action types in macro-averaging. The micro-averaged F1 scores are slightly higher than the reported macro-averaged scores for most of the experiments. While the results in Table 3 are for action templates collected by the same person, results were almost identical in a user-independent analysis of W1-6. Here, we used action templates from one of the six runs for recognizing actions during another recording in the same building. Taking templates from W1-6 to the other environments E1-4 did not work well, as windows, doors and water taps had different handles.

Figure 5 depicts the confusion matrices for E2 and E4, representing typical experiment outcomes. The matrices confirm that LocAFusion corrects most substitution errors between objects, and discards some insertion errors. For obvious reasons, LocAFusion cannot resolve confusions between interactions with the same object (e.g., `d1_open` and `d1_close`). Furthermore, LocAFusion cannot correct LCSS deletion errors (left-most column of the confusion matrix).

For semantic mapping, the mean F1 score over all experiments was $F_{1,\Theta}^M = 0.96$. In average, there were 0.1 object insertions, 0.3 deletions and 0.5 substitution errors per experiment. Most of the substitutions were actually insertions of objects at a place where an object of another object category was. For example, the semantic mapping inserted a water tap and a door at the same place, even though only a door was there. Movies on **`https://vimeo.com/user25933319`** show the semantic maps and the LocAFusion outcomes synchronized with the videos from the head-mounted camera.

Reference Dataset

We also applied LocAFusion to the Opportunity dataset[3] [16]. The four drill-runs of this dataset contain a total of 1485 labeled actions in 17 categories, e.g., `d_open`, `s_toggle`, etc. For LCSS applied to the accelerometer data of a wrist-mounted IMU, [13] report a sample-based F1 score of 0.43, with most of the confusions between the opening and closing of various drawers and doors in the environment. We added mock-up locations to each object interaction in the dataset and applied LocAFusion as proposed in this paper. In our event-based analysis, the macro-averaged scores $\tilde{\hat{F}}_1^M$ were between 0.48 and 0.63 for the four drill-runs with Warping LCSS alone, using a single training template per action. When we added location, $\tilde{\bar{F}}_1^M$ was between 0.72 and 0.80. The mean improvement in the object-specific score was 21%, and for object-unspecific action recognition 13%.

DISCUSSION

Although the evaluation was done in constrained settings, the results indicate that through unsupervised fusion of location

[3] **`http://www.opportunity-project.eu/challengedatasetdownload`**

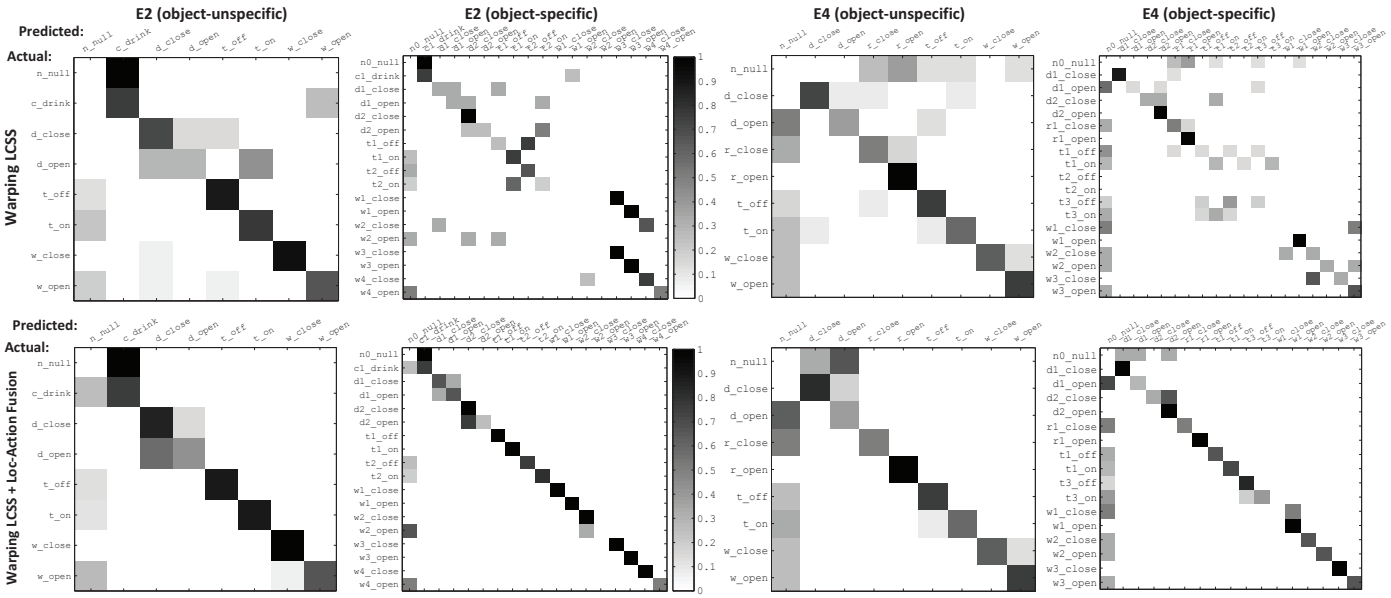

Figure 5. Confusion matrices for E2 and E4 with Warping LCSS and LocAFusion. The darker the square at a position, the higher the percentage of actual actions predicted to be in that category.

and action recognition, accurate semantic maps of the environment can be built. These maps then help to better discriminate object-related actions, leading to overall recognition accuracies above 70% in problems with up to 23 classes. Most of the remaining errors are deletions or confusions between the possible interactions with a single object. A system with pre-trained location-action relationship would also fail at resolving these issues. We can therefore conclude that posterior fusion with location-awareness achieves performances similar to a supervised location-activity recognition system, but at reduced costs for model learning. For example in experiment E2, only a single `w_open` action must be pre-learned (with a window that is similar to `w1-4`, but not necessarily in the target environment), and nevertheless the Loc-Action System can distinguish between all four windows the person opened. This also reduces the computation costs, as Warping LCSS similarities do not need to be calculated for all object-specific action templates, but only for one or few templates per object-unspecific action type.

The main reason for the remaining deletion and substitution errors is at the level of action recognition, where a single training template is often not sufficient to account for the large variance in the way people execute object interactions. The issue may be handled by recording a more extensive set of training templates. However, this comes at the cost of increased deployment effort. In future, we will investigate ways of improving the template set at run-time, again by means of semantic mapping. The imagined location-based adaptive learning system would start with an initial set of action templates, but replace the originals with more representative versions once it observes an action multiple times at the same place. Even a fully unsupervised approach for detection of motion patterns that repeatedly occur at the same location could be developed, e.g., with motif discovery as in [2]. In addition, we may use the state attribute of objects to correct for substitutions between interactions with the same object. Consider for example a situation where template matching finds two actions `d_close` and `d_open` that occur at the same time, with the score $\hat{s}_{\texttt{d\_close}}$ slightly higher than $\hat{s}_{\texttt{d\_open}}$. Normally, we would choose the action $\bar{A}_i = \texttt{d\_close}$, because $\hat{s}_{\texttt{d\_close}} > \hat{s}_{\texttt{d\_open}}$. However, if the last object interaction was already `d_close` with $\hat{s}_{\texttt{d\_close}} \gg \hat{s}_{\texttt{d\_open}}$, the current state of the door is most likely $V_{j,\texttt{d}} = \texttt{closed}$, and consequently $\bar{A}_i = \texttt{d\_open}$ the likelier alternative.

CONCLUSION

We introduced the post-processing algorithm LocAFusion for improving activity recognition through autonomous semantic mapping of the environment, followed by action correction. Thus, we use the accumulated information about what happens at a given place to recognize actions, rather than the current measurements only. In experiments, the method improved the accuracy in detecting object-unspecific hand actions by up to 13% and for object-specific actions by $8-31\%$, depending on the types and diversity of activities performed. Furthermore, LocAFusion constructed semantic maps of the environment that well reflect the arrangement of objects in the environment. In average, the method added less than one incorrect object label per experiment.

While the algorithm is agnostic to the source of the activity and location inputs, we also proposed a specific, fully wearable implementation of the method. The *Loc-Action System* tracks the user location with *ActionSLAM*, performs hand action recognition through *Warping LCSS*, and then combines the outcomes with *LocAFusion*. Prior to deployment, the system only requires a training set with object-unspecific actions, which are not necessarily collected by the target user. As a result, the system is both unobtrusive and easily deployed, without the need for pre-installed tracking infrastructure or the extensive training of location-action models.

In future, we target a joint framework for simultaneous location tracking and action recognition. Rather than constructing the semantic map in post-processing, such a system would continuously build and improve the semantic map with estimations of the object types and their state, and in parallel use the same map for location estimation with SLAM. [5] and [6] applied action recognitions as landmark observations in SLAM, but they only considered activities such as *sitting* and *stair climbing*, which can be recognized with very high accuracy. Further work is required to obtain a fully probabilistic framework that can handle action recognition accuracies in the range of $< 80\%$, as reported for template matching in this paper. This may be achieved through assigning probabilistic type and state attributes to object landmarks in SLAM, and update them by means of a Bayesian filter.

ACKNOWLEDGMENTS

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

REFERENCES

1. Berkhin, P. A survey of clustering data mining techniques. In *Grouping multidimensional data*. Springer, 2006, 25–71.
2. Berlin, E., and Van Laerhoven, K. Detecting leisure activities with dense motif discovery. In *Ubiquitous Computing, 14th ACM Conference on*, ACM (2012), 250–259.
3. Berndt, D. J., and Clifford, J. Using dynamic time warping to find patterns in time series. In *KDD workshop*, vol. 10, Seattle, WA (1994), 359–370.
4. Figo, D., Diniz, P. C., Ferreira, D. R., and Cardoso, J. M. Preprocessing techniques for context recognition from accelerometer data. *Personal and Ubiquitous Computing 14*, 7 (2010), 645–662.
5. Grzonka, S., Karwath, A., Dijoux, F., and Burgard, W. Activity-based estimation of human trajectories. *Robotics, IEEE Transactions on 28*, 1 (2012), 234–245.
6. Hardegger, M., Roggen, D., Mazilu, S., and Tröster, G. Actionslam: Using location-related actions as landmarks in pedestrian slam. In *Indoor Positioning and Indoor Navigation, Third International Conference on*, IEEE (2012), 1–10.
7. Hirschberg, D. S. Algorithms for the longest common subsequence problem. *Journal of the ACM 24*, 4 (1977), 664–675.
8. Hoey, J., Plötz, T., Jackson, D., Monk, A., Pham, C., and Olivier, P. Rapid specification and automated generation of prompting systems to assist people with dementia. *Pervasive and Mobile Computing 7*, 3 (2011), 299–318.
9. Ko, M. H., West, G., Venkatesh, S., and Kumar, M. Using dynamic time warping for online temporal fusion in multisensor systems. *Information Fusion 9*, 3 (2008), 370–388.
10. Liao, L., Fox, D., and Kautz, H. Location-based activity recognition. *Advances in Neural Information Processing Systems 18* (2006), 787.
11. Liao, L., Fox, D., and Kautz, H. Extracting places and activities from GPS traces using hierarchical conditional random fields. *The International Journal of Robotics Research 26*, 1 (2007), 119–134.
12. Lu, C.-H., and Fu, L.-C. Robust location-aware activity recognition using wireless sensor network in an attentive home. *Automation Science and Engineering, IEEE Transactions on 6*, 4 (2009), 598–609.
13. Nguyen-Dinh, L.-V., Roggen, D., Calatroni, A., and Tröster, G. Improving online gesture recognition with template matching methods in accelerometer data. In *Intelligent Systems Design and Applications, 2012 12th International Conference on*, IEEE (2012), 831–836.
14. Nüchter, A., and Hertzberg, J. Towards semantic maps for mobile robots. *Robotics and Autonomous Systems 56*, 11 (2008), 915–926.
15. Pham, C., and Olivier, P. Slice&dice: Recognizing food preparation activities using embedded accelerometers. In *Ambient Intelligence*. Springer, 2009, 34–43.
16. Roggen, D., Calatroni, A., Rossi, M., Holleczek, T., Forster, K., Tröster, G., et al. Collecting complex activity datasets in highly rich networked sensor environments. In *Networked Sensing Systems, Seventh International Conference on*, IEEE (2010), 233–240.
17. Sokolova, M., and Lapalme, G. A systematic analysis of performance measures for classification tasks. *Information Processing & Management 45*, 4 (2009), 427–437.
18. Stiefmeier, T., Ogris, G., Junker, H., Lukowicz, P., and Tröster, G. Combining motion sensors and ultrasonic hands tracking for continuous activity recognition in a maintenance scenario. In *Wearable Computers, 10th International Symposium on*, IEEE (2006), 97–104.
19. Stiefmeier, T., Roggen, D., Ogris, G., Lukowicz, P., and Tröster, G. Wearable activity tracking in car manufacturing. *Pervasive Computing 7*, 2 (2008), 42–50.
20. Wang, L., Gu, T., Tao, X., and Lu, J. A hierarchical approach to real-time activity recognition in body sensor networks. *Pervasive and Mobile Computing 8*, 1 (2012), 115–130.
21. Wilson, D. H., and Atkeson, C. Simultaneous Tracking and Activity Recognition (STAR) using many anonymous, binary sensors. In *Pervasive computing*. Springer, 2005, 62–79.
22. Zheng, V. W., Zheng, Y., Xie, X., and Yang, Q. Collaborative location and activity recommendations with GPS history data. In *World wide web, 19th international conference on*, ACM (2010), 1029–1038.
23. Zinnen, A., Blanke, U., and Schiele, B. An analysis of sensor-oriented vs. model-based activity recognition. In *Wearable Computers. 13th International Symposium on*, IEEE (2009), 93–100.