Chapter 12

Benefits of Dynamically Reconfigurable Activity Recognition in Distributed Sensing Environments

Clemens Lombriser*, Oliver Amft*†, Piero Zappi‡§, Luca Benini‡ and Gerhard Tröster*

* *Wearable Computing Laboratory, ETH Zürich, Switzerland*
† *ACTLab, Signal Processing Systems, TU Eindhoven, The Netherlands*
‡ *Micrel Lab, Università di Bologna, Italy*
§ *Computer Science and Engineering, University of California San Diego, USA*
{*lombriser,amft,troester*}*@ife.ee.ethz.ch*, {*piero.zappi,luca.benini*}*@unibo.it*

The automatic detection of complex human activities in daily life using distributed ambient and on-body sensors is still an open research challenge. A key issue is to construct scalable systems that can capture the large diversity and variety of human activities. Dynamic system reconfiguration is a possible solution to adaptively focus on the current scene and thus reduce recognition complexity.
In this work, we evaluate potential energy savings and performance gains of dynamic reconfiguration in a case study using 28 sensors recording 78 activities performed within four settings. Our results show that reconfiguration improves recognition performance by up to 11.48 %, while reducing energy consumption when turning off unneeded sensors by 74.8 %. The granularity of reconfiguration trades off recognition performance for energy savings.

12.1. Introduction

Activity recognition is widely considered as basic service in smart environments and on-body assistants, where novel ways of interaction and assistive tools are sought. Examples include personal healthcare,[1] safety, and user comfort.[2] Advances in sensing technology and embedded systems support this trend and a substantial amount of research results indicates applicability of various activity recognition approaches. Nevertheless, obtaining satisfactory system performances is often a major challenge.[3] This can be attributed to complexity and variability of human behavior, and further, to persisting limitations in sensing and recognizing large activity catalogs. Research efforts have often considered activity recognition with a focus on individual settings, which allowed to constrain recognition regarding sensors used and the activity catalog needed. While this concept is valid for some applications, it does not scale to broad activity catalogs and the full capabilities needed for today's smart environment visions. It can be expected that reconfiguration of activity recognition systems will allow merging benefits of situation-specific performance and scalability requirements when several settings are involved.

Knowledge extraction in smart environments essentially benefits from distributed sensing and information fusion, such as when using a mix of body-worn, object-integrated, and ambient-installed sensors with embedded processing facilities. Such sensors are generally battery-driven, techniques to reduce energy consumption are thus vital to ensure acceptable system lifetime. Energy can e.g. be saved using local processing and communicating only when needed as opposed to sending continuous data streams.[4] Additionally, in many smart environments monitoring a multitude of user activities, each sensor node might not be required at all times. Consequently, effective energy saving should incorporate knowledge on the actual need for a node to be present and when it is safe to turn off.

This work investigates the benefits of dynamic reconfiguration in adapting activity recognition systems to a specific setting. We deployed a distributed activity sensing and processing architecture that performs local activity event spotting at each sensor node and communicates recognized events only. We integrated this system with a state-based reconfiguration scheme to evaluate adaptation benefits regarding the actually relevant activity catalog and required sensor nodes.

In particular, this paper addresses the following questions:

(1) **How much energy is saved by reconfiguring a distributed recognition system?** Here, we evaluate potential benefits of reconfiguration on energy savings by letting sensors sleep when they are not needed for monitoring activities in the current situation. Since reconfiguring sensor nodes increases communication overhead, we include the communication cost for node adaptations in our analysis.

(2) **How does reconfiguration impact the recognition performance?** We evaluate benefits of reconfiguration on recognition performance. Reconfiguration of sensor nodes was used here to load specific recognition models, adapted to individual situations and their relevant activities. Starting from a baseline that included all activities, the activity catalog was reduced to contain situation-specific activities only.

(3) **Which reconfiguration granularities generate which benefits?** We propose three different reconfiguration granularities relating to the current setting, activity composite, and object used. Subsequently, we compare their benefits regarding energy consumption and recognition performance to the baseline without reconfiguration.

The aim of this work was to quantify potential energy savings and recognition performance increases for the most resource-constraint layer in a activity recognition stack: the sensor nodes that recognize basic activities. This work does not target algorithm optimizations for robust recognition. Instead, we focus our analysis on a particular configuration of sensor nodes and algorithm set used to investigate fundamental reconfiguration benefits.

To evaluate our reconfiguration approach, we analyzed a large dataset comprising 78 atomic activities in four settings. Sensor data from 28 nodes (accelerometers and light sensors) deployed on-body, on tools, and in the infrastructure were used to train a distributed, continuous activity recognition system and analyze reconfiguration effects. Three reconfiguration granularities were explored and compared to the baseline (no reconfiguration): (1) setting-specific (adapting to current situation), (2) composite-specific (adapting to current activity composite), (3) object-specific (adapting to currently used object).

Section 12.2 discusses previous approaches related to reconfigurable activity recognition. Section 12.3 introduces our recognition architecture and terminology used in this work to denote a distributed recognition of human activities. Our architecture is extended in Sec. 12.4 to enable dynamic reconfiguration of the system at different granularities. The implementation of our approach is summarized in Sec. 12.5. Section 12.6 describes our evaluation dataset. Section 12.7 presents the reconfiguration results obtained for all three granularities. Section 12.8 finally discusses our main findings and indicates opportunities for further research.

12.2. Related Work

Various hierarchical abstraction techniques have been considered to capture complex human activities. Nevertheless, those approaches differ in granularity of abstractions and recognition goals. For example, Ryoo and Aggarwal[5] used three layers and context-free grammars to describe image sequences. Kawanaka et al.[6] used a hierarchical architecture of interacting hidden Markov models to represent sequences of activities. Reconfiguration of sensor networks was not specifically addressed in those works. While we utilize a hierarchical abstraction concept in the current work as well, higher-layer probabilistic modeling is not addressed, in favor for investigating fundamental sensor-based energy and recognition benefits of reconfiguration.

Another concept used to address the complexity problem in activity recognition is to incorporate location information as a results filter. For example, Naya et al.[7] used an infrared location estimation system to mask location-dependent activities. In total 13 activities of a typical nursing workflow were classified from body-worn accelerometers with recalls ranging from 14.8–97.4%. Ogris et al.[8] recognized 20 activities for quality inspection in car production. After a first spotting step with high recalls of 79.0%, a masking step was applied in which location and force sensors at the lower arm were used to refine recognition results. This approach changed the performance to 47.8% precision at 70.6% recall in the considered dataset.

Besides spatial information, also temporal relations between subsequent activities can be used to rule out impossible sequences of activities. For example, Murao et al.[9] could improve the recognition of nine leg-based activities by prohibiting impossible transitions between activities, such as a direct change from bicycling to running. Performance changed by 3.99% to 91.74%. A similar approach is known in speech recognition as triphones or context-dependent phone modeling.[10] Phone recognition used the immediate left and right neighboring phones as context to estimate the current one, reducing recognition error rate by 60%. These approaches used additional knowledge to improve recognition performance. Activity models in sensors were typically not completely exchanged in those approaches. However, this could allow to minimise processing and the activity catalog to be recognised.

Energy-efficient activity recognition has been considered in many investigations as a system design parameter. For example, Stäger et al.[11] showed the tradeoff between recognition performance and power consumption at node level by adapting sampling rate and duration, and choosing appropriate features. At the network level, redundancy between sensor nodes was exploited to turn off unneeded sensor nodes: Ghasemzadeh et al.[12] provided bounds for selecting the smallest number of sensor nodes while maintaining service quality. The approach is based on modeling actvitiy class discrimination capabilities of sensor nodes. Zappi et al.[13] demonstrated that clustering sets of active nodes in a gesture recognition setting could extend the network lifetime more than 7 times, while keeping recognition rates above 80 % and more than 4 times for more than 90 %. The work did not address energy saving benefits for different situations, while accounting for reconfiguration-related communications efforts.

Several frameworks have been developed that allow dynamic reconfiguration and could enable adaptations in context recognition algorithms for resource-limited sensor networks. RUNES[14] defines a framework that uses high-level application descriptions and deploys them on top of a run-time kernel in networked sensor nodes. Titan[15] describes context recognition applications as service graphs and distributes them among sensor nodes of wireless sensor networks. SPINE[16] provides a set of libraries for rapid prototyping of health care applications in body sensor networks. Osmani et al.[17] introduced a concept, called "Context Zones", where sensor nodes join a zone if they can contribute with events to the inference engine executed by this zone. These frameworks have been evaluated with respect to their networking performance, but their impact on activity recognition performance has not yet been quantified.

12.3. Distributed activity recognition

Activities are often captured in a hierarchical recognition stack. At its lowest layer, a stack would process raw sensor data to identify *atomic activities*, which are considered basic, non-dividable activity units in the particular recognition stack. Examples include "picking up a book" and "using a screwdriver". Higher layers are frequently used to agglomerate atomic activities into more complex activity composites, representing workflow expressions. As an example, the atomic activities of picking up a glass of water, tilting it, and putting it down may make up the composite "drinking from a glass of water". Recognizing composite activities is conceptually different from recognizing atomic activities, as for composites discrete events are processed instead of streaming sensor data.[18] In this work, we focus on evaluating reconfiguration benefits at the critical atomic activity layer, which directly affects processing at distributed sensor nodes.

In our previous work,[19] we investigated the performance of a distributed activity recognition architecture. In this work, wireless sensor nodes performed a local spotting and identification of activity events from sensor readings and communicated detected events. This concept was shown to considerably reduce data amounts that needed to be communicated through a network, and thus sensor node energy consumption. In this work, we consider this distributed architecture as baseline for our reconfiguration analysis.

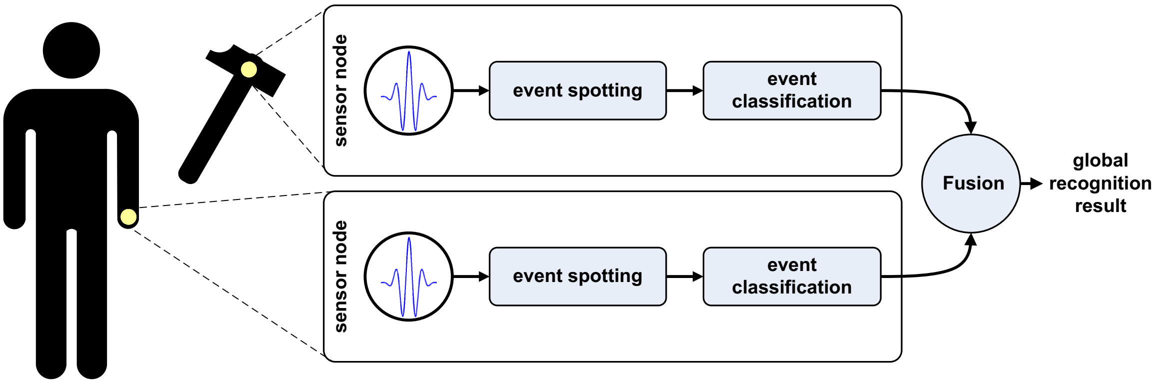

Fig. 12.1. Architecture of our distributed activity recognition system. Each sensor node performs local data acquisition, activity spotting, and classification. Only local detector events are communicated.

The recognition architecture is shown in Fig. 12.1. Each sensor node runs a detector, which recognizes patterns in locally acquired sensor data streams according to training examples of atomic activity events (activity spotting). Detected events are subsequently classified according to the type of atomic activity. Results are communicated within the network to allow further distributed or centralized processing. In this work, we deployed a network fusion scheme to centrally filter erroneously reported events. A recognition architecture to deal with distributed activity event recognition is presented below. The implementation of event spotting, classification, and network fusion is detailed in Sec. 12.5.

12.3.1. *Distributed activity recognition architecture*

In this section we introduce a model to represent distributed activity events at each sensor node (detector). This model will be subsequently extended to include dynamic reconfiguration in Sec. 12.4.

Atomic activities $a_i \in \mathscr{A}$ represent the global detection catalog in our architecture. However, distributed sensor nodes may observe atomic activities differently. For a particular sensor node, several atomic activities could be represented by identical event patterns, e.g. a wrist detector could potentially not distinguish between atomic activities for picking up a tool and returning it. Nevertheless, for a sensor node attached to the tool itself, these atomic activities may exhibit entirely different patterns. Thus, a distributed architecture can discriminate the two atomic activities, while individual sensor nodes may not. To represent these properties, we map atomic activities into detector events.

Each contributing sensor node delivers detector events $\mathscr{E}_{i,j}$ representing local observations of a performed atomic activity. Each detector event represents a set of atomic activities that are locally observed as being identical. The set of all disjunct detector events forms detector set $\mathscr{D}_i$ of a sensor node i:

$$\mathscr{E}_{i,j} = \{a_{i,1}, \ldots, a_{i,n}\} \subseteq \mathscr{A} \tag{12.1}$$

$$\mathscr{D}_i = \{\mathscr{E}_{i,1}, \ldots, \mathscr{E}_{i,n} \mid \forall \mathscr{E}_{i,j}, \mathscr{E}_{i,k} \in \mathscr{D}_i,\ j \neq k:\ \mathscr{E}_{i,j} \cap \mathscr{E}_{i,k} = \emptyset\} \tag{12.2}$$

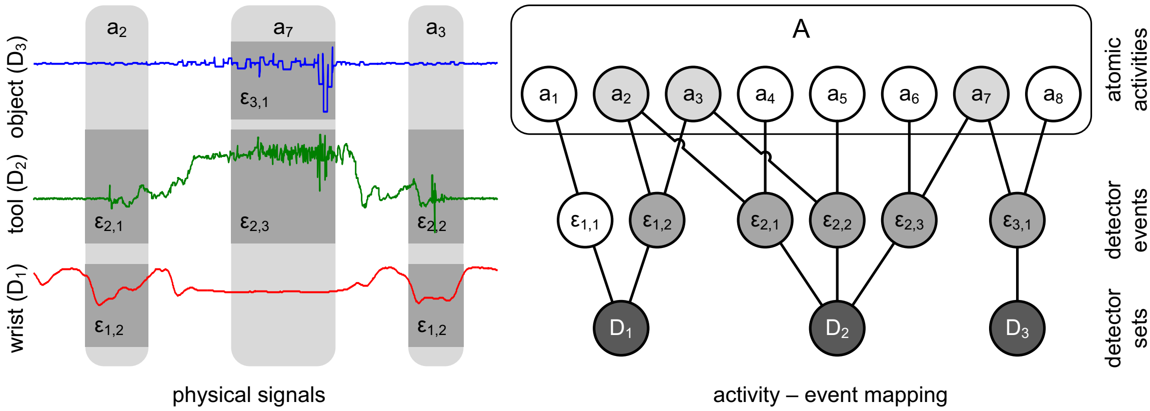

Fig. 12.2. Exemplary illustration of the relationship between atomic activities a_i, detector events $\mathscr{E}_{i,j}$, and detector sets $\mathscr{D}_i$ in an object manipulation task. A tool is being picked up (a_2), used on an object (a_7), and placed down again (a_3). Left: acceleration signals of involved sensor nodes in which detector events are recognized. Right: mapping of atomic activities to detector events.

The relationship between atomic activities and detector events is illustrated in Fig. 12.2. In this example, the atomic activities "picking up a tool" (a_2), "using it to manipulate an object" (a_7), and "placing the tool down" (a_3) are visualized. Three sensor nodes with the detector sets D_1 to D_3 are used, each operating on their locally acquired signals using an individual mapping of atomic activities to detector events. In this example, the sensor node implementing D_1 (wrist) does not distinguish between the atomic activities "picking up" (a_2) and "placing down" (a_3). Thus, both atomic activities are mapped to the same detector event $\mathscr{E}_{1,2}$, which could be understood as "reaching down". In contrast, the sensor node implementing D_2 (tool) observes different patterns for a_2 and a_3. It issues the detector events $\mathscr{E}_{2,1}$ and $\mathscr{E}_{2,2}$.

The mapping function between atomic activities and detector events is determined during training of the distributed activity recognition architecture. While it would be conceivable to use an automatic procedure for this purpose, we performed this step using expert knowledge of location and function of each sensor node. This was done in an attempt to minimize potential error sources for our reconfiguration analysis.

Based on atomic activities we can describe activity composites $\mathscr{C}_n$ as a sequence of not strictly ordered m atomic activities:

$$\mathscr{C}_n = \{a_{n,1}, \ldots, a_{n,m}\} \in \mathscr{A}^m$$

For the example illustrated in Fig. 12.2, a composite activity "manipulating task" could be defined as $\mathscr{C}_n = \{a_2, a_7, a_3\}$. The recognition of such composites was discussed in our previous work.[19] In this work, we use the composite description to denote reconfiguration granularities, as detailed in Sec. 12.4 below.

12.4. Dynamic reconfiguration of activity models

Our reconfiguration approach uses additional domain knowledge to dynamically adapt a distributed recognition system. In particular, we constrained the set of atomic activities and let unused sensor nodes sleep depending on the system state. Trigger events are used to switch between configurations.

12.4.1. *Reconfiguration concept*

To track system states for reconfiguration we used a state machine to model a set of recognition states S as shown in Fig. 12.3. Each state $s_i \in S$ of this state machine describes its own activity catalog $\mathscr{A}_s(s_i) \subseteq \mathscr{A}$ that is relevant when the state is active. Transitions $\delta : S \times \mathscr{A} \to S$ between states are activated by trigger activities $\mathscr{A}_t(s_i) \subset \mathscr{A}_s(s_i)$. In the example shown in Fig. 12.3, activity a_3 will trigger a transition from state s_2 to state s_3. When state s_3 is entered, the set of relevant atomic activities is changed to $\mathscr{A}_s(s_3)$ and all sensor nodes are reconfigured for this state. Detector events $\mathscr{E}_{i,j}$ that do not contain atomic activities of $\mathscr{A}_s(s_3)$ are removed from detector set $\mathscr{D}_i$. Subsequently, sensor nodes with empty detector sets will enter a low-power state to save energy until they are needed again.

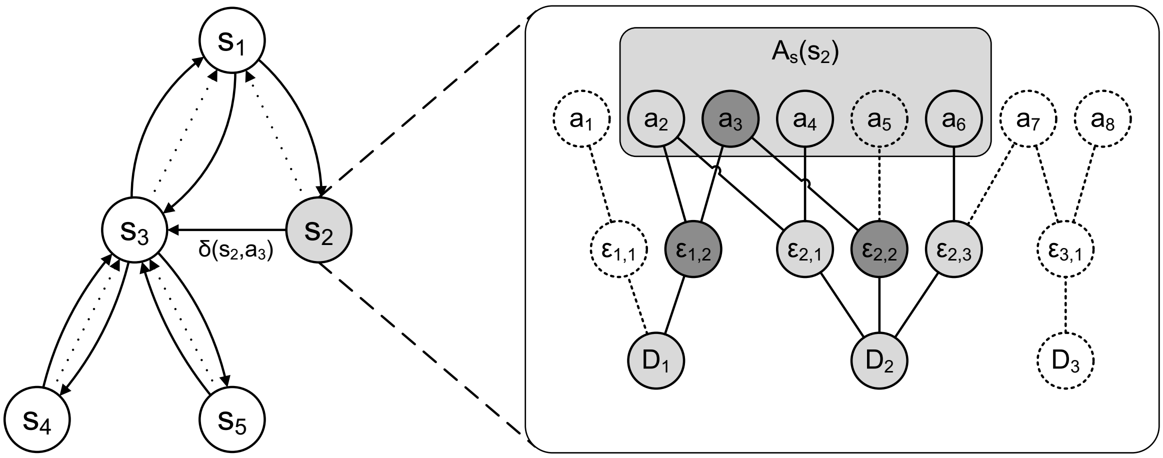

Fig. 12.3. Illustration of the reconfiguration approach. In each system state a subset of relevant atomic activities $A_s(s_i)$ from the total set $\mathscr{A}$ is defined. Here, state s_2, includes $\mathscr{A}_s(s_2) = \{a_2, a_3, a_4, a_6\}$. Thus, events $\mathscr{E}_{1,1}$ and $\mathscr{E}_{3,1}$ will never occur. Detector $\mathscr{D}_3$ does not have events to report and can be turned off. Trigger activity a_3 will cause a state machine transition to state s_3 and subsequent reconfigurations.

Reconfiguration states can be organized in a hierarchy, where each layer allows further constraints to the relevant activity set, resulting in a finer reconfiguration granularity. In Fig. 12.3, reconfiguration state s_1 comprises all atomic activities of a catalog $\mathscr{A}_s(s_1) = \mathscr{A}$. State s_2 however reduces the activity catalog to $\mathscr{A}_s(s_2) = \{a_2, a_3, a_4, a_6\}$. This modeling could, e.g. be used for location-dependent activity sets: when a user enters a kitchen, it becomes unlikely that he will pick a book from the living room shelf.

In Fig. 12.3, state s_3 provides options to further restrict an activity catalog by transition to s_4 or s_5. This may be used for object-specific reconfigurations, e.g. when detecting that a user picked up a pan and thus cannot use other kitchen objects at the same time with this hand. In our evaluation, we investigated effects of hierarchy depth for such state machines, hence the effect of different reconfiguration granularities.

12.4.2. *Reconfiguration granularities*

To evaluate benefits of this approach in detail, three different reconfiguration granularities were considered in addition to a baseline without reconfiguration. During reconfiguration, only parameters such as activity recognition models and sensor node power mode were changed, no modifications to the architecture or algorithm types was made to enable direct comparison.

The three reconfiguration granularities in addition to baseline describe the hierarchy levels of our reconfiguration state machine. Figure 12.4 shows how those reconfigurations take place during runtime.

- **Baseline.** The baseline represents a recognition that does not reconfigure and thus includes all atomic activities within its activity catalog. It is the most general state which is assumed when no opportunity for a reduction of atomic activities is available.
- **Setting-specific reconfiguration.** Here, the activity catalog is restricted to a specific setting, such as a room or place. The complexity per reconfiguration state corresponds here to most activity recognition results found in literature, e.g.[7,8,20]
- **Composite-specific reconfiguration.** Activity composites represent individual steps within a workflow. An example could be the steps involved in following a recipe during cooking. Reconfiguration restricts the activity catalog to a set of atomic activities which are required to fulfill a certain step.
- **Object-specific reconfiguration.** A further refinement is the reconfiguration following interactions with individual objects. This assumes that a person is only using that object as a tool while it is being held. Trigger activities may be picking up and placing down an object, while the activity catalog contains atomic activities related to the tool.

12.5. Implementation of the activity recognition chain

Our distributed activity recognition architecture builds on sensor nodes that operate as activity event detectors on locally sampled continuous sensor data streams. In this section we detail spotting and classification algorithms that were employed for sensor nodes and fusion algorithms.

12.5.1. *Event recognition at distributed sensor nodes*

Activity event spotting was performed using the Feature Similarity Search (FSS) algorithm.[21,22] The algorithm uses continuous sensor streams to spot events that are embedded in arbitrary data and can cope with variable-length motions. FSS was applied for each detector event type separately. A subsequent filtering stage was used to derive detector events for each sensor node. We briefly outline key elements of feature extraction, event spotting, and selection below.

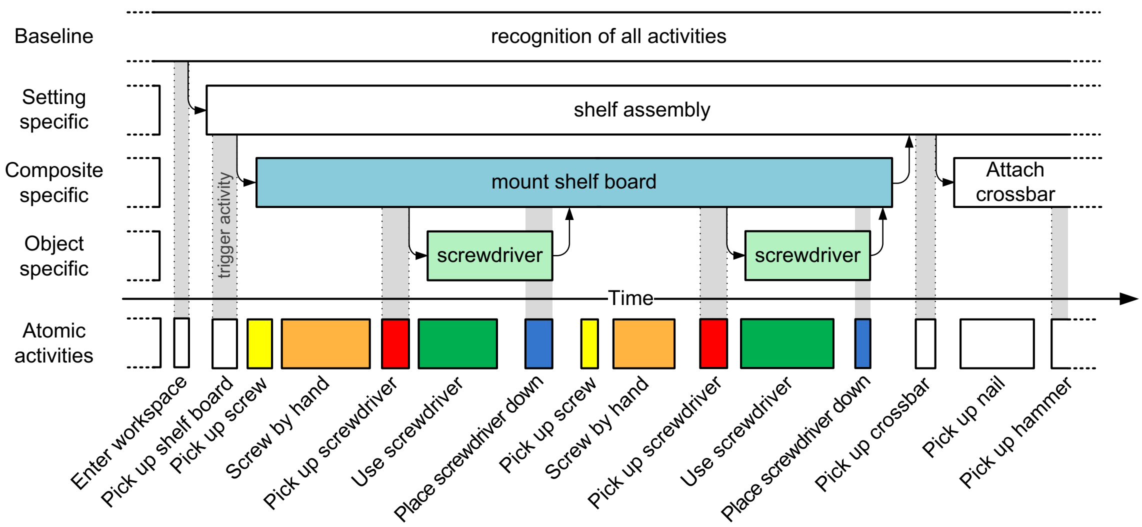

Fig. 12.4. Illustration of a reconfiguration sequence at the finest granularity considered in this work. The atomic activities (lowest line) are recognized within reconfiguration states of different granularities (shown above). A finer reconfiguration granularity allows to reduce the number of atomic activities that needs to be described by pattern models.

12.5.1.1. *Feature processing and selection*

A general set of time-domain features were computed to model event data patterns of all 3D-acceleration sensors in the network. These features included sums and absolute sums; first and second deviations; minimum, average, and maximum amplitudes, and event duration. Besides acceleration sensors, we used light sensors, to e.g. detect open drawers. Activity events for these sensors are limited to light-on/off transitions, which required less features to model.

The feature set was derived from three evenly distributed sections of sensor data and the entire event instance. To select relevant features, a Mann-Whitney-Wilcoxon test was used to compare event instances to embedding data of a training set. This ranking was refined by analyzing correlations among all features. A set of 20 features was selected according to a method yielding highest rank and minimum correlation scores.[23]

12.5.1.2. *Event spotting using feature similarity search (FSS)*

The FSS algorithm consists of a signal pattern modeling (training) and a search stage. A separate training dataset was used to determine FSS model parameters and select a feature set. A validation set was subsequently used to determine performance results as described in Sec. 12.5.4 below.

To search continuous data, we used an equidistant data segmentation of 0.25 Hz. This setting provided sufficient resolution for all motions. The FSS algorithm operation can be summarized as follows:[21] For each segmentation point, a variable-sized window of previously received data is analyzed. The features in this window are compared to a trained model and the Euclidean distance d_S to this model is computed. A distance threshold

θ_d is used to determine sections that are considered as spotting result (retrieved items). Each retrieved event is associated with a model confidence derived by mapping d_S to an a-posteriori confidence with respect to the training model. Settings for threshold and variable-sized window bounds were determined during training.

12.5.1.3. *Event filtering and classification of sensor nodes*

For each detector event, an individual FSS stage was employed and returned its results independently. This allows the algorithm to scale starting from one detector event. Nevertheless, for each sensor node, only one event can occur at each point in time. Thus, a filtering is applied to generate non-overlapping events by selecting events with highest confidence.[21] For this purpose, a sliding window is maintained to capture temporal collisions between previously retrieved events and new ones.

This local event detection result of each sensor node is forwarded to the network for further processing. In this work, we employ a central fusion of all detector results to analyze reconfiguration.

12.5.2. *Network fusion of distributed detector events*

In our architecture, all detector events communicated by sensor nodes were delivered to a central network fusion. In this final event processing step, we aimed at identifying individual atomic activities from detector events reported by all sensor nodes as well as to correct detection errors.

To implement this behavior, we used a sliding window to capture all concurrently reported detector events for a particular atomic activity and applied a sum rule fusion using reported detector event confidences.[24] When the subject performs atomic activity a_3, sensor node i may, e.g., report detection event $\mathscr{E}_{i,1} = \{a_1, a_3\}$ and sensor node j detector event $\mathscr{E}_{j,3} = \{a_3, a_5\}$. Our fusion then created a fusion event $\hat{a} = \mathscr{E}_{i,1} \cap \mathscr{E}_{j,3} = \{a_3\}$ with the summed confidence of both detector events. An additional filtering step generates non-overlapping fusion events by selecting fusion events with highest confidence.

The resulting fusion events represent the global recognition result and the system's output.

12.5.3. *Architecture and reconfiguration complexity metrics*

Several metrics are relevant to describe architecture and reconfiguration complexity for the recognition task. They can be derived from our architecture model.

The total number of detector nodes $|\mathscr{D}|$ determines the overall recognition system size. The cardinality of individual detector event sets $|\mathscr{D}_i|$ relates to processing requirements imposed on each detector node. Measures that describe frequency and granularity of reconfigurations, are number of recognition states $|S|$ and size of atomic activity set in a state $|\mathscr{A}_s|$.

An additional relevant metric is the average size of detector event sets $\overline{|\mathscr{E}_{i,j}|}$, describing how specific detector events model atomic activities. Thus, a high number for $\overline{|\mathscr{E}_{i,j}|}$ indicates that a detector is capable to identify abstract events describing a group of atomic activities only. To identify the actual atomic activity in this case, an activity recognition system requires the presence of other detectors providing complementary detector events. A low number for $\overline{|\mathscr{E}_{i,j}|}$ indicates that the detector is capable to identify specific atomic activities. Similarly, the average size of a fusion event $\overline{|\hat{a}_i|}$ is an indicator for the specificity of a fusion result. It represents the number of atomic activities that cannot be distinguished by the whole sensor network.

12.5.4. *Performance evaluation*

A ten-fold cross-validation was used to determine training and validation sets for activity spotting and classification of sensor nodes, thus to avoid overfitting. Activity recognition performance were analyzed using the information retrieval metrics *precision* and *recall*, and an event alignment check with a jitter of 0.5.[21]

Figure 12.5 shows an example precision-recall performance graph derived by sweeping a threshold on retrieved event confidence values, which illustrates detector result quality. Best performance is found towards high precision and recall. In this example, the performance of a sensor node attached at the left knee is shown for four detector events. As Fig. 12.5 indicates, the node performs well for all except the "picking up" detector event. This low performance can be attributed to the fact that some of the atomic activities mapped to this detector event could not be satisfactorily modeled by the applied recognition procedures. In this particular case, the detector event included a large set of 22 variable atomic activities in which legs were bend to pick up something from the ground. Nonetheless, we consider the overall performance of all detectors as adequate to explore reconfiguration in the considered complex activity dataset.

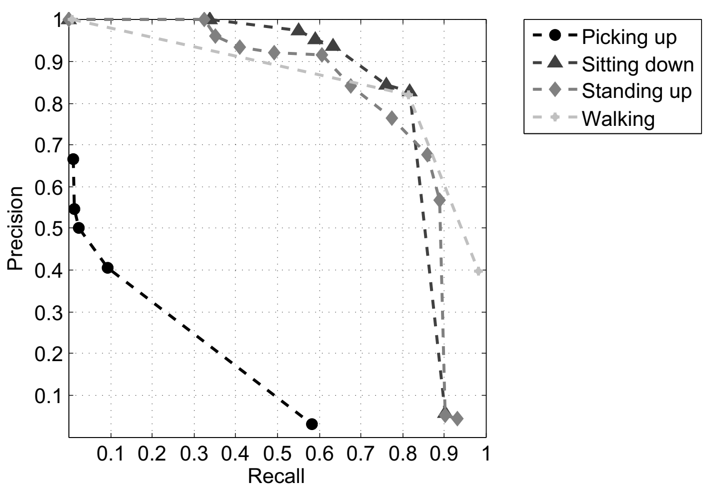

Fig. 12.5. Example recognition performance graph for a sensor node at the left knee sensor. The graph was derived by sweeping a threshold on retrieved event confidence values for four detector event types. Detector event "picking up" is not sufficiently modeled representing a typical issue in activity spotting tasks.

Table 12.1. Settings, recordings sessions, and composite activities of the evaluation dataset. All sessions included activities of the subject entering and leaving the setting, e.g. a room. The dataset includes 18 composite and 78 atomic activities. $|\mathscr{A}_s|$ denotes the size of an atomic activity set.

Setting	Session	Composite Activities	$\lvert\mathscr{A}_s\rvert$
Kitchen	Prepare dinner	Heat water, add soup, cook soup, slice bread, use computer, prepare table, eat, cleaning up dishes	44
Study	Relaxing	Selecting a book from the shelf, reading, Returning the book to the shelf	12
	Working	Use the computer, drink a glass of water, selecting a book, reading, writing, returning the book	21
Assembling furniture	Assembling shelf	getting the tools, mounting middle board, mounting upper board, mounting lower board, returning tools	14
	Attaching crossbar	getting tools, mounting crossbar, hammering in a nail, returning tools	12
Distractions		scratching the head, using the phone, tying shoes, coughing	0

In this evaluation, we have chosen the precision-recall point maximizing the F_1 score (harmonic mean of precision and recall) as best operation point for the further analysis of reconfiguration.

12.6. Evaluation dataset

To illustrate the benefits of reconfiguration, we analyzed a complex activity dataset with several settings, which could naturally benefit from a set of dedicated activities. These settings are often found in rooms or localized environments of a specific purpose, a work plan that has to be followed, etc.

We have chosen a smart environment dataset consisting of four real-world settings and six sessions involving different activities, such as working and relaxing in a living room. In total 18 composite activities and 78 atomic activities were recorded from 28 sensors that were worn on-body, attached to tools, and embedded into the environment. In total 120 recordings were made with 4615 atomic activity instances performed by two subjects.[25]

Table 12.1 provides an overview on our dataset, including settings, sessions, and composite activities. The sessions involved cooking a soup in the kitchen, assembling a shelf with three boards and attaching a metal crossbar to it, working on a desk reading, writing, and using the computer (two sets), and performing arbitrary activities. The latter session served as embedding data for activity spotting under realistic conditions.[26] Thus, these activities were not intended to be recognized, but to evaluate robustness of recognition algorithms. In total, 196 arbitrary activity instances were recorded and annotated. Separate sections of these activities were added to training and validation sets.

12.6.1. *Experimental procedure*

Two subjects were asked to wear 3D-accelerometers at both wrists and at the right thigh. Bend sensors were used to monitor finger extension at the right hand. Additional accelerometers were placed on 12 objects and tools that subjects interacted with. An ad-

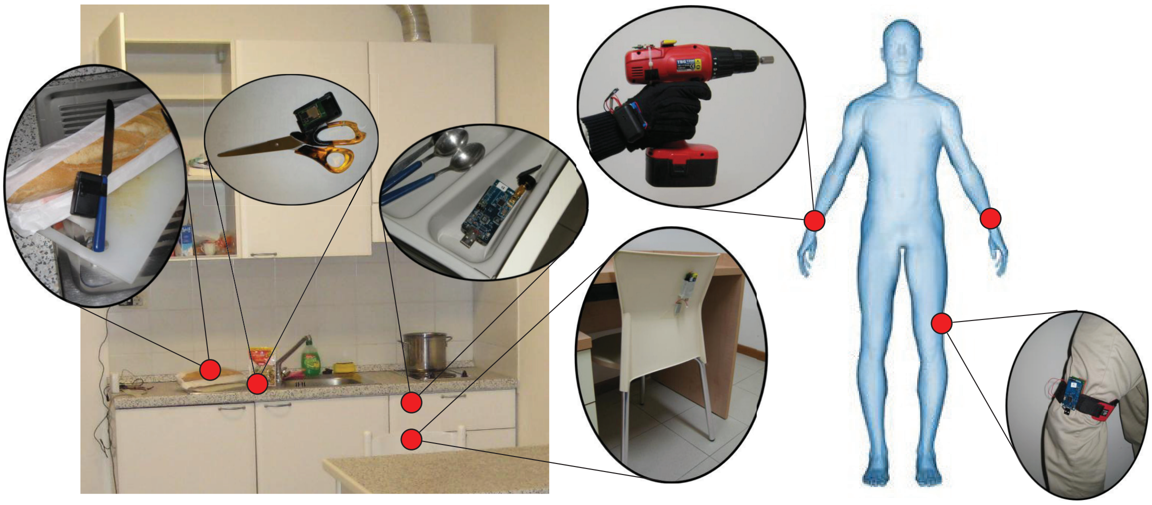

Fig. 12.6. Illustration of sensor selection used during the dataset recordings. The body-worn and object-integrated sensors recorded 3D-acceleration, while drawers and cupboards were equipped with light sensors to monitor opening and closing activities.

ditional 8 light sensors were placed into drawers and cupboards to monitor when they were used by subjects. Work on the computer was sensed by counting keys pressed and mouse movements using a background application. A pyroelectric infrared (PIR) motion sensor was deployed to detect when subjects entered and left a room.

In each recording, subjects entered a room, performed all composite activities in a scripted sequence and left the room again. While activity composites were scripted, subjects had the freedom to perform individual atomic activities in their habitual style. As the aim of this work was to identify reconfiguration benefits for sensor nodes which recognize independent atomic activities, scripting activity composites did not influence performance results.

Each session was recorded from two subjects performing 10 iterations each. The activities were annotated during the experiment by an observer and manually refined in a post-processing step to serve as ground truth. All recordings were videotaped to support later analysis.

12.6.2. *Sensor node complexity*

Table 12.2 shows complexity metrics for all sensor nodes $|\mathscr{D}_i|$, i.e. the number of events to be recognized by each detector. Highest $|\mathscr{D}_i|$ can be observed for the right and dominant hand of all subjects. Almost all performed activities involve this hand.

Objects and tools with attached sensor nodes typically contributed to the recognition task with events directly related to them, e.g. being picked up, used, and placed down. Ambient installed sensors, such as light sensors in drawers and cupboards only observed whether the furniture was used.

Table 12.2. Complexity metrics $|D_i|$ for all sensor nodes considered in the dataset. Most detector events were addressed by body-worn sensors, especially the dominant right hand. Objects and tools typically provided three activities: "picking up", "using", and "placing down". Drawers and cupboard sensors monitored "open" and "close".

Category	Position	$\lvert\mathscr{D}_i\rvert$	Category	Position	$\lvert\mathscr{D}_i\rvert$
Body worn	right hand	35	Infrastructure	Pyroelectric infrared	2
(Acceleration)	left hand	5		Computer mouse	1
	Left knee	4		Computer keyboard	1
Tools	Hammer	3	Furniture	Shelf board	3
(Acceleration)	Screwdriver	3	(Acceleration)	Shelf leg	1
	Scissors	3		Chair	1
	Knife	3	Furniture	Dish cupboard	2
	Book 1	3	(Light)	Cutlery drawer	2
	Book 2	4		Garbage	2
	Phone	0		Pot drawer	2
	Stirring spoon	3		Food cupboard	2
	Drill	3		Tool drawer	2
	Small wrench	3		Desk drawer	2
	Big wrench	3			
	pen	3	Total		92

Table 12.3. Complexity measures for all four reconfiguration granularity levels. The number of states of the state machine, the average number of atomic activities per detector event $\overline{|\mathscr{E}_{i,j}|}$, the total number of reconfigurations at runtime, and the average number of sensors $\overline{|D|}$ is shown.

Granularity	states	$\overline{\lvert\mathscr{E}_j^i\rvert}$	Reconfigurations	$\overline{\lvert D\rvert}$
Baseline	1	1.58	1	26
Setting specific	5	1.36	112	12.7
Composite specific	28	1.21	595	7.7
Object specific	16	1.23	245	7.6

12.7. Results

Table 12.3 summarizes the complexity measures of our reconfiguration approaches. It shows the number of states that are added to the state machine with each level of reconfiguration granularity. Furthermore, the additional number of reconfigurations executed within the complete dataset is indicated. It is important to note that with increasing granularity, the number of relevant sensors decreases as well as the number of atomic activities per detector event. This shows that the recognition becomes more specific to a situation.

12.7.1. *Baseline results*

For the baseline activity recognition, the complete dataset was used to create training and validation sets. Overall performance of all detector nodes amounted to 50.46% recall and 40.90% precision. After fusion, these recognition results were improved to 56.45% recall and 61.67% precision. These results show that recognition of 92 detector events at baseline is challenging.

12.7.2. *Setting-specific results*

To investigate reconfiguration, the dataset was split into a setting specific recognition. Our recognition architecture was therefore trained on data relevant for one setting only. Table 12.1 includes the number of activities that needed to be recognized in each setting in the rightmost column. A total of 78 initial atomic activities at baseline was reduced to 44 (56%) for kitchen, 12 (15%) and 21 (27%) for study, and 14 (18%) and 12 (15%) for assembly respectively.

An additional effect of this setup was that the specificity of fusion events $\overline{|\hat{a}_i|}$ reduced from an average of 1.20 atomic activities per fusion event at baseline, to 1.07 for setting-specific reconfiguration. Recognition accuracy improved to 61.80% and 64.19% for recall and precision, which corresponds to a +6.83% increase in the F_1 score compared to baseline.

As trigger for entering a setting-specific state we selected a PIR sensor which recognized a person entering or leaving the setting. Upon entry, our activity recognition system loaded a setting-specific model and switched back on exiting this setting. For our evaluation, we used ground truth events to determine trigger points, such as to evaluate the gain on perfect triggering. Recall and precision values however include the actually recognized PIR results, which are 92.77% precision and 56.25% recall.

12.7.3. *Composite-specific results*

To further reduce atomic activities, reconfiguration was made with regard to composite activities. In this setup, we have not further restricted the dataset for training detectors, as there were not enough instances to reliably train event spotters.

Again we used perfect triggers at beginning and end of each composite activity, identified by first and last activity belonging to each composite activity. The presented results should therefore be considered as a maximum gain that can be achieved through composite-specific reconfiguration.

The activity recognition performance resulted in 68.12% recall and 63.46% precision, an improvement of +11.48% compared to baseline. This corresponds to a +4.35% higher F_1 score than obtained for our setting-specific evaluation. The average number of atomic activities per fusion event $\overline{|\hat{a}_i|}$ increased to 1.08. This increase is due to the fact that fewer erroneous events were reported. This result shows that recall could be improved while maintaining precision in a composite-specific setup.

12.7.4. *Object-specific results*

For the finest reconfiguration granularity, trigger activities were chosen to be atomic activities, such as picking up and placing down individual tools, or opening and closing drawers.

This choice further reduced the average duration of a reconfiguration state from 48.15 seconds and 8.64 atomic activities in the composite-specific setup, to 13.46 seconds and 7.27 atomic activities. Detector events were at 1.12 atomic activities per detector event less specific, but the number of active sensor nodes was at an average of 7.4 the smallest compared to other reconfiguration setups.

As shown in Fig. 12.4, this setup used the full depth of our reconfiguration state machine. The system reconfigures for setting specific, composite specific, and for object interactions. Thus, in each state fewer activities need to be recognized. Using this fine reconfiguration granularity resulted in a recognition performance of 64.12% for recall and 62.38% for precision. This is an improvement of +7.28% compared to the baseline, but decrease of -3.76% compared to a composite-specific reconfiguration.

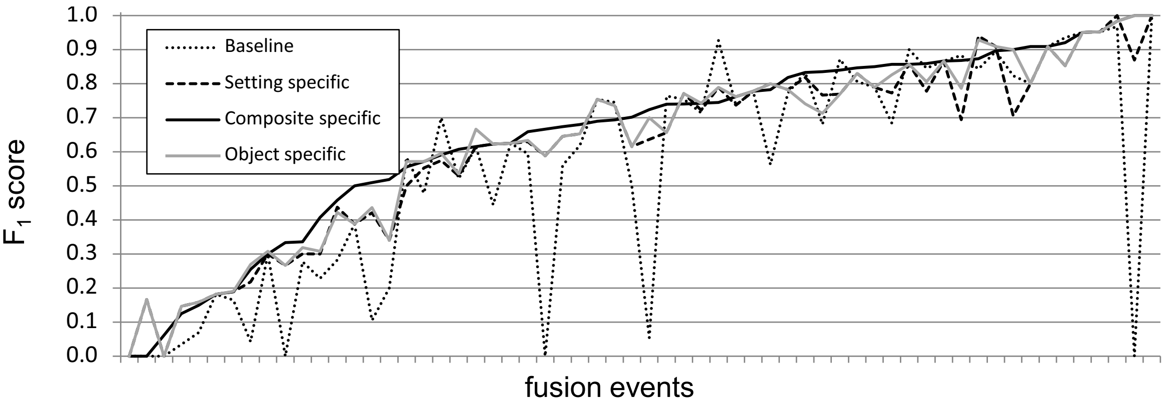

Fig. 12.7. Summary of fusion event performances (F_1 scores) for all reconfiguration granularities. Fusion events are ordered according to increasing composite-specific reconfiguration performance.

Figure 12.7 shows the recognition performance of all fusion events and all considered reconfiguration granularities. The F_1 scores are ordered with regard to increasing composite-specific recognition performance. Table 12.4 summarizes these performance figures for average recall, precision, and F_1 score weighted by their respective event occurrence.

The average unweighted F_1 scores for all fusion events and reconfiguration granularities were [0.581, 0.645, 0.679, 0.659] for baseline, setting-specific, composite-specific, and object-specific reconfiguration, respectively. For the best performing composite specific reconfiguration, 37 % of all fusion events were recognized with a F_1 score over 0.8, and 78% had a F_1 score larger than 0.5. A total of 80 % of these events performed better or at least as well as baseline.

Fig. 12.7 shows clear performance benefits of the reconfiguration granularities compared to a baseline performance. The baseline incurred several low performing events that were clearly visible. These events could not be adequately modeled when considering the complete dataset. A particular example, accounting for the right-most performance drop is the atomic activity "filling pot with water" which was derived from wrist sensor data only. This activity exhibited signal patterns of arm postures similar to those when holding boards during shelf assembly. Holding a shelf board was part of our embedding data, that was not modeled, however made recognizing "filling pot with water" difficult. Using our reconfiguration, this source of confusion was omitted as the involved atomic activities fall

Table 12.4. Comparison of recognition performances for different reconfiguration granularities. Besides precision and recall, the F_1 score and the average fusion event size $\overline{|\hat{a}_i|}$ are shown. Standby time and energy cost determined cost benefit of reconfiguration and mark a trade-off between recognition performance and power consumption.

Reconfiguration detail	$\overline{\lvert\hat{a}_i\rvert}$	Performance [%]			Standby	Energy
		Recall	Precision	F_1	[%]	[J]
Baseline	1.20	56.45	61.67	.589	0.0	24'140
Setting specific fusion	1.13	64.19	61.80	.630	47.0	14'143
Composite specific fusion	1.08	68.12	63.46	.657	74.8	8'889
Object specific fusion	1.12	64.12	62.38	.632	84.5	6'168

into different reconfiguration states, leading to improved recognition performances.

12.7.5. *Costs of reconfiguration*

Each time a reconfiguration is performed, all sensor nodes need to be reconfigured. Data required for this reconfiguration includes information about sampling rate of sensors, spotting window sizes, settings for features, spotting models, and fusion thresholds.

An average configuration for our activity recognition approach includes 25 parameters. Encoded with 16-bit values, this configuration data amounts to 50 bytes. This data is transmitted via a wireless link using a TI CC2420 transceiver, which consumes an average of 135 nJ/bit.[27] For each reconfiguration, sensors get one message, requiring an approximate energy consumption of 1.7 J per reconfiguration.

We can derive the total energy costs for reconfiguration by considering that 1 reconfiguration is required for baseline, 112 reconfigurations for setting-specific, 595 for composite-specific, and 245 for object-specific reconfiguration setups. For an overview, we computed the average time sensors can be turned off as well, which is zero for baseline, 46.5%, 74.2%, and 84.4 % of the total time in all other settings. In our experiments, we used the widely available TelosB sensor nodes.[28] They have an average standby power consumption of 100 μW and a transceiver consumption of 40 mW. Assuming a 10% duty-cycle for listening in standby state, results in a idle-waiting power consumption of 4 mW. We can sum up the values to obtain a recognition performance vs. power consumption trade-off. The resulting energy consumption values are 23.21 kJ, 13.68 kJ, 8.53 kJ, 5.90 kJ across for baseline and all reconfiguration setups in our 6.56 hours dataset.

The energy required for reconfiguration itself amounted to 157.3 J, 835.4 J, 344.0 J. When configurations are cached within sensor nodes, the reconfiguration costs can be reduced to 1 broadcast message per reconfiguration. This would reduce reconfiguration costs to 6.0 J, 32.1 J, and 13.24 J, which includes 1 transmission of each states' configuration, and only 1 broadcast reconfiguration message for subsequent appearances of the reconfiguration state. The main benefit of using cached reconfiguration is in the reduced reconfiguration time that a context switch requires.

The context switch time is another cost of reconfiguration. From the time when a trigger activity is detected until the system is reconfigured and a feature window is filled, no detector events could be obtained. Thus, a short activity could not be recognized during reconfiguration if its falls into a sensor network reconfiguration. The exact time depends on the reconfiguration implementation. E.g., for the Titan framework[15] a reconfiguration time of 0.9 s for six sensor nodes was needed. In our dataset, this is short enough to fit between any two consecutive activities and thus allowed to neglected this effect from further investigation. In other scenarios however, it may be essential that reconfigurations are not triggered during successive activities, which exhibit short performance durations.

12.8. Discussion

Our results show that using reconfiguration is beneficial for distributed recognition systems running on wireless sensor nodes. Reconfiguration can be seen as focusing on currently relevant activities, thus a system can make better use of the limited resources on sensor nodes. Our results confirm that reconfiguration can reduce energy costs and improve recognition performance of human activities. Nevertheless, there are trade-offs to be considered.

Setting-specific training improved the F_1 score (+6.83 %) by reducing the number of activities to be recognized compared a baseline configuration. Moreover, our results show that reconfiguration can be performed at increasingly finer granularity as long as the remaining activity set allows to capture all possible activities at a particular moment and robust models can be constructed. To this end, our composite-specific results show further improvement in recognition performance (+4.35 %) compared to setting-specific results. In contrast, our object-specific results dropped by -3.76 % compared to composite-specific ones, indicating limits in reducing a relevant activity set. We attribute this decrease to reduced information available during the network fusion step. Consequently, errors are left unfiltered that otherwise were removed by competing detector events.

One example illustrating this effect is a wrist sensor reporting many false positives for "screwing". During reconfiguration to the composite activity "mount lower shelf board", a knee sensor may report whether the knee is "bent down" or not. The fusion step may use this information to cancel out false positives from the wrist. During object-specific reconfiguration however, the seemingly irrelevant knee sensor is turned off, and a fusion could not use such information to refine results. While keeping too many sensors may introduce confusing information, a too aggressive limitation of involved sensors can omit subtle information content.

It can be hypothesized that alternate recognition methods may produce different trade-offs points. Any such method however needs to cope with sparse and multi-activity situations. Activity sparseness is limited by the natural requirement to capture all possible activities at a particular moment.

Our results confirm that reconfiguration enables energy saving by turning off sensors when they are not needed. In this way the total energy usage of our distributed system was reduced by up to -84.5 %. In contrast to recognition performance changes, fine-grained reconfiguration further decreases energy costs, as costs for wirelessly sending reconfiguration data amounted to a maximum of 9.79 % of the total energy spent.

Whether reconfiguration can be successfully applied is application-dependent. Settings

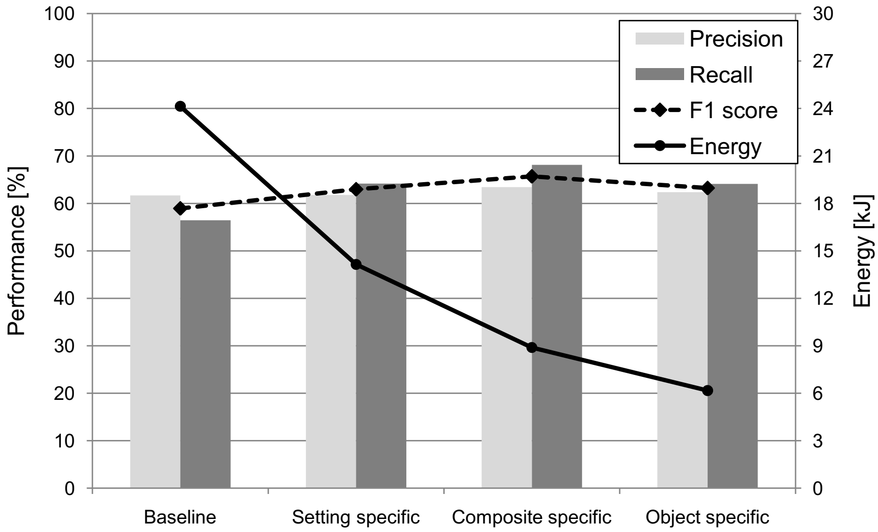

Fig. 12.8. Illustration of the reconfiguration evaluation results of Table 12.4. Recognition performance and energy consumption are shown for individual reconfiguration granularities. Energy consumption decreases monotonic for increasing granularity. Recognition performance shows a maximum for composite activities.

where activity sequences or activity sets are predominantly performed under certain circumstances, locations, etc., provide large potential for reconfiguration.

Domain knowledge of the work place, e.g., the kitchen, etc., allows constraining the set of relevant activities. Widely used approaches in activity recognition apply prior probabilities to tune recognition to a situation. These priors are subsequently dynamically modified based on sensor readings to reflect relevance. Our reconfiguration approach is more rigorous in that it completely excludes activities that are not of interest at the moment. In turn, this exclusion enables using highly optimized recognition models at particular moments, comprising those remaining activities only. Reconfiguration benefits come at the drawback that some decision must be made on relevant activities for a particular reconfiguration. Activities left out of this catalog will not be recognized. By using domain knowledge, we implemented these constraints with the idea that left-out activities are not of interest at the current time. A consumer of the activity recognition service may not even know how to handle such unexpected results.

While our work focused on the optimization of recognition models for low-level activities that are identified from patterns within sensor data streams, our approach is generalizable also to higher-level event processing algorithms. Besides offering the option to use optimized activity models for certain domains, reconfiguration can help to achieve scalable online activity recognition since processing effort for activities is reduced.

We have manually included expert knowledge in defining when and to which activity set to reconfigure, and evaluated our approach with perfectly recognized trigger activities. While this approach was required to identify potential benefits of reconfiguration, additional steps are needed to make reconfiguration suitable for a broad range of applications. To this end, we expect that the following two challenges need to be addressed in further research.

The first challenge is related to automatically construct activity sets of a recognition state machine and derive trigger activities. A statistical analysis of the datasets could reveal recurring activity sequences or domains that offer potential for reconfiguration. Typically, setting-specific reconfigurations can be rapidly derived by using dedicated location sensors (e.g. a PIR sensor at the door) as triggers, and collecting all activities within a room or environment involving localized activities into an activity set. Automatically identifying activity sequences, such as cooking a meal, for composite-specific reconfigurations is however more challenging. Besides identifying relevant activity sequences, a suitable high-level activity recognition system needs to be able to predict a subsequent composite activity to reconfigure to. We have presented such high-level recognition algorithms in our previous work.[19] A current overview of high-level activity pattern detection approaches can be found in Kim et al.[18]

The second challenge is to reliably detect reconfiguration triggers, thus when to switch from one reconfiguration state to another. The recognition of atomic activities from sensor signal patterns as well as that of composite activities is inherently limited in reliability. Hence a reconfigurable system may be cued into erroneous states from which it must be able to escape. By monitoring reported activity patterns in a state, the system could detect that the state might be wrong and an action must be taken. For example, Surie et al.[29] modeled activity event occurrences within a composite activity using a Hidden Markov Model and were able to correct their decisions after a certain delay. In the scope of reconfiguration and online recognition systems, the drawback of delayed decisions is that recognition may be delayed or time periods of incorrect recognitions occur. The structure of our state machine allowed a fallback to a coarser granularity, which includes more activities. This behavior could be helpful if the current state was determined to be not appropriate.

12.9. Conclusion

We have proposed and evaluated benefits of reconfiguration in distributed activity recognition systems using wireless sensor nodes that are worn on the body, are integrated in tools, and in the environment.

Our empirical evaluation showed recognition performance gains and energy cost savings by evaluating reconfiguration in a dataset comprising four settings with 18 composite and 78 atomic activities. Our results demonstrate that fine-grained reconfiguration granularities are beneficial for activity recognition performance. Performance increased from baseline to composite-specific reconfiguration by 11.48 %. When further decreasing reconfiguration granularity to object-specific reconfiguration however, a 4.85 % lower recognition performance was observed. This result indicates that an optimal reconfiguration granularity exists.

With finer reconfiguration granularity, sensor node energy costs reduced monotonically. This result was achieved by switching off sensors. We observed up to 84.5 % in energy saving for a object-specific reconfiguration, and 74.8 % for a composite-specific reconfiguration. Energy costs for reconfiguring the sensor network was negligible compared to energy savings achieved by dynamically turning off sensor nodes.

These results confirm that reconfiguration is a promising research direction to improve scalability and energy efficiency in activity and context recognition systems. Further work should address strategies for recognizing and correcting erroneous reconfiguration states and an automatic composition of the reconfiguration strategy.

References

1. D. Geer, Will gesture recognition technology point the way?, *Computer*. **37**(10), 20–23 (Oct., 2004).
2. D. J. Cook and S. K. Das, How smart are our environments? an updated look at the state of the art, *Pervasive and Mobile Computing*. **3**(2), 53–73, (2007).
3. D. J. Cook, J. C. Augusto, and V. R. Jakkula, Ambient intelligence: Technologies, applications, and opportunities, *Pervasive and Mobile Computing*. **5**(4), 277–298, (2009).
4. G. Pottie and W. Kaiser, Wireless integrated network sensors, *Communications of the ACM*. **43**, 51–58, (2000).
5. M. Ryoo and J. Aggarwal. Recognition of composite human activities through context-free grammar based representation. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, vol. 2, pp. 1709–1718, (2006).
6. D. Kawanaka, T. Okatani, and K. Deguchi, HHMM based recognition of human activity, *IEICE Trans. Information Systems*. **E89-D**(7), 2180–2185 (July, 2006).
7. F. Naya, R. Ohmura, F. Takayanagi, H. Noma, and K. Kogure. Workers' routine activity recognition using body movements and location information. In *Proc. IEEE Int. Symp. Wearable Computers (ISWC)*, pp. 105–108, (2006).
8. G. Ogris, T. Stiefmeier, P. Lukowicz, and G. Tröster. Using a complex multi-modal on-body sensor system for activity spotting. In *Proc. Int. Symp. Wearable Computers (ISWC)*, pp. 55–62, (2008).
9. K. Murao, T. Terada, Y. Takegawa, and S. Nishio. A context-aware system that changes sensor combinations considering energy consumption. In *Proc. 6th Int. Conf. Pervasive Computing (Pervasive)*, pp. 197–212, (2008).
10. K.-F. Lee, Context-dependent phonetic hidden markov models for speaker-independent continuous speech recognition, *IEEE Trans. Acoustics, Speech and Signal Processing*. **38**(4), 599–609, (1990).
11. M. Stäger, P. Lukowicz, and G. Tröster, Power and accuracy trade-offs in sound-based context recognition systems, *Pervasive and Mobile Computing*. **3**(3), 300–327 (June, 2007).
12. H. Ghasemzadeh, E. Guenterberg, and R. Jafari, Energy-efficient information-driven coverage for physical movement monitoring in body sensor networks, *IEEE Journal on Selected Areas in Communications*. **27**(1), (2009).
13. P. Zappi, C. Lombriser, T. Stiefmeier, E. Farella, D. Roggen, L. Benini, and G. Tröster. Activity recognition from on-body sensors: Accuracy-power trade-off by dynamic sensor selection. In *Proc. Europ. Conf. Wireless Sensor Networks (EWSN)*, pp. 17–33, (2008).
14. G. Batori, Z. Theisz, and D. Asztalos. Domain specific modeling methodology for reconfigurable networked systems. In *Proc. Int. Conf. Model Driven Engineering Languages and Systems (MoDELS)*, pp. 316–330, (2007).
15. C. Lombriser, M. Rossi, A. Breitenmoser, D. Roggen, and G. Tröster. Recognizing context for pervasive applications with the titan framework. Technical report, Wearable Computing Laboratory, ETH Zurich, (2009).
16. R. Gravina, A. G. an Giancarlo Fortino, F. Bellifemine, R. Giannantonio, and M. Sgroi. Devel-

opment of body sensor network applications using SPINE. In *Proc. IEEE Int. Conf. Systems, Man and Cybernetics (SMC)*, (2008).

17. V. Osmani, S. Balasubramaniam, and D. Botvich, Human activity recognition in pervasive health-care: Supporting efficient remote collaboration, *Network and Computer Applications*. **31**(4), 628–655, (2008).
18. E. Kim, S. Helal, and D. Cook, Human activity recognition and pattern discovery, *Pervasive Computing Magazine*. **9**(1), 48–53, (2010).
19. O. Amft, C. Lombriser, T. Stiefmeier, and G. Tröster. Recognition of user activity sequences using distributed event detection. In *Proc. Europ. Conf. Smart Sensing and Context (EuroSSC)*, pp. 126–141, (2007).
20. W. Dargie and T. Tersch, Recognition of complex settings by aggregating atomic scenes, *IEEE Intelligent Systems*. **23**(5), 58–65, (2008).
21. O. Amft and G. Tröster, Recognition of dietary activity events using on-body sensors, *Artificial Intelligence in Medicine*. **42**(2), 121–136, (2008).
22. H. Junker, O. Amft, P. Lukowicz, and G. Tröster, Gesture spotting with body-worn inertial sensors to detect user activities., *Pattern Recognition*. **41**(6), 2010–2024 (June, 2008).
23. Q. Xu, M. Kamel, and M. M. A. Salama. Significance test for feature subset selection on image recognition. In *Proc. Int. Conf. Image Analysis and Recognition (ICIAR)*, pp. 244–252, (2004).
24. L. I. Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms*. (Wiley Interscience, 2005).
25. P. Zappi, C. Lombriser, E. Farella, L. Benini, and G. Tröster. Experiences with experiments in ambient intelligence environments. In *Proc. IADIS Int. Conf. Wireless Applications and Computing*, pp. 171–174 (June, 2009).
26. O. Amft. Adaptive activity spotting based on event rates. In *Proceedings of the IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC)*, pp. 169–176, (2010). doi: 10.1109/SUTC.2010.63.
27. B. Bougard, F. Catthoor, D. C. Daly, A. Chandrakasan, and W. Dehaene. Energy efficiency of the IEEE 802.15.4 standard in dense wireless microsensor networks: Modeling and improvement perspectives. In *Proc. Conf. Design, Automation and Test in Europe (DATE)*, pp. 196–201, (2005).
28. J. Polastre, R. Szewczyk, and D. Culler. Telos: enabling ultra-low power wireless research. In *Proc. Int. Symp. Information Processing in Sensor Networks (IPSN)*, p. 48, (2005).
29. D. Surie, F. Lagriffoul, T. Pederson, and D. Sjlie. Activity recognition based on intra and extra manipulation of everyday objects. In *Proc. Int. Symp. Ubiquitous Computing Systems (UCS)*, pp. 196–210, (2007).