

Spotting Human Activities and Gestures in Continuous Data Streams

A dissertation submitted to
TECHNISCHE UNIVERSITÄT DARMSTADT
Fachbereich Informatik

for the degree of
Doktor-Ingenieur (Dr.-Ing.)

presented by

ANDREAS ZINNEN

Dipl. Inform.

born 5th of June, 1978
in Bernkastel-Kues, Germany

Prof. Dr. Bernt Schiele, examiner
Prof. Dr. Paul Lukowicz, co-examiner

Date of Submission: 26th of May, 2009

Date of Defense: 7th of July, 2009

Darmstadt, 2009

D17

Abstract

In this thesis we use algorithms on data from body-worn sensors to detect physical gestures and activities. While gesture recognition is a promising and upcoming alternative to explicitly interact with computers in a mobile setting, the user's activity is considered an important part of his/her context which can help computer applications adapt automatically to the user's situation. Numerous context-aware applications can be found ranging from industrial to medical to educational domains. A particular emphasis of this thesis is the recognition of short activities or quick actions, which often occur amid large quantities of irrelevant data.

Embedded in different application scenarios, we focus on four challenges in gesture and activity recognition: multiple types and diversity of activities, high variance in performance and user independence, continuous data stream with large background and finally activity recognition on different levels. We make several contributions to overcome these challenges. We start with a method for activity recognition using short fixed positions of the wrist to extract activities from a continuous data stream. Postures are used to recognize short activities in continuous recordings.

In order to evaluate the distinctiveness of gestures in continuous recordings of gestures in daily life, we present a new approach for the important and challenging problem of user-independent gesture recognition. Beyond the recognition aspects, we pay particular attention to the social acceptability of the evaluated gestures. We performed user interviews in order to find adequate control gestures for five scenarios.

Activity recognition is typically challenged by spotting a large number of activities amid irrelevant data in a user-independent manner. We present a model-based approach using joint boosting to enable the automatic discovery of important high-level primitives that are derived from the human body-model. Subsequently, we systematically analyze the benefit of body-model derived primitives in different sensor settings for multi activity recognition. Furthermore, we propose a new body-model based approach using accelerometer sensors thereby reducing the sensor requirements significantly.

The proposed methods to recognize ‘atomic’ activities such as drilling, handshaking, or walking do not scale well for high-level tasks composed of multiple activities. A prohibitive amount of training would be required to cover the high variability and the large number of possibilities to execute high-level tasks. To this end, an approach considering temporal constraints encoded in UML diagrams enables a reliable recognition of composed activities or high-level tasks without requiring large amounts of training data. We show the validity of the approach by introducing a realistic and challenging data set.

Zusammenfassung

In dieser Arbeit verwenden wir Algorithmen, um in den Daten tragbarer Sensoren physische Gesten und Aktivitäten zu erkennen. Während Gestenerkennung eine viel versprechende Alternative zur expliziten Interaktion mit dem Computer in mobilen Szenarien ist, kann die Aktivität eines Benutzers als wichtiger Teil seines Kontextes berücksichtigt werden, um Computer-Anwendungen automatisch an die Situation des Benutzers anzupassen. Für die automatische Erkennung von Kontext existieren eine Vielzahl von Einsatzfeldern, beispielsweise im industriellen, medizinischen oder pädagogischen Bereich. Diese Arbeit setzt den Schwerpunkt auf die Erkennung von kurzen und schnellen Aktivitäten, die nicht selten inmitten großer Mengen irrelevanter Daten auftreten.

Eingebettet in unterschiedliche Anwendungsszenarien konzentrieren wir uns auf vier Herausforderungen bei der Erkennung von Gesten und Aktivitäten: Erstens eine Vielzahl unterschiedlicher Aktivitäten; zweitens eine hohe Varianz in der Ausführung sowie die Unabhängigkeit vom Benutzer; drittens eine Erkennung im kontinuierlichen Datenstrom inmitten von Hintergrunddaten; und schließlich das Erkennen von Aktivitäten auf verschiedenen Ebenen. Um sich den Herausforderungen zu stellen, leistet diese Arbeit mehrere Beiträge: Wir beginnen mit einer Methode, die kurze feste Positionen des Handgelenks berücksichtigt, um Aktivitäten in einem kontinuierlichen Datenstrom zu segmentieren.

Zur Beurteilung, in wieweit sich explizite Gesten von alltäglichen Gesten unterscheiden, präsentieren wir einen neuen Ansatz, der das wichtige und schwierige Problem der Benutzer-unabhängigen Gestenerkennung addressiert. Neben technischen Aspekten der Erkennung legen wir besonderen Wert auf die soziale Akzeptanz der bewerteten Gesten. Dazu haben wir Interviews durchgeführt, um Benutzer über passende explizite Gesten in fünf unterschiedliche Szenarien zu befragen.

Typischerweise sind vielfältige Aktivitäten inmitten irrelevanter Daten sowie eine Unabhängigkeit vom Benutzer große Herausforderungen bei der Erkennung von Aktivitäten. Wir stellen einen Modell-basierten Ansatz vor, bei dem Joint Boosting automatisch aus einem Körper-Modell abgeleitete Bewegungs-Primitive erkennt. Anschließend analysieren wir systematisch den Nutzen unseres Ansatzes für eine Erkennung von Aktivitäten unter Berücksichtigung verschiedener Sensor-Konfigurationen. Außerdem schlagen wir ein neues Körper-Modell vor, das ausschließlich auf Daten von Beschleunigungssensoren basiert und somit die Anforderungen an die Sensoren deutlich verringert.

Die vorgeschlagenen Methoden zur Erkennung von atomaren Aktivitäten wie Bohren, Hände Schütteln oder Gehen eignen sich schlecht für die Erkennung von höherwertigen Aufgaben, die aus mehreren Aktivitäten zusammengesetzt sein können. Viele Trainingsdaten wären notwendig, um die hohe Variabilität sowie die große Zahl der Möglichkeiten bei der Ausführung von höherwertigen Aufgaben abzudecken. Zu diesem Zweck stellen wir einen Ansatz vor, der zeitliche Einschränkungen in UML-Diagrammen berücksichtigt und damit eine zuverlässige Erkennung von höherwertigen Aufgaben mit weniger Trainingsdaten ermöglicht. Wir zeigen die Stärken unseres Ansatzes auf einem neuen Datensatz mit realistischen und anspruchsvollen Daten.

Acknowledgments

First I would like to express my sincerest gratitude to my academic supervisor, Professor Bernt Schiele, for his guidance and motivation throughout this thesis. Both our inspiring discussions about thoughts or problems and his help in streamlining my ideas are an inherent part of this thesis. Apart from his constructive and productive cooperation, he introduced me to techniques for time management as a requisite for coordinating various tasks such as research, lectures and projects at the university and at SAP.

Special thanks goes to Professor Paul Lukowicz. I thank him especially for his interest in my work and for agreeing to be the co-examiner of this thesis. In addition, I encountered interesting work and discussions with him and his group in the wearIT@work project.

The work on this thesis would not have been possible without the MIS group in Darmstadt of which I was a part. Special thanks goes to Uschi who could always spare a minute for work, but also for private matters. I am very obliged for her patience and encouragements within the past years. Equally, I thank Micha Andriluka, Ulf Blanke, Victoria Carlsson, Mario Fritz, Tâm Huỳnh, Nicky Kern, Kristof van Laerhoven, Diane Larlus, Bastian Leibe, Niko Mayer, Paul Schnitzspan, Edgar Seemann, Michael Stark, Ulrich Steinhoff, Maja Stikic, Stefan Walk, and Christian Wojek.

I would also like to thank SAP for the opportunity to get insight into both international research and industrial environment. It was a pleasure to work with my colleagues from SAP, foremost Tobias Klug, Fernando Lyardet and Odilia Machuca Bernal as well as my project team members Andreas Faatz, Eicke Godehardt, Manuel Goertz and Robert Lokaiczyk.

Several colleagues have contributed to specific parts of this thesis. Major thanks go to Kristof van Laerhoven, who provided me with the wireless XSens, corrected my submissions and volunteered for several data recordings. Mario Fritz provided valuable support and discussions for our work on gesture recognition as described in Chapter 4. Christian Wojek made the joint boosting code available as used in the Chapters 5, 6, and 7. Beyond that, I would like to thank him for never ending help and patience with my machine learning problems. Ulf Blanke was of crucial help for our work in Chapter 6. He implemented a new body-model based approach using accelerometer sensors thereby reducing the sensor requirements significantly and took part in writing the submission.

Finally and most of all, I would like to thank my family and friends for their help and support. Thank you to Jelly who supported my decision to start a Ph.D. and to Alla, Arya and Dan for correcting parts of this thesis. Special thanks go to my parents. Only with their love and patience, I was able to successfully finish this work.

Contents

1	Introduction	1
1.1	Challenges	2
1.1.1	Main challenges addressed in this thesis	2
1.1.2	Challenges not directly addressed in this thesis	4
1.2	Contributions	5
1.3	Thesis Outline	7
2	Related Work	9
2.1	Context-Aware Applications and Activity Recognition	9
2.1.1	Context-aware systems	9
2.1.2	Activity recognition	10
2.2	Work Directly Addressing the Thesis' Contributions	12
2.2.1	Segmentation methods of continuous data streams	12
2.2.2	Low-level gesture and activity recognition approaches	15
2.2.3	Hierarchical activity recognition approaches	17
2.3	Machine Learning	18
2.3.1	Hidden Markov model (HMM)	19
2.3.2	Boosting and joint boosting	20
2.4	Case Study on Gesture Interaction During Clinical Ward Rounds	21
2.4.1	Introduction	21
2.4.2	Low cost sensors & software	23
2.5	Discussion	24

3	Recognizing Short and Non-Repetitive Activities from Wearable Sensors	25
3.1	Introduction	25
3.2	Experimental Setup	27
3.2.1	Data set & annotation	27
3.2.2	Hardware - MTx inertial measurement unit (IMU)	27
3.3	Activity Recognition with Hidden Markov Models	28
3.3.1	HMMs on hand segmented activities	29
3.3.2	Left-right HMMs on continuous data	30
Evaluation criteria	30	
Results of left-right HMMs on continuous data	31	
3.4	Segmentation and Classification Using Postures	32
3.4.1	Segmentation of the continuous data stream	32
Grouping of the data in regions	33	
Segmenting the data	34	
Discarding overlapping segments	34	
3.4.2	Classification using postures	35
Classification using separate postures	35	
Classification with combined postures	37	
3.5	Conclusion	37
4	User-Independent Gesture Recognition in Continuous Data Streams	39
4.1	Introduction	39
4.2	Experimental Setup	40
4.2.1	Collecting background data	40
4.2.2	Defining and collecting the navigation gestures	41
4.3	Gesture Segmentation and Classification in Continuous Data Streams	43
4.3.1	Segmentation of the continuous data stream	43
4.3.2	Calculation of features	45
4.3.3	Gesture classification	47

4.4	Experiments and Results	48
4.4.1	Evaluation of control gestures against background	48
4.4.2	Confusion between control gestures	51
4.5	Conclusions	51
5	Multi Activity Recognition based on Body-Model-Derived Primitives	53
5.1	Introduction	53
5.2	Activity Segmentation in Continuous Data Streams	54
5.2.1	Body-model	54
5.2.2	Segmentation	57
5.3	Calculation of Primitives and Features	58
5.3.1	Motion Primitives	59
Height Primitives	59	
Push-Pull Primitives	59	
Bending Primitives	61	
Twist Primitives	61	
Direction Histograms	62	
5.3.2	Posture Features	63
5.3.3	Location Features	63
5.4	Multi-Activity Recognition using Joint Boosting	63
5.5	Experiments and Results	65
5.5.1	Car-quality control data set	65
5.5.2	Evaluation of the model-based approach to activity recognition	65
User-dependent experiments	66	
Across-user experiments	67	
General results	67	
5.6	Discussion and Conclusion	70

6 An Analysis of Sensor-Oriented vs. Model-Based Activity Recognition	71
6.1 Introduction	71
6.2 Additional Features and Adapted Body-Model	72
6.2.1 Body-model by IMUs (BM_IMU) and location features	72
6.2.2 Body-model by acceleration (BM_ACC)	73
6.2.3 Signal Features	74
6.3 Multi-Activity Recognition using Joint Boosting and HMMs	75
6.3.1 Hidden Markov models (HMMs)	75
6.4 Evaluation	76
6.5 Experimental Results	77
6.6 Conclusion	81
7 Hierarchical Activity Recognition Using UML Diagrams	83
7.1 Introduction	83
7.2 Data Set	85
7.2.1 Sensor setup and annotation	89
7.3 Hierarchical Activity Recognition using UML	89
7.3.1 Level-1: Multi-activity recognition using joint boosting	90
7.3.2 Level-2: Sequential constraints given by UML diagram	90
7.3.3 Level-3: Activity recognition considering temporal constraints of UML diagrams	91
7.4 Experiments and Results	91
7.4.1 Evaluation procedure	92
7.4.2 Results for level-1 activities	93
7.4.3 Results for composite level-2 activities	95
7.5 Conclusion	96
8 Conclusion and Outlook	97
8.1 Conclusions	97
8.2 Outlook	99

List of Figures	101
List of Tables	105
Bibliography	107

1

Introduction

In recent years, the price and size of computers decreased while becoming even more powerful. In addition, new technologies like GPS, UMTS, WiFi or Bluetooth provide an opportunity to access information or to be accessible almost anywhere and anytime. Due to this trend, these days computers and sensors are embedded in mobile devices or in the environment, for example, in telephones, cameras, cars, digital meeting rooms or buildings. We all experience and realize that computers already have high impact on our daily life today. Moreover, the spreading of computers is an ongoing development. The new era of ubiquitous computing has started.

As a source of conflict, the move from stationary to ubiquitous usage of computers often means that users have to share attention between computers and their daily lives. Clearly this trend impacts the human interaction with computers more and more. The interaction with devices, especially in mobile scenarios, has to change. Considering current computer technology, *explicit interaction* means that discrete actions are issued by the user, who in turn expects specific responses from the system. Whereas in stationary scenarios, the use of screens, mice and keyboards dominates the human-computer interaction, those standard devices are limited in ubiquitous computing. Speech and gesture recognition are promising and upcoming alternatives when explicitly interacting with computers in a mobile setting.

In addition to alternative explicit interaction methodologies, designers of ubiquitous systems aim for a shift to more *implicit interaction* [[Schmidt 2000](#)] based on situational context. By leveraging the availability of processing power and advanced sensor technology, the computer ideally can obtain a certain understanding of our behavior in the given situation. This knowledge is considered as additional input to the computer thereby reducing the necessary amount of explicit human computer interaction. As a result, the user can concentrate on his/her task rather than on the computer interface.

For example, less urgent notifications may be withheld from a user if the system detects that he/she is engaged in a meeting, or a computer system might be able to pre-fill checklists in a maintenance scenario or to automatically display corresponding patient records in medical ward rounds thereby reducing the user's effort for explicit interaction. Or a computer system, aware of necessary and already performed work steps, can warn a user about missing and critical sub-tasks.

A user's context is typically characterized by his/her identity, his/her current location, the activity he/she is performing, his/her social interactions and the state of his/her environment [Dey 2001]. To estimate the user's current context, context-aware computing uses sensors that fall into two categories. First, some are worn or carried in the sense that they move along with us. Wearable sensors are especially suited for several applications because they provide means to leverage the power of gathering personal context while at the same time providing the means to be mobile. In a second category devices are part of the environment or worn by people whom we meet.

The focus of this thesis is a sub-field of context-aware computing. Using wearable sensors, combined with methods from machine learning, we are able to record, learn and recognize different types of atomic or low-level activities. In this thesis, we define the term *spotting* as a method that detects and recognizes short activity occurrences of less than a couple of seconds in large background data. In addition, we introduce an approach for high-level activity recognition. In this thesis, a high-level or composite activity is composed of several atomic activities allowing a high variability in temporal ordering.

As we will see in this work, activity recognition with wearable sensors is an important component for numerous context-aware applications ranging from industrial to medical to educational domains. We are also able to directly apply these methods to explicit gesture recognition due to the fact that activities are implicitly composed of gestures.

Next, we present some of the most important challenges faced when recognizing activities and gestures. Subsequently, we give an overview of our contributions in this thesis, which address several of these challenges.

1.1 Challenges

In this section, we discuss some important research challenges in activity and gesture recognition. First (Section 1.1.1), we discuss four challenges in more detail that are central to this thesis. Second (Section 1.1.2), we give a short overview of challenges that are not directly addressed in this thesis.

1.1.1 Main challenges addressed in this thesis

Multiple types and diversity of activities Realistic scenarios typically require distinguishing between multiple activities. A consideration of multiple activities is challenging because of two reasons.

First, activities can differ significantly in their constitution. Several activities are characterized by repetitive movements like turning the arm when screwing or moving the arm up and down while hammering. Beyond these activities of longer duration (>10 seconds), very short activities (<3 seconds) like opening a door, drilling, or hanging up boxes are of

interest. Not only the variable duration complicates the detection of activities. In addition, short activities often do not contain discriminant arm movements. Whereas hammering or turning screws can be identified by measurable arm movements, the arm position hardly changes for activities like cutting a paper template or marking holes for drilling.

Second, different activities are often ambiguous due to their similarity to each other (inter class similarity). For example in the maintenance area, activities like hammering, drilling and screwing occur frequently and only differ slightly when performed in diverse working steps. Since it is desirable to identify the activities in the specific work steps, the mentioned similarities of arm postures and movements challenge a recognition system.

High variance in performance and user independence Although subjects perform a same activity, a high variance in execution (intra class variability) can be observed for many interesting activities. Not only does the speed of performance have an impact on the execution of an activity. For instance a user who opens a car door can use both hands in various ways. Considerable variability also occurs because the user is often not constrained to use a specific arm when performing an activity. Additionally, a change of the user's position relative to an object will influence the way of interaction. Obviously, high intra-class variance is a key factor that has to be addressed when recognizing activities.

This variance in performance is further reinforced across different subjects. Note that activity recognition in most cases relies on annotated recordings of activity data in order to train a machine learning algorithm. Obtaining such data in realistic settings, especially together with sufficiently detailed annotations is tedious and time-consuming. Given the large amount of different activities, it is not feasible to expect an annotated data recording for each user. Instead, successful recognition approaches have to enable robust activity recognition across-users from annotated data of only a few users.

Continuous data stream with large background Short physical activities or gestures, for instance explicit hand-gestures or implicit gestures while performing an activity, are usually embedded in a large background data stream. Therefore, an extraction of relevant segments to separate activities from irrelevant background data seems important. Particularly the observation that daily gestures in the background data often do not differ much from gestures in an annotated activity make the segmentation of the continuous data stream a crucial and challenging part of activity recognition.

Activity recognition on different levels A majority of work in activity recognition focuses on low-level activities. While good results have been reported for ‘atomic’ activities (such as drilling, handshaking, or walking), the recognition of ‘composite’ activities or complete tasks has proven to be far more challenging. For instance, activities like *building a wooden cabinet* or *changing a tire of a car* are composed of many sub-activities, often performed in changing orders. Whereas state-of-the-art machine learning techniques

have proven to be appropriate to train classifiers for ‘atomic’ activities, if they have access to sufficient training data, these methods do not scale well to the recognition of high-level tasks that are composed of multiple activities. The main reason is that a prohibitive amount of training data would be required to cover the high variability and the large number of possibilities to execute high-level tasks. Obviously, systems have to consider additional information to make a step toward a reliable and realistic recognition of ‘composite’ high-level activities.

1.1.2 Challenges not directly addressed in this thesis

Usability Embedding computers in everyday life increases the demands on a system’s usability. When designing wearable devices one must consider trade-offs between usability, portability, and unobtrusiveness for every device as motivated by [Starner 2001]. Architecture and experience reports to assist the study of usability in wearable computer applications are rarely available. [Lyons and Starner 2001] motivate that to fully understand the interaction between the user and the machine, the researcher must also examine the context in which the interaction takes place. Especially the sensors’ size, the number, placement, as well as the runtime strongly influence the users’ subjective perception of a wearable system. The number and types of sensors affect the power consumption and thereby the users’ effort to recharge batteries. Different sensor types will also influence the sensor size and placement of the sensors. Therefore, it is crucial that the development of activity recognition systems considers a scenario-specific analysis of type and number of necessary sensors as well as the placements for reliable and sufficient recognition. To that end, chapter 6 includes an analysis of sensor settings in a car quality assessment scenario.

Hardware and power In recent years, significant progress was made in miniaturization of sensors, increased memory size and processing power [Van Laerhoven *et al.* 2006, XSens 2009]. Despite those advances, the energy supply of wearables remains a bottleneck. Although the power shortage does not directly affect the research of this thesis, it will become a relevant problem when deploying a corresponding system in a realistic environment. In addition and especially for wireless sensors, the reliability and the robustness are still challenges in hardware design. Since activity recognition is still in an early stage, we did not focus on hardware issues. For the recordings of this thesis, a runtime of more than three hours was sufficient when using wireless sensors. Additionally, in some recordings the sensors were directly wired to a laptop.

Long term data recordings Long-term recording of wearable sensor data has gained interest especially in the medical field. Typically, the resources required for achieving this are high for both users and medical staff [Van Laerhoven *et al.* 2008, Begole *et al.* 2003]. Having continuous activity data spanning several months allows a wide variety of applications, for example correlating mood swings of psychiatry patients with collected

sensor data. Logging such detailed activity data for months is challenging. It is crucial that sensor systems have to be worn 24 hours per day, thereby reducing the user interaction to a minimum.

Unsupervised approaches The annotation and recording of training data for supervised activity recognition can be time-consuming and erroneous. Therefore, several approaches apply unsupervised learning methods [Clarkson and Pentland 1999, Liao *et al.* 2007a, Huynh and Schiele 2006a, Huynh and Schiele 2006b]. The ability to learn from no or sparsely-annotated data would greatly simplify many problems in activity recognition. Consequently, semi-supervised and unsupervised machine learning methods gain importance. For short activities in a large background, as considered in this thesis, unsupervised learning is not a feasible approach since the background prevails the activities of interest. Therefore, an automatic discovery of how the activity data is organized is not possible.

Privacy and security As ubiquitous computing penetrates more and more areas of daily and working life, privacy critical data is stored and processed on several computers [Langheinrich 2005, Mayrhofer and Gellersen 2007]. The established IT security solutions are not fully viable in pervasive scenarios, in particular if a subset of these machines acts autonomously as substitutes for humans or organizations. Obviously, effort has to be taken in IT security research to adapt trust models to the upcoming needs of ubiquitous computing. In addition, the automatic recognition and documentation of performed activities are vulnerable to abuse for performance control of employees in a company. In this thesis, we do not directly address privacy and security issues in ubiquitous computing. But Chapter 8 touches on impacts of activity recognition on a user's privacy regarding performance control as discussed in [Zinnen *et al.* 2008].

1.2 Contributions

The thesis takes several steps to address the main listed challenges in Section 1.1.1. Altogether, we make three contributions facing the main challenges as outlined in the enumeration.

1. **Segmentation method of continuous data streams** Applying a segmentation method, segments of interest are reliably extracted from the background data that are easier to classify than for example using a sliding window approach on the continuous data stream. The segmentation turns out to be a crucial part of the presented recognizer. Therefore, the method is expanded to meet additional demands in several scenarios.
2. **Low-level gesture and activity recognition approach** We propose a low-level gesture and activity recognition approach by applying joint boosting. The definition and detection of general movement primitives and postures thereby enables

user-independent low-level activity and gesture spotting of both non-repetitive and repetitive movements. In a first step, the primitives are directly detected in the data stream of one sensor. An extension is based on a human body-model. Especially, the body-model allows the calculation of robust features for the subsequent multi-activity recognition across users.

3. **Hierarchical activity recognition approach** Based on the second contribution, this approach bridges the gap from low-level activities and gestures to high-level activity recognition. Here, temporal information, for example encoded in UML diagrams, can be used to cope with little training data for high-level activities.

In this section we will describe the contributions in more detail. We begin our research with a method for short-term activity recognition. Start and end postures (short fixed positions of the wrist) are used in order to identify segments of interest in a continuous data stream. Whereas existing approaches often use sliding windows with a variable window length, this approach enables the efficient differentiation of short activities from unimportant background data. This proposed method provides a basis to meet the challenge *Continuous data stream with large background*. Experiments in Chapter 3 show the applicability of using postures to recognize short activities in continuous recordings. While this approach is already able to sufficiently deal with little training data, the recognition is still user dependent and does not model the movement part between the postures yet.

In challenge *High variance in performance and user independence* of Section 1.1.1, we argue that the variance when performing activities and gestures is high. Additionally, we stress the importance of user independence. Chapter 4 proposes a new way for the important and challenging problem of user-independent gesture recognition in continuous data streams. The work extends the segmentation introduced in Chapter 3 and defines movement primitives as subparts of a gesture. The recognition algorithm considers both histograms over the movement direction in a primitive and the movement shape between primitives for classification. In contrast to most papers in this area, we evaluate seven gestures of different complexity (note that the gestures have to be distinguishable) against a realistic background class of daily gestures in five different scenarios. The work in Chapter 4 still recognizes explicit gestures. This task is obviously less complex than the recognition of activities because explicit gestures do not vary as much in execution. The recognition of implicit gestures in activities is considered in the following chapters.

Chapter 5 makes a first step toward user-independent and multi activity recognition. Therefore, we transfer the results of Chapter 4 to activity recognition. We propose a novel and robust model-based approach using high-level primitives that are derived from a human body-model estimated from sensor data. Joint boosting enables the automatic discovery of important and distinctive features. A study with 8 participants performing 20 activities in a quality inspection of a car production process shows the feasibility of the approach to address the challenges *Multiple types and diversity of activities* and *High variance in performance and user independence*.

Chapter 5 proposes model-based activity recognition as an alternative to signal-oriented recognition. The benefits of body-model derived features compared to standard signal-oriented features have not been evaluated yet. In the first part, Chapter 6 systematically analyzes the benefits of body-model derived primitives in different sensor settings for multi activity recognition. Results of incorporating location are presented in the second part and the third part reports on the results of reducing the number of sensors. This analysis basically addresses the challenges *High variance in performance and user independence* (Section 1.1.1) and *Usability* (Section 1.1.2), in particular the impact of different sensor settings on the recognition performance.

At the end of this thesis, we bring together all introduced themes. Chapter 7 proposes a novel method for multi-level activity recognition. Temporal constraints, for example encoded in UML diagrams, enable reliable recognition of composed activities or high-level tasks without requiring large amounts of training data. The recognition of low-level activities builds on the approaches of the previous chapters. We introduce a realistic and challenging data set of ten independent subjects and experimentally show the ability of the approach to model and recognize highly variable activities and tasks across users. This chapter addresses the main challenges *Multiple types and diversity of activities*, *Continuous data stream with large background*, *Activity recognition on different levels* and *High variance in performance and user independence* as introduced in Section 1.1.1.

Parts of this thesis have been published or submitted as refereed conference papers. The navigation system employing face detection for segmentation as motivated in Section 2.4 is reported in [Zinnen et al. 2007b]. The method for short-term activity recognition using start and end posture (Chapter 3) was first proposed in [Zinnen et al. 2007a]. A new way for user-independent gesture recognition in continuous data streams is published in [Zinnen and Schiele 2008] (Chapter 4). The results of Chapter 5 extending and transferring the methods to activity recognition are addressed in [Zinnen et al. 2009b]. A comparison of model-based and signal-oriented features as well as the impact of location and number of sensors is published in [Zinnen et al. 2009a] (Chapter 6). Finally, [Zinnen et al. 2008] includes a discussion about privacy issues of activity recognition as considered in the conclusion of this thesis in Chapter 8.

1.3 Thesis Outline

This thesis is organized as follows.

Chapter 2 gives an overview of related work in the area of context-awareness and activity recognition with wearables. We start with a general overview of context-aware systems, activity and gesture recognition. Accordingly, we review work related to our main contributions. We close the chapter by presenting general related machine learning approaches that have been proposed for activity recognition.

Chapter 3 presents our first approach for activity spotting. The introduced segmentation of the continuous data stream turns out to be crucial for all following works. Postures

show high discriminative power to recognize short activities in a user dependent manner.

Chapter 4 proposes a new approach for gesture recognition in continuous data streams. We define primitives as subparts of gestures and evaluate the user-independent recognition of seven gestures against a realistic background class of daily gestures in five different scenarios.

In **Chapter 5**, we introduce a novel model-based approach to activity recognition using high-level primitives that are derived from a human body-model. The approach shows robustness in a specific scenario composed of 20 activities in quality inspection of a car production process.

Chapter 6 compares benefits of the introduced body-model with standard signal-oriented features. In addition, usability issues are considered. We propose a new body-model based approach using accelerometer sensors thereby reducing the sensor requirements significantly.

Chapter 7 expands the activity recognition of the previous chapters to multi-level activity recognition. Temporal constraints encoded in UML diagrams enable to reliably recognize composed activities or high-level tasks without requiring large amounts of training data.

Chapter 8 summarizes the work of the thesis, draws conclusions and gives an outlook of possible future work.

2

Related Work

As motivated in Chapter 1, the consideration and detection of context [Dey 2001], especially of the user’s activity, is crucial for computer systems directly supporting people in everyday activities. This chapter gives an overview of current trends and the state-of-the-art in context awareness and activity recognition.

The chapter is organized as follows. First we review general literature on context-aware applications and activity recognition (Section 2.1). We then present a deeper discussion on related work considering the main contributions of this thesis in Section 2.2. Next, we provide a review of machine learning approaches that have been applied for activity recognition (Section 2.3). The section focuses on generative and discriminative supervised methods as applied in this thesis. Subsequently, we summarize an exemplary case study on gesture interaction during clinical ward rounds (Section 2.4). We conclude the chapter with a short discussion (Section 2.5).

2.1 Context-Aware Applications and Activity Recognition

The diversity of context-aware systems proposed and explored within the last decade is large. The following section first presents works of context-aware systems in general. Subsequently, we focus on the user’s activity as an important ingredient of his/her context.

2.1.1 Context-aware systems

Early context-aware applications had a strong focus on location [Harter and Hopper 1994]. [Want *et al.* 1992, Harter *et al.* 1999] present the *active badge* system using ultrasound beacons to determine the position of the active badges. By coupling it to a standard telephone exchange the system is able to route telephone calls to the phone next to its recipient. *Teleporting* presented by [Bennett *et al.* 1994] is a tool to dynamically map the user interface onto the resources of the surrounding computer and communication facilities. Also based on the *active badge* system, this tool can track a user’s location so that the application follows the user while he/she moves around.

[Brown 1996] presents *stick-e notes* delivering notes to the user activated by a specific context. As soon as a certain context, such as location or time, is recognized, the system becomes active. The authors envision the use of such notes in multiple scenarios like tourist guide applications, mobile workers, making surveys, or control panels. [Dey and Abowd 2000] present a similar system for context-aware reminders.

The use of context-aware spaces has also been investigated. [Abowd 2000] introduces the *classroom 2000* system to record lectures that can later be retrieved by the students. [Nagel *et al.* 2001] present the *family intercom* application allowing setting up an audio connection to any family member in a house. Instead of using fixed wall-mounted intercom in a house, room-mounted microphones are utilized.

In addition, much of the work on the management of notifications can in fact be considered context-aware. [Horvitz *et al.* 2002] present challenges of forecasting computer users' availability and describe the creation of *COORDINATE*, a system with the ability to log events from multiple devices for this purpose. *COORDINATE* considers the user activity and proximity from multiple devices, in addition to an analysis of the content of users' calendars, the time of day, and day of week to infer his/her current context.

Recent work by [Dourish 2004, Dey and Mankoff 2005, van Sinderen *et al.* 2006, Anid Dey and Kodama 2006] analyzes how computation can be made sensitive and responsive to its setting. The authors especially explore how computational devices can be attuned to variations in the settings in which they are used when moving from one physical or social setting to another.

[Pentland *et al.* 2005, Pentland 2007] analyze social patterns in organizations, thereby extending techniques such as activity-recognition to possibly large networks of individuals. Analyzing face-to-face conversations using wearable audio, the authors model social networks that help to identify experts in an organization or improve the composition of project teams. [Eagle and Pentland 2006] use context information gathered from mobile phones in order to identify common structures in the users' daily routines.

The mentioned applications are just samples in a large number of diverse context-aware scenarios. The following part gives a brief overview of related work in activity recognition, especially with body-worn sensors.

2.1.2 Activity recognition

Current research in activity recognition from wearable sensors is characterized by a wide range of topics and applications. The following section outlines application areas for activity recognition systems in wearable or mobile settings. We begin with applications for health care and assisted living. Afterwards, we summarize industrial applications. We close with applications for entertainment and gaming.

Health care and assisted living Longer life expectancy and lower birthrates are increasing the proportion of the elderly population thereby making high demands on existing health care systems. A major goal of current research in activity recognition and context-aware computing in general is the development of technologies that can help in addressing these challenges. For instance, activity recognition could help elderly people to live more independent lives. [Backman *et al.* 2006] and [Si *et al.* 2007] aim to support persons suffering from dementia through the use of context-aware reminders and similar assistance. In addition, potentially dangerous situations in a person's life could be detected to call external help automatically [Jafari *et al.* 2007, Bourke *et al.* 2007, Liszka *et al.* 2004, Villalba *et al.* 2006, Anliker *et al.* 2004].

Other applications employ long-term monitoring to detect unusual patterns in a person's daily life indicating early symptoms of diseases. Whereas automatic detection of fine-grained behavioral changes is highly challenging and still a long-term goal of research in activity recognition, applications summarizing statistics about daily activities [Choudhury *et al.* 2006] or performing continuous recordings of physiological parameters [Anliker *et al.* 2004, Liszka *et al.* 2004, Paradiso *et al.* 2005] can already be valuable for physicians to estimate a person's physical constitution.

In contrast to the previous approaches, other systems focus on a healthy lifestyle of all age and health groups. [Maitland *et al.* 2006, Consolvo *et al.* 2008] use fluctuations in mobile phone signals or wearable sensors to estimate a person's activity. The authors discover that awareness increases the motivation for daily activities. [Andrew *et al.* 2007] pursue similar goals by combining activity and location information from wearables to suggest spontaneous exercises.

[Bardram and Christensen 2007, Tentori and Favela 2008] aim for an automatic support of hospital staff in their daily routines. Health records of nearby patients are displayed on a mobile display, and a prioritization of patients is based on the patient's health condition. [Tentori and Favela 2008] envision a bracelet worn by nurses, fitted with LEDs for each patient that change color based on the patient's health condition.

Industrial applications In industrial settings, one can imagine that users could benefit from activity recognition in a wide range of applications. Activity-aware applications have the potential to supply workers with manual information while performing their primary work task, reduce the risk of mistakes or support workers in documentation by for example pre-filling a check-list.

[Ward *et al.* 2006] combine data from wearable microphones and accelerometers in order to track wood shop activities such as sawing or hammering. A reliable recognition reduces the user's cognitive and physical effort while interacting with wearable systems to prevent distracting the user from his/her primary task.

[Lukowicz *et al.* 2007] explore new solutions to support the workers of the future. The authors test the effectiveness and applicability of wearable computing technology on four different pilot studies in the fields of health care, emergency rescue, aircraft maintenance

and production management. In these scenarios, wearable technology and activity recognition are used to provide an intelligent assistant able to find or navigate any information (for example electronic manuals or patient records) a worker may need, assist in training of new workers or provide summaries of performed activities.

[Minnen *et al.* 2007b] use a wearable sensing platform to categorize soldier activities. Considering the activities, action reports can automatically be compiled. In addition, the system can help in recalling situations during missions.

Entertainment and gaming Similar to the maintenance area, the industry for entertainment and gaming is getting aware of market potentials of activity recognition. Here, the adoption by users may be faster than in other domains since classification accuracy is less crucial. [Zhang and Hartmann 2007, Heinz *et al.* 2006] attach motion sensors to the body and use it to control video or martial arts games. Nintendo's Wii platform [Nintendo 2009] or the Apple iPhone [Apple Inc. 2009] further increase the popularity of game controls based on wearable sensors.

2.2 Work Directly Addressing the Thesis' Contributions

In the following we report on related work addressing the three contributions of this thesis (Section 1.2). We start with work including segmentation methods of continuous data streams in Section 2.2.1. Low-level gesture and activity recognition methods will be reported in Section 2.2.2 differentiating between gesture recognition, activity spotting, and body-model based approaches. Finally, Section 2.2.3 summarizes research of hierarchical activity recognition.

2.2.1 Segmentation methods of continuous data streams

Segmentation is a crucial step in many application fields, such as segmentation of objects in images [Sklansky 1978, Ida and Samborsugi 1995, Ghosh and Mitchell 2006] or in video [Brand and Kettnaker 2000, Niu and Abdel-Mottaleb 2005], text [Quint 2000, Borkar *et al.* 2001], or words in speech [Milone *et al.* 2002]. The goal of segmentation in the example fields is to separate specific objects from a surrounding context thereby serving as pre-processing for a subsequent classification or detection. The following summary first focuses on segmentation of gestures. Subsequently, we report on approaches including segmentation of human activities.

Segmentation of gestures In [Zinnen *et al.* 2007b], we explore contact free interaction methods as a promising way to ensure sterility guidelines and save time when explicitly interacting with computers in a health care area. We address the problem of contact

free access and browsing of patient documents using gestures to navigate through the documents. Since hand movements and gestures are a natural way of human interaction, the system must distinguish navigation gestures from common ones. The activation of our system is based on face detection. Only if the doctor’s line of sight intersects with the screen, the system is automatically activated. Section 2.4 reports on the system in more detail.

In contrast to our approach, most approaches in gesture recognition [Kela *et al.* 2006, Mäntylä *et al.* 2000, Hofmann *et al.* 1998, Iacucci *et al.* 2004] expect an explicit activation or deactivation of the system while performing a gesture. [Argyros and Lourakis 2006] present a vision-based interface for controlling a computer mouse via 2D and 3D hand gestures. A gesture involving both hands, each of which is presented with five extended fingers activates and deactivates the interpretation of gesture-based commands for the mouse.

For many scenarios, explicit gestures or face detection are not an adequate solution. Whereas face detection can only be used in stationary scenarios, explicit segmentation is obtrusive for users. Although many gesture recognition approaches exist, few deal with the spotting task itself. [Lee and Xu 1996] develop a system for online gesture recognition using hidden Markov models. They are among the first researchers to use segmentation as a pre-processing step to gesture recognition and are able to recognize 14 different gestures online. A similar approach is applied in [Deng and Tsui 2000] to segment and spot hand gestures including an exhaustive search with regard to possible observation sequences.

Segmentation of activities Similar to gestures, short physical activities are usually embedded in a large background data stream. In this section we summarize state-of-the-art approaches to separate activities from irrelevant background data.

In [Lukowicz *et al.* 2004], the authors present a technique to automatically track the progress of maintenance or assembly tasks using body worn sensors. They combine data from accelerometers with frequency sound classification. While performing activities sound can also help to segment the continuous motion stream.

[Minnen *et al.* 2006] step into an approach to segment human activities in a continuous motion stream. A mixture of Gaussian distributions are fitted to accelerometer and gyroscope data. The Gaussian models allow the definition of a symbol sequence. The authors use a minimum description length to detect motifs in the symbol sequences. Without manually constructing detectors, the approach enables the spotting of occurring activities in an unsupervised manner.

[Min and Kasturi 2004] propose an approach deriving motion trajectories of body parts and joints from a video stream. These trajectories are subsequently used to pre-segment the motion of dance movements. Using hidden Markov models, the authors determine the endpoints of potentially detected activities.

Many other approaches introduce segmentation methods for continuous data streams based on hidden Markov models (HMMs) as described in Section 2.3.1. HMMs are well

suited to cope with variability in spatio-temporal patterns, as they occur in human activity recognition.

The authors in [Morguet and Lang 1998b, Morguet and Lang 1998a] examine the normalized output of the Viterbi algorithm, i.e. the probability of the most likely sequence of hidden states that results in a sequence of observed events. For each point in time of an observation sequence, a Viterbi calculation is performed. Characteristic peaks in the output scores of the respective class models indicate the presence of an activity.

A more sophisticated method of using HMMs for segmentation is introduced by [Lee and Kim 1998, Lee and Kim 1999]. The authors train an HMM for each activity and compose a complex ergodic model including all single HMMs. The resulting model represents any sub-pattern that is included in all single HMMs. While performing recognition, the output of the complex model is taken as a reference threshold to interpret the likelihood outputs of the individual activity models.

Like HMMs, dynamic time warping (DTW) is an algorithm for measuring similarity between two sequences which may vary in time or speed. The authors in [Ko *et al.* 2008] use DTW for online gesture recognition. Parts of the continuous data stream are isolated applying a sliding window of fixed window length. The segments of interest are compared with training sequences. The success of this approach strongly depends on the window length. It is not suitable to cope with a large variability in activity execution length.

[Amft *et al.* 2005, Junker *et al.* 2008] present a segmentation and similarity search, an approach to segment motion data in two stages. The first stage calculates features from the motion data stream. Next, the feature stream is pre-segmented by the SWAB (Sliding Window and Bottom-up) algorithm. The algorithm presented by [Keogh *et al.* 2001] combines a sliding window and a bottom-up algorithm that tries to approximate a given time series by piecewise linear segments. In contrast to pure online capable sliding-window algorithms, SWAB is able to find rather good approximations of the stream because the bottom-up algorithm requires to have a global view on the whole time series to produce a good segmentation. While the first stage performs segmentation only, the partitioned data is classified in the second stage applying a similarity search. Each detected segment is a potential area of interest, namely the start and end point of a potential activity. For all derived potential motion segments, a feature vector is calculated and compared to pre-computed training vectors. Depending on a threshold, the segments are assigned to trained classes or discarded.

Many approaches apply HMMs or DTWs for segmentation. All those approaches are computationally expensive because they depend on sliding windows, often with variable window lengths. A high variability in activities' lengths as well as large background data complicate the usage of HMMs and DTWs further. Conceptually, the last approach by [Amft *et al.* 2005, Junker *et al.* 2008] is related to what our approach (Chapter 3 and Chapter 4) can achieve. A notable difference is that our method is based on short and fixed hand positions or turning points in arm movements as they typically enclose human activities. In contrast to our method, the complexity of the SWAB algorithm is higher when applied on multi-dimensional data.

2.2.2 Low-level gesture and activity recognition approaches

This section reports on related work in low-level gesture and activity recognition. Once more, we consider activities of a couple of seconds in large background data as low-level activities. Spotting of low-level activities is the method for detecting and recognizing those low-level activities in the continuous data. First, we step into gesture recognition. Subsequently, we introduce approaches addressing activity spotting in continuous data streams. Finally, related work using body and motion models are outlined.

Gesture recognition The user's gestures seem to be a good alternative to common input devices for explicit interaction. There are different levels of gestures that can be recognized, such as gestures involving the fingers only or gestures that use one or both arms.

Finger-only gestures have received attention because they are versatile and unobtrusive. [Perng *et al.* 1999] fit a glove with acceleration sensors on all fingers to recognize hand-gestures. [Vardy *et al.* 1999, Rekimoto 2001] use wrist-mounted sensors to recognize a special finger alphabet.

Gestures involving entire arms have received attention in other scenarios. [Starner *et al.* 1998] use a hat-mounted camera to recognize gestures of American sign language with the goal of translating them automatically into spoken language for deaf-mute people. [Brashear and Starner 2003] extend this approach by adding acceleration sensors on the arms to distinguish gestures that are not discriminable from video alone.

[Mäntylä *et al.* 2000] show an approach for enhancing customization in accelerometer based gesture interaction. [Kela *et al.* 2006] propose a system offering freely trainable gesture commands. Using those gestures, users can control external devices with a hand held wireless accelerometer sensor unit. Acceleration based gesture recognition using hidden Markov models are studied in [Pylvänäinen 2005, Kallio *et al.* 2006, Mäntylä *et al.* 2000].

Activity spotting The focus of this thesis is the recognition of short activities that happen in a short time frame (spotting). The activities usually occur occasionally in between long periods of unrelated, arbitrary events. This paragraph summarizes approaches dealing with the spotting of activities defined by short sequences of arm movements amid irrelevant data.

[Stiefmeier *et al.* 2008, Ogris *et al.* 2008, Lukowicz *et al.* 2004, Stiefmeier *et al.* 2006] use information gathered from wearable and environmental sensors for tracking activities of workers in car manufacturing plants, for example to provide realtime feedback to the worker about upcoming assembly steps or to issue warnings when procedures are not properly followed.

[Patterson *et al.* 2005] present results related to achieving fine-grained activity recognition for context-aware computing applications. The authors examine the advantages

and challenges of reasoning with globally unique object instances detected by an RFID glove.

[Amft *et al.* 2005] detect arm gestures related to typical meal intake. Information retrieved from such a system can be used for automatic dietary monitoring in the domain of behavioral medicine.

[Junker *et al.* 2008] present a method for spotting sporadically occurring gestures in a continuous data stream from body-worn inertial sensors. The method is based on a natural partitioning of continuous sensor signals and uses a two-stage approach for the spotting task. The approach is verified on two scenarios that together include nearly a thousand relevant gestures. Study 1 evaluates the interaction with different everyday objects. In study 2, the consumption of nutrition is analyzed.

[Kern *et al.* 2003] look at activities, such as keyboard typing, writing on a white-board and shaking hands. [Cakmakci *et al.* 2002] try to identify when a person was looking at the watch.

Body-model and primitives Several approaches for human motion modeling in activity recognition can be found in the literature. The human activity language (HAL) [Guerra-Filho and Aloimonos 2007] aims for a modeling of human motion similar to linguistics. Another approach uses context-free grammars to model complex human activities such as interactions with objects. The activities are represented by composite actions and those in turn by atomic actions [Ryoo and Aggarwal 2006].

[Kojima *et al.* 2002] propose a method for describing human activities from video images based on concept hierarchies of actions. In general, the concepts of events or actions of humans can be classified by semantic primitives. By associating these concepts with the semantic features extracted from video images, appropriate syntactic components such as verbs or objects are determined and then translated into natural language sentences.

Similar to a decomposition of speech into phonemes, the authors in [Green and Guan 2004] decompose human motion into *dynamemes* as units of activities. The dynamemes are derived from a sequence of motion vectors in the video stream using particle filtering.

[Ali and Aggarwal 2001] segment a continuous human activity into separate actions and correctly identify each action. The authors compute the angles between three major components of the body and the vertical axis, namely the torso, the upper component of the leg and the lower component of the leg. Using these three angles as a feature vector, frames are classified into breakpoint and non-breakpoint frames. Breakpoints indicate an action's start or termination.

The Moven [Moven 2009] motion capture suit is an easy to use system for full-body human motion capture. Moven is based on unique, state-of-the-art miniature inertial sensors [XSens 2009], biomechanical models and sensor fusion algorithms.

While body-models are proposed for motion capturing by [Moven 2009] and are used for activity recognition in computer vision [Ryoo and Aggarwal 2006, Kojima *et al.* 2002, Green and Guan 2004, Ali and Aggarwal 2001], we are not aware of similar work detecting body-model derived primitives in the area of activity recognition using body worn sensors.

2.2.3 Hierarchical activity recognition approaches

A large part of research in activity recognition focuses on rather low-level and short-term activities. Many applications however require the analysis and recognition of more complex and composite activities. Despite their importance, comparatively little work has dealt with composite or high-level activities due to the increased complexity and involved difficulties.

[Liao *et al.* 2007b] use information from GPS sensors to construct models of high-level activities such as work or leisure and to identify important places. Similarly, [Krumm and Horvitz 2006] use location sensors to make high-level predictions about driving destinations.

[Wang *et al.* 2007] propose to break down activities such as cleaning windows into small movements called actions, such as wipe horizontally and wipe vertically. [Lühr *et al.* 2003] present an approach to learn the hierarchical structure of human action sequences in a home setting based on hierarchical hidden Markov model. [Clarkson and Pentland 1999] use wearable vision and audio sensors to recognize scenes such as a user visiting a supermarket or a video store. [Oliver *et al.* 2002] propose a layered probabilistic representation of activities using hidden Markov models when modeling an office awareness application.

[Antifakos *et al.* 2002] propose a framework for proactive guidance aiming to overcome limitations of printed instructions. By attaching multiple sensors onto parts of the assembly, the system can recognize the actions of the user and determine the current state of the assembly.

[Shi *et al.* 2004] present propagation networks, an approach to represent and recognize sequential activities that include parallel streams of action. Each activity is represented using partially ordered intervals. Each interval is restricted by both temporal and logical constraints, including information about its duration and its temporal relationship with other intervals. The temporal and logical relationships are prescribed by a knowledge engineer. A system that can do this requires models of the activities of interest.

[Perkowitz *et al.* 2004] show how to mine definitions of activities such as cooking pasta or watching a video in an unsupervised manner from the web. RFID sensors allow to formulate activity models by translating labeled activities, such as ‘cooking pasta’, into probabilistic collections of object terms, such as ‘pot’. Their approach is complementary to the one proposed in Chapter 7 of this thesis as such text mining could be used to automatically obtain prior knowledge and task models used in our approach.

[Huynh *et al.* 2008] propose a novel method to recognize daily routines as a probabilistic combination of activity patterns. The use of topic models enables the automatic discovery of such patterns in a user’s daily routine.

Although in many applications the analysis and recognition of high-level and longer-term activities is an important component, there is not much work addressing the specific challenge of recognizing high-level activities allowing a high variability in temporal ordering. In computer vision, there exists some prior work that can deal with this kind of relationship (for example [Xiang 2003, Pinhanez 1999]). Particularly in the area of wearable context research, very little work exists addressing hierarchical activity recognition, especially focusing on methods that reduce the amount of training data.

2.3 Machine Learning

This section explains learning methods that are used in the thesis. In general, one can distinguish between *supervised* or *unsupervised* as well as *discriminative* or *generative* learning methods.

Unsupervised learning [Clarkson and Pentland 1999, Liao *et al.* 2007a, Patterson *et al.* 2004, Minnen *et al.* 2007a, Huynh and Schiele 2006b] aims to directly construct models from unlabeled data. The algorithms either estimate the properties of its underlying probability density or by discovering groups of similar examples. By contrast, supervised learning [Van Laerhoven *et al.* 2003, Tapia and Intille 2007, Bao and Intille 2004, Ward *et al.* 2006, Ogris *et al.* 2008] requires labeled data on which an algorithm is trained. In this thesis, we exclusively use supervised approaches.

Generative approaches infer the class-conditional distributions $p(\mathbf{x}|C_i)$ of the input data \mathbf{x} given class C_i . Together with an estimate of the prior class probabilities $p(C_i)$, the posterior class probabilities $p(C_i|\mathbf{x})$ can be determined via Bayes’ theorem.

$$p(C_i|\mathbf{x}) \propto p(\mathbf{x}|C_i)p(C_i). \quad (2.1)$$

Examples of generative models include Gaussian distribution or mixture model, hidden Markov model (see Section 2.3.1) or naive Bayes. Naive Bayes classifiers deal with a simple probabilistic classifier based on applying Bayes’ theorem with strong independence assumptions. A naive Bayes classifier assumes that a particular feature of a class is unrelated to any other feature. In spite of this simplifying and often unrealistic assumption, naive Bayes classifiers often work well in many complex real-world situations. Particularly, naive Bayes classifiers tend to require relatively small amounts of training data to estimate the parameters (for example means and variances of the variables) necessary for classification. Because independent variables are assumed, only the variances of the individual variables need to be determined and not the entire covariance matrix.

Discriminant approaches try to directly solve the problem of determining the posterior class probabilities $p(C_i|\mathbf{x})$ without modeling the class-conditional densities $p(\mathbf{x}|C_i)$. Main focus of discriminant approaches is on learning the class decision boundaries, and not on modeling the properties of the individual classes. Often, the classification performance of discriminant approaches is superior to those obtained by generative models [Jaakkola and Haussler 1998, Ng and Jordan 2002]. Examples of discriminative models used in machine learning include support vector machines, (joint) boosting [Friedman *et al.* 2000, Torralba *et al.* 2007], neural networks or conditional random fields. This thesis applies joint boosting as explained in Section 2.3.2.

2.3.1 Hidden Markov model (HMM)

Hidden Markov models (HMMs) [Rabiner 1989] are especially known for their application in temporal pattern recognition such as speech, activity or gesture recognition. The hidden Markov model is a statistical tool for modeling a wide range of time series data. In a regular Markov model, the state is directly visible to the observer. Therefore the state transition probabilities are the only parameters. In a hidden Markov model, the state is not directly visible. Instead, the visible output depends on the hidden state. Note that modeled data of an HMM is assumed to be a Markov process with unknown parameters. Each state has a probability distribution over the possible output variables. Therefore the sequence of variables generated by an HMM gives information about the sequence of states. An HMM is characterized by the following parameters:

1. N, the number of states in the model. Although the states are hidden, for many applications there is some physical significance attached to the states. The individual states are denoted as $S = \{S_1, S_2, \dots, S_N\}$, and the state at time t as q_t .
2. M, the number of distinct observation symbols per state. The observation symbols correspond to the physical output of the system being modeled. We denote the individual symbols as $V = \{V_1, V_2, \dots, V_M\}$.
3. The state transition probability $A = \{a_{ij}\}$ where

$$a_{ij} = P(q_{t+1} = S_j | q_t = S_i), 1 \leq i, j \leq N.$$

4. The observation symbol probability in state j , $B = \{b_j(k)\}$, where

$$b_j(k) = P(v_k \text{ at } t | q_t = S_j), 1 \leq j \leq N, 1 \leq k \leq M.$$

5. The initial state $\pi = \{\pi_i\}$ where

$$\pi = P(q_1 = S_i), 1 \leq i \leq N.$$

An HMM requires the specification of two model parameters M and N , specification of observation symbols, and the specification of three probability measures A, B , and π .

$$\lambda = (A, B, \pi)$$

Given an observation sequence $O = O_1 O_2 \dots O_t$ and a model, the forward algorithm computes the probability that the observed sequence was produced by the model. The Viterbi algorithm attempts to uncover the hidden part of the model and finds the optimal state sequence given an optimality criterion. The Baum-Welch algorithm [Dempster *et al.* 1977] is a common method to adjust the model parameters. A detailed description of the algorithms can be found in [Rabiner 1989].

There are several types of HMMs for diverse applications. In ergodic or fully connected HMMs, every state of the model can be reached from every other state of the model. Left-right models are characterized by an underlying state sequence having the property that as time increases, the state index increases. In addition, there are many possible variations and combinations as summarized in [Rabiner 1989].

2.3.2 Boosting and joint boosting

Boosting (for example [Friedman *et al.* 2000]) is a widely used state-of-the-art machine learning technique for classification that has been applied to many different tasks such as camera-based object detection [Viola and Jones 2004]. The basic idea is to combine multiple weighted weak classifiers $h_m(x)$ in order to form a strong classifier H . Often, decision tree stumps and regression stumps on a single dimension d of the feature vector x_m^d are used as weak classifiers. In this work we employ regression stumps defined as:

$$h_m(x) = \begin{cases} a_m & \text{if } x_m^d > \theta_m \\ b_m & \text{if } x_m^d \leq \theta_m \end{cases}$$

with θ_m being an automatically determined threshold. [Torralba *et al.* 2007] extend the idea of boosting to multi-class problems in a way that weak classifiers can be shared across multiple classes while the strong classifiers are learned jointly.

Thus, contrary to binary boosting the weighted error is not only sought to be reduced for a single strong classifier in each round, but for a whole set of strong classifiers. However, the number of possible sharing sets is 2^C where C is the number of overall classes. Clearly, an exhaustive search over all combinations even for a moderate number of classes is infeasible. As proposed by [Torralba *et al.* 2007] we adopt a greedy best first search strategy to obtain an approximation to the best sharing set, which reduces complexity to $O(C^2)$. Torralba *et al.* show that surprisingly few weak classifiers are required for joint boosting to obtain the same classification performance compared to independently learned classifiers. Hence, classification runtime is reduced, in particular, when the number of classes becomes large.

More formally, in each boosting round m joint boosting selects the weak classifier which reduces the weighted squared error most for a set of classes S_m . The final strong classifier for a class c given a feature vector x is:

$$H_c(x) = \sum_m h_m(x) \delta(c \in S_m)$$

2.4 Case Study on Gesture Interaction During Clinical Ward Rounds

The digitalization of the health care sector aims to minimize the use of paper in hospitals. Hence medical staff needs new interaction methods to access electronic information efficiently and effectively. To reduce risk of infection, doctors must sterilize their hands both between getting in contact with different patients but also when they contact input devices such as a mouse or keyboard. Since sterilization is time consuming, contact free interaction methods are a promising way to ensure sterility guidelines and save time.

2.4.1 Introduction

In cooperation with an Austrian hospital, we designed and developed a novel gesture interaction system for the ward round that supports mobility and preserves sterility through contact-free interaction. The emphasis is to provide doctors with a contact free, mouse-like document browser that requires as little training as possible. Therefore the doctors are equipped with a small wristband sensor board. The wrist placement of the sensor keeps doctors' hands free and prevents contamination. For example to scroll a document list the user moves the arm up and down and the mouse pointer position on the screen changes and follows the arm movements. The direct feedback reduces errors because users can correct imprecise movements immediately.

Figure 2.1 (left) shows a highlighted entry of a document list that is selected with the wrist-band controlled mouse pointer. Actions like open/close or scroll up/down are triggered by keeping the pointer on a specific area for a fixed period of time. Figure 2.1 (right) contains arrows and a circle as triggering areas. By keeping the mouse pointer for a fixed time on those areas, the browser performs the corresponding action (scroll up/down, close). The general idea of simulating a mouse supports most features without demanding the user to learn new gestures.

A recent workshop shows that even unskilled users can use the system with little training effort. Due to the mouse-like interaction, the users felt very comfortable and certain using the system. Since hand movements and gestures are a natural way of human interaction, the system must distinguish navigation gestures from common ones. An activation and deactivation of the gesture interaction becomes necessary. There are different ways for activation and deactivation:

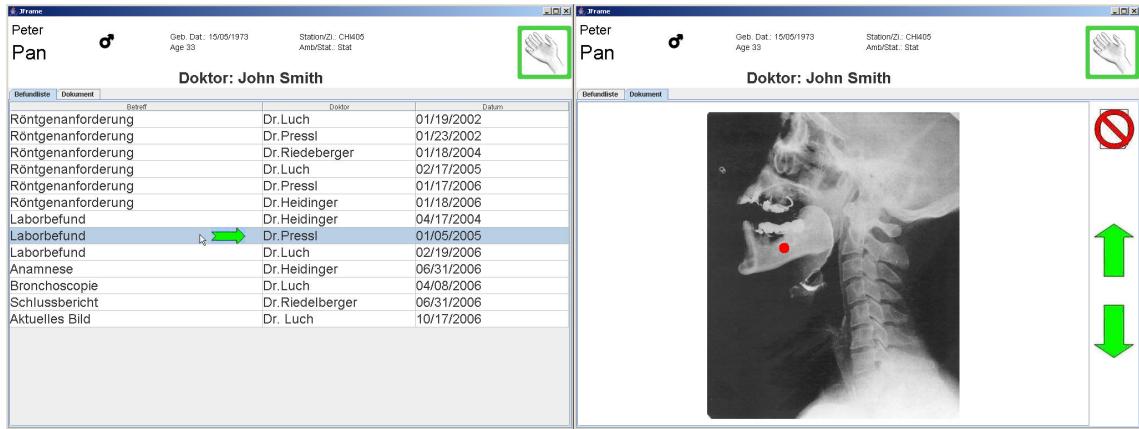


Figure 2.1: List view (left) and document view (right) of the provided touchless document browser as applied in the ward round scenario. Actions like open/close or scroll up/down are triggered by keeping the mouse pointer on a specific area for a fixed period of time. By keeping the mouse pointer for a fixed time on those areas, the browser performs the corresponding action (scroll up/down, close).

1. Explicit gestures for activation and deactivation. Those need to be rather complex and should not occur within daily gestures to avoid accidental (de)activation.
2. When using implicit gestures for interaction, the system is always active. All used gestures must be complex and distinguishable from daily gestures, not only those for activation and deactivation.
3. An external system like face detection is used for activation and deactivation of the system.

Complex gestures for activation and deactivation will not be accepted by the doctors because of the training effort. Therefore, explicit and implicit gestures do not seem to be an adequate solution in our scenario. Furthermore, end users often forget to explicitly deactivate the system which causes accidentally performed actions.

Our activation is based on face detection. Therefore, we place a WebCam on the screen to find out if the doctor's line of sight intersects with the screen. Figure 2.2 (left) illustrates the intersection by a square around the detected face. The system only needs to be active if the doctor navigates within the documents or reads a file. While the doctor is talking to the patient or the nurse, the system should/can be deactivated. Even fast task switches in the patient doctor interaction are feasible with less interaction steps of the user. Preliminary tests illustrate the active area of the face detection (see Figure 2.2 (middle)). The tests were performed with 8 independent persons including 2 women and 6 men of different sizes and under changing angles towards the camera. Faces are reliably detected in a minimum area spanned by an angle of 50° and a distance of 140cm from the camera. Based on observations at the ward round work flow in the hospital, the area where doctors stay while interacting with the patient will often be covered by this active area.

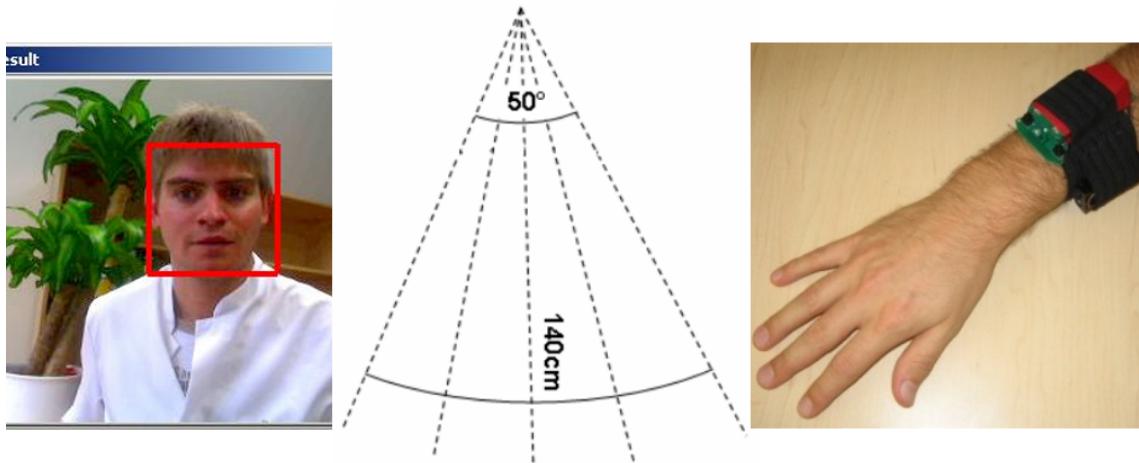


Figure 2.2: Left: Activation of the gesture interface using face detection. When the doctor's line of sight intersects with the screen (square), the system is activated. Middle: Preliminary results of the active area of the face detection. Right: Wireless sensor with 5m range can be worn as a wristband.

Extending the face detection by face recognition could distinguish the doctor's attention from nurses' or patients' towards the screen.

2.4.2 Low cost sensors & software

Low Cost Sensors. The *rotation sensing platform* is built with the hardware of a common gyro mouse [Gyration 2009]. The device contains a gyroscope for rotation measurement and radio frequency (RF) technology for connection to a PC. In order to decrease the size of the device, we slightly modified the original circuit board of the air mouse. As illustrated in Figure 2.2 (right), the sensor with 5m range can be worn as a wristband. For *face detection*, we use a Logitech Quick Cam for Notebooks that is placed on top of the bedside display. A *bedside display* is fixed to browse patient records.

Software. Our approach is based on a commercial gyration mouse driver [Gyration 2009]. The software can be used without any adaptation to the modified scenario. The software uses the rotation sensor data as input for mouse navigation on the screen. The (de)activation of the system is based on a face detection algorithm proposed by Viola & Jones [Viola and Jones 2004] and provided by the OpenCV library [OpenCVLibrary 2009]. If a face is detected in the picture, the program will send an activation command in less than 300ms to the user interface. Vice versa, frames without a face detection lead to a deactivation of the system.

The GUI for browsing documents consists of a static header for patient information like name or age and a body with two different views. The list view displays a list of

all available documents for the current patient. Opening a document will switch to the document view displaying a picture or document. Closing the document will switch to the list view again. A picture of a green or red hand in the upper right of the display signs the current activation status of the system. We have developed a system that can easily be extended with further functionality without training effort for the user, for example zooming into a picture. Only a new button has to be added invoking a zoom function.

2.5 Discussion

This chapter gives an overview of current trends and the state-of-the-art in context awareness and activity recognition. Although current research in the corresponding domains is characterized by a wide range of topics and applications, previous work has aimed to address this thesis' challenges only individually.

We are not aware of many approaches addressing user-independent spotting of activities that are characterized by a high variance in performance. Furthermore, only a few approaches deal with multiple types of activities. Although many applications however require the analysis and recognition of more complex and composite activities, comparatively little work has dealt with composite or high-level activities due to the increased complexity and involved difficulties. Particularly in the area of wearable computing, very little work exists addressing hierarchical activity recognition. Many approaches for segmentation of a continuous data stream are computationally expensive because they depend on sliding windows, often with variable window lengths. Other approaches are characterized by a high complexity when applied on multi-dimensional data.

Embedded in different application scenarios, the remaining chapters will address these under-explored areas. As mentioned before, a particular focus is placed on multiple types and diversity of activities, high variance in performance and user independence, continuous data stream with large background and activity recognition on different levels.

3

Recognizing Short and Non-Repetitive Activities from Wearable Sensors

Many interesting activities, for example those in the maintenance domain, are non-repetitive and happen in a very short time frame of only a few seconds. To give a reasonable estimation of whether the maintenance worker has finished a task such as performing an oil check, the system needs to identify sub-activities like opening the oil container, pushing or pulling the oil stick, and opening or closing the hood. Both the shortness and the individual performance of the activities make the recognition a highly challenging task. In addition, the activities may only occur occasionally in between long periods of unrelated, arbitrary events. This chapter presents a method for short-term activity recognition. Postures are used in order to identify segments of interest in a continuous data stream and for the subsequent classification of 10 short activities.

3.1 Introduction

This chapter pursues a recognition of activities that are defined by short sequences of arm movements amid irrelevant data. For recognition, one sensor (Section 3.2.2) placed at the user's dominant wrist is used.

We present a method for activity recognition using start and end postures (short fixed positions of the wrist) in order to identify segments of interest in a continuous data stream. Thereby, we focus on spotting a subset of activities that require interaction with objects, for instance the knob when turning on the heating. We observed that such activities can be separated into three segments: First, the user keeps the arm fixed for a very short time when grabbing the object. Second, the user moves the arm and therefore changes the position of his/her arm while performing the activity. Third, the user's arm has reached an end position for a short time due to the affordances of the respective object. It can be observed that the fixed positions before and after the movement are similar for different users.

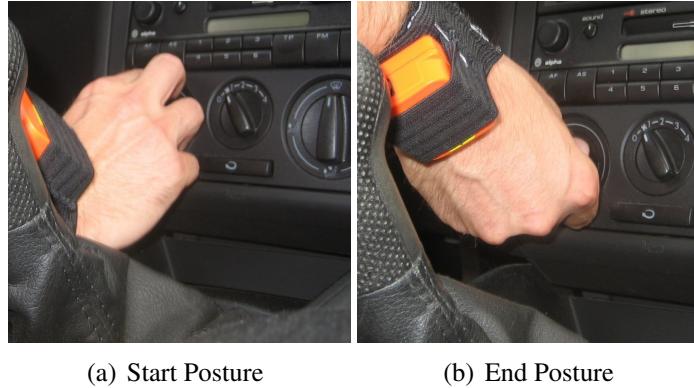


Figure 3.1: Start and end postures surrounding the activity turning on the heating. Considering these postures, the activities can be separated into three segments. The later classification is performed on segments enclosed by postures.

Example: Figures 3.1(a) and 3.1(b) illustrate the short fixed positions of the arm before and after turning on the heating of a car. In Figure 3.1(a), the user keeps the arm fixed for a very short time when grabbing the heating knob. After turning the arm, the user's arm reaches an end position for a short time due to the affordances of the knob as illustrated in Figure 3.1(b).

The rest of this chapter is organized as follows. Section 3.2 describes the experimental setup including the recorded data sets and the applied hardware. In Section 3.3, existing approaches using hidden Markov models (HMMs) for spotting relatively short activities are applied. Fully-connected as well as left-right hidden Markov models are trained for ten activities on hand segmented training data. Whereas HMMs achieve high recognition rates on hand segmented test data, the models cannot easily be transferred to the continuous test data for short activity recognition.

The main contribution of this chapter is introduced in Section 3.4. A new classification method is described using short fixed positions of the subject's hands (*postures*) to extract and identify possible segments for a specific activity in the continuous data stream. We will show in a preliminary study that the subjects' fixed arm positions before and after performing an activity can differentiate the short activities from the rest of the data surprisingly well. Classifiers are defined using features based on the start postures, on the end postures, as well as on the difference between both.

In the experiments, we compare the performance of different classifiers and show that fusing several features on postures attains the best recognition results. Since many segments of the data can be discarded by calculating the low-cost posture-based features, the approach reduces the computing requirements over the data stream. Therefore it will allow for a more costly classification over the segmented parts in the following chapters.

We close the chapter with a summary and conclusion of the main contributions in Section 3.5.

3.2 Experimental Setup

Section 3.2.1 describes a small data set of 10 car-related activities. The data set is used in this chapter to evaluate the feasibility of our classification method based on a segmentation of the continuous data stream and posture-based features. Subsequently, Section 3.2.2 briefly introduces the MTx sensor [XSens 2009]. The inertial measurement unit is used in all data recordings of this thesis. It provides 3D orientation as well as kinematic data: 3D acceleration, 3D rate of turn and 3D earth-magnetic field.

3.2.1 Data set & annotation

For the initial experiments reported in this chapter, a data set of 10 different activities performed by one subject was recorded. The test and training data were recorded on two different days. While recording the test data, the subject was given a script containing 10 very short activities in a car context: Turning the heating on/off, pulling and releasing the handbrake, opening and closing the sun shield, hood and oil container. The subject recorded these activities during driving and while performing an oil check. Later, the data was annotated using recorded audio data. The data was recorded with one XSens MTx sensor (Section 3.2.2) placed at the subject's right wrist. The 10 activities were repeated 15 times in different sequences, and interrupted by unrelated activities such as reading a map, driving or browsing the glove locker and trunk. The overall test data set has a length of about 18 minutes with a sample frequency of 100 Hz. The longest activities were opening and closing the hood with a maximum length of 2.5 seconds. Opening the sun shield was the shortest activity with a maximum length of 0.5 seconds. Since the performance of most considered activities takes less than one second, the data of interest makes up less than 2 minutes of the overall data. Training data was recorded on a different day, and the subject performed all activities ten times. Using the training data, a minimum and maximum length of the recognized activities was defined. The values are summarized in Table 3.1.

3.2.2 Hardware - MTx inertial measurement unit (IMU)

The MTx [XSens 2009] is an inertial measurement unit (IMU) that integrates a 3-dimensional magnetometer as well as calibrated 3-dimensional linear acceleration, rate of turn and magnetic field data. A sensor fusion algorithm is used which allows the MTx to accurately calculate absolute orientation in three-dimensional space by integrating gyroscopes, accelerometers and magnetometers in real-time. The orientation as calculated by the MTx is the orientation of the sensor-fixed co-ordinate system with respect to a Cartesian earth-fixed co-ordinate system.

The MTx is used for all data recordings that are considered in this thesis. Here, in Chapter 3, one sensor at the user's right wrist is used. Because of magnetic disturbances

Activity	Minimum Length[s]	Maximum Length[s]
Open Hood	0.6	2.5
Close Hood	0.8	2.2
Open Oil container	0.2	0.6
Close Oil container	0.2	0.6
Open Sun shield	0.2	0.5
Close Sun shield	0.2	0.7
Pulling Handbrake	0.2	0.6
Releasing Handbrake	0.2	0.6
Heating on	0.2	0.7
Heating off	0.5	1.2

Table 3.1: Minimum and maximum times for the considered activities in the training data. The training data contains ten performances of all activities by one subject.

in the corresponding car scenario, any data from the magnetometers is completely disregarded. Therefore, we do not obtain an absolute orientation towards the magnetic north. The provided orientation data can still be used to reliably calculate the position of the sensor towards the gravity giving a strong feature for the position and posture of the user's arm. The data set of explicit and implicit gestures in Chapter 4 considers relative arm movements estimated from the sensor's absolute orientation towards the magnetic north. Again, one sensor is attached to the user's right wrist using a wireless version of the MTx sending the data via Bluetooth to a computer. The research in the remaining chapters (5, 6, 7) is based on five XSens located at the users' upper arms, lower arms and torso. Based on the orientation of the five sensors, a body-model can be estimated as described in the corresponding chapters.

3.3 Activity Recognition with Hidden Markov Models

Section 2.3.1 introduces hidden Markov models (HMMs) as statistical models in which the modeled data is assumed to be a Markov process with unknown parameters. During learning, the parameters are estimated from the observations. Hidden Markov models [Rabiner 1989] are widely adopted for gesture and speech recognition. Their intrinsic properties make them especially attractive for gesture recognition. In this section, hidden Markov models are applied to relatively short activities in a longer recording. In this thesis, we use the hidden Markov model toolbox for Matlab written by Kevin Murphy [Murphy 2009].

HMMs trained on segmented data are first applied to hand-segmented test data (Section 3.3.1). Whereas the results using fully-connected HMMs are not sufficient, left-right HMMs are better suited for the considered activities. Then we analyze the suitability of HMMs for spotting activities in continuous data streams (Section 3.3.2).

3.3.1 HMMs on hand segmented activities

For each activity, we train a fully-connected HMM and a left-right HMM with three states and single Gaussian distribution. As features, we consider the three dimensional acceleration values as well as the sensor's orientation towards gravity. Since the activities in the training data are annotated, we can directly calculate all features on the positive training segments. The resulting feature vectors can be used as direct input for the training phase of the HMMs. All models are trained on ten hand-segmented instances of the corresponding activity in the training data.

This section evaluates the suitability of HMMs to distinguish activities from each other. Therefore, only segments in the continuous data including activities are considered in the following evaluation. These segments are hand-annotated. The feature calculation for the test data is similar to the training data. For all activity instances in the test data, we can directly calculate the features. Thereafter, we compute a probability for all models that the observed sequence was produced by the corresponding model. Maximum likelihood classification returns the most probable fully-connected and left-right model for the sequence. The results of the fully-connected and the left-right models are described below.

Fully-connected HMMs on hand segmented activities The overall recognition rate of the activities is 80%. The activities *open hood* and *open oil container* have a low recall under 50%. Furthermore, *close hood* and *close/open oilcontainer* have a precision under 70%. Good results can be obtained for *releasing handbrake* and *heating on/off*. Both recall and precision are over 93% with fully connected HMMs. *Open hood* and *opening oil container* are often confused: 7 out of 15 times, the system recognizes *close hood* instead of *open hood*. 12 out of 15 times, the system recognizes *close oil container* instead of opening it. Table 3.2 summarizes the confusion of the activities.

	OH	CH	COC	OOC	HON	HOFF	BON	BOFF	OSS	CSS	Rec	Prec
Open Hood:OH	7	7	0	0	0	0	0	1	0	0	0.47	0.88
Close Hood:CH	0	15	0	0	0	0	0	0	0	0	1	0.68
Close Oil Cont:COC	0	0	13	2	0	0	0	0	0	0	0.87	0.5
Open Oil Cont:OOC	0	0	12	3	0	0	0	0	0	0	0.2	0.6
Heating On:HON	1	0	0	0	14	0	0	0	0	0	0.93	1
Heating Off:HOFF	0	0	1	0	0	14	0	0	0	0	0.93	1
Brake On:BON	0	0	0	0	0	0	15	0	0	0	1	0.71
Brake Off:BOFF	0	0	0	0	0	0	0	15	0	0	1	0.94
Open Sun Shield:OSS	0	0	0	0	0	0	2	0	13	0	0.87	1
Close Sun Shield:CSS	0	0	0	0	0	0	4	0	0	11	0.73	1

Table 3.2: Confusion matrix for ten short activities on hand segmented data using fully-connected HMMs with three states and single Gaussian distribution. The average recognition rate is 80%.

Left-right HMMs on hand segmented activities Left-right models perform better than fully-connected models on the test data. On average, the recognition of all 10 activities

improves (see Table 3.3). The average recognition rate for all activities is 88%. The recall increases or stays at the same level except for *close sun shield*. Especially for *open hood* and *open oil container*, a higher recall can be observed. *Close hood* and *close/open oil container* have a higher precision compared with the fully connected results. Apart from minor exceptions, we observed that both the precision and the recall are better for left-right HMMs. Due to higher precision and recall, we apply the same left-right models to the continuous data stream in the next section.

	OH	CH	COC	OOC	HON	HOFF	BON	BOFF	OSS	CSS	Rec	Prec
Open Hood:OH	10	3	0	0	0	0	1	1	0	0	0.67	0.77
Close Hood:CH	0	15	0	0	0	0	0	0	0	0	1	0.83
Close Oil Cont:COC	0	0	15	0	0	0	0	0	0	0	1	0.83
Open Oil Cont:OOC	3	0	3	9	0	0	0	0	0	0	0.6	1
Heating On:HON	0	0	0	0	14	0	0	1	0	0	0.93	1
Heating Off:HOFF	0	0	0	0	0	15	0	0	0	0	1	1
Brake On:BON	0	0	0	0	0	0	15	0	0	0	1	0.71
Brake Off:BOFF	0	0	0	0	0	0	0	15	0	0	1	0.83
Open Sun shield:OSS	0	0	0	0	0	0	0	0	15	0	1	1
Close Sun shield:CSS	0	0	0	0	0	0	5	1	0	9	0.6	1

Table 3.3: Confusion matrix for ten short activities on hand segmented data using left-right HMMs with three states and single Gaussian distribution. The average recognition rate is 88%.

3.3.2 Left-right HMMs on continuous data

The results in the last section have shown that left-right HMMs are proper for the considered activities in the segmented test data. In this section, we analyze the suitability of left-right HMMs for spotting activities in continuous data streams. The training of the left-right models is similar to the training for segmented test data as explained in Section 3.3.1. For each activity, we train a left-right HMM. Once again, we consider the three dimensional acceleration values as well as the sensor's orientation towards gravity as features.

The calculation of the features for the test data is more complex. We apply the left-right HMMs directly to sliding windows over the continuous data. Note that the window length is different for each activity. In the following analysis, the maximum length of each activity (see Table 3.1) is taken as window width. For each point in time and for all trained HMMs, the log-likelihood is calculated over sliding windows.

Evaluation criteria

As mentioned before, we calculate the log-likelihood on sliding windows for all trained HMMs and for each point in time. The subsequent step smooths the log-likelihood curve with a smoothing window length equal to the corresponding maximum activity length.

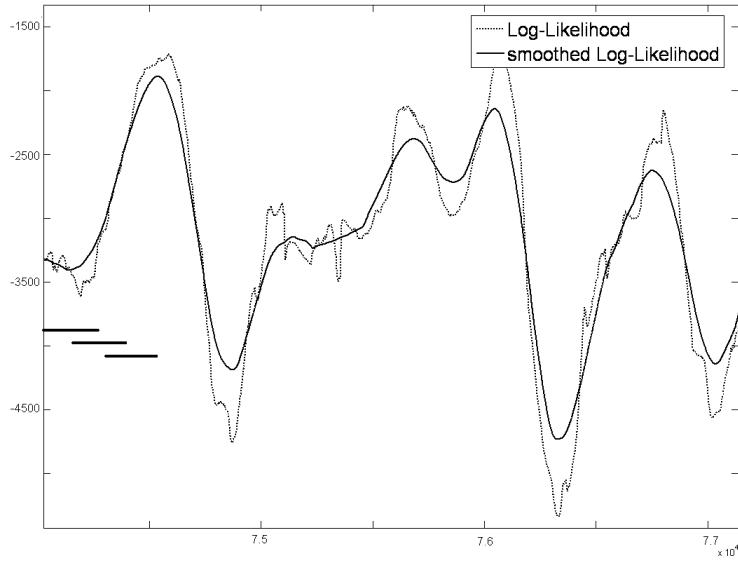


Figure 3.2: Log-likelihood and smoothed log-likelihood for activity *open hood* over the continuous data stream. The left-right HMM is directly applied to sliding windows. In addition, three example evaluation windows are illustrated.

Figure 3.2 illustrates the log-likelihood (dashed curve) and the smoothed log-likelihood (solid curve) for activity *open hood*.

For evaluation, we count the activities' detections for a specific threshold as follows. We set the *tolerance* length for evaluation to the corresponding maximum activity length. An evaluation window of *tolerance* length slides over the smoothed data. The sliding step size is half of the *tolerance* length. Figure 3.2 illustrates three example evaluation windows. If the current evaluation window contains a smoothed value higher than the threshold, we count it as detection. If, for a detection, the center of the evaluation window and the center of an annotation are closer to each other than half the *tolerance*, the detection will be counted as a *true positive*. Otherwise, we count the detection as a *false positive*. For a varying threshold and for each activity, the *true* and *false positives* are analyzed and evaluated in precision-recall characteristics.

Results of left-right HMMs on continuous data

In this experiment, the left-right HMMs trained on labeled data are applied to fixed sliding windows. The HMMs do not yield high recognition rates on the data set. Contrary to the segmented case where the HMMs were able to classify the activities in a satisfying manner, the number of false positives in the continuous case increases tremendously. For all activities, the precision drops when increasing the threshold for detections. *Close sun shield* performs best among all activities. Nevertheless, for a recall of 10%, the precision is already less than 10% and even drops for higher recall. For activities with a high

variance in acceleration like *pulling/releasing hand brake*, the precision even drops and stays under a value of 1% for a recall higher than 0%.

In general, we observe a low precision when aiming for high recall. The results indicate that HMMs may not be adequate models for recognition of short activities in continuous data. Tuning of the parameters for modeling of states, transitions or distribution as well as an extension of the features would probably lead to slightly better results. However there still remains a high modeling effort since the shortness and variance of the activities require different parameters for each activity. Section 6.5 will report on a more detailed analysis and the results when applying HMMs on a different data set. In the next section, we will introduce a new method using postures with the ability to identify short activities in longer recordings.

3.4 Segmentation and Classification Using Postures

As expected and as shown in the previous section, HMMs alone applied to sliding windows are not sufficient for the recognition of short activities. This section introduces a new approach using users' *postures* as short fixed positions of the body or of body parts to differentiate short activities from unimportant data in a continuous data stream. As mentioned before, in many cases the user keeps the arm still for a very short time before and after performing activities.

On one side, we use postures to extract data segments from the continuous data (Section 3.4.1). On the other side, we illustrate for these segments that postures themselves can be used to discriminate short activities from unrelated data (Section 3.4.2). The first experiment compares the distinctiveness of the posture before and after the activity as well as the distinctiveness of the difference in orientation of both posture vectors. Each of the classifiers using the postures and the difference in orientation performs on average better than the approach applying HMMs to a sliding window. Subsequently, the results can be further improved by combining the three classifiers.

3.4.1 Segmentation of the continuous data stream

Fixed postures can be used for a reliable segmentation of the data stream. As mentioned before, many object related and human short term activities are characterized by movements which start and end by very brief fixed positions of the body or of body parts. We call these fixed positions at the beginning and the end of a movement *postures*. First, the data is grouped in regions where the user either moves the arm (movement) or keeps the arm position still (posture). The grouping is based on the variance of the current sensor orientation. If the sensor's orientation changes quickly, the variance is high and a movement will be recognized. If the sensor is kept still, low variance indicates a posture. Using those regions of movements and postures, we build data segments that are enclosed

by two separate postures. The classification can be performed on those segments. Before the classification, overlapping segments have to be discarded to ensure a correct recognition evaluation. The following paragraphs go into more detail regarding the segmentation algorithm and elimination of overlapping segments. Section 3.4.2 describes in detail how the classification is done using postures.

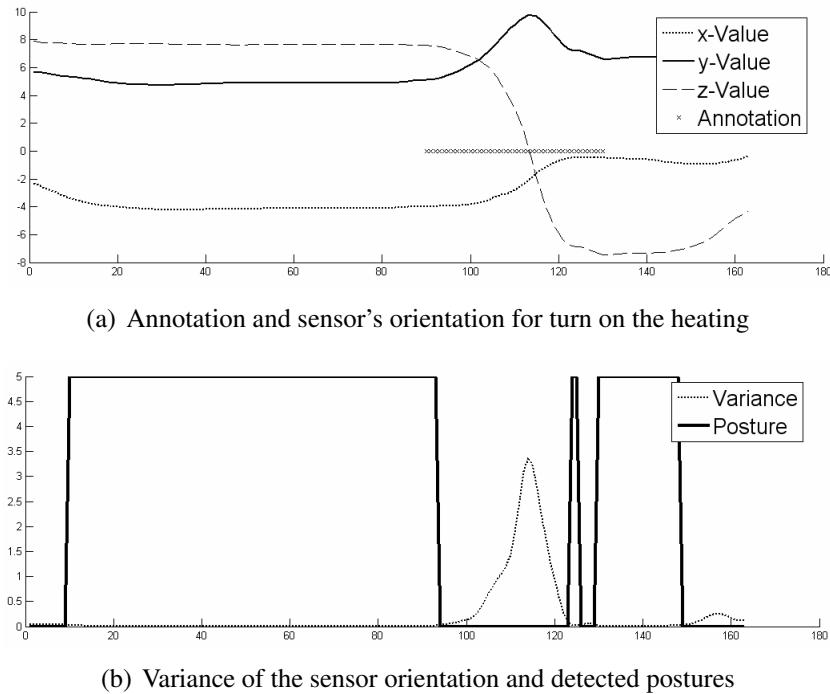


Figure 3.3: Segmentation of activity *turn on the heating* based on the variance of the current sensor orientation. The upper figure shows the annotation and the sensor's orientation. The lower figure contains the variance of the sensor orientation and the detected postures.

Grouping of the data in regions

At each point in time and for each sensor position, the gravity in the sensor's coordinate system points to a specific direction. When a user keeps his/her arm in almost the same position for a short period of time, this direction will not change. A user performing a movement, for example when pulling the hand brake, will cause a significant change of this direction. Using the rotation matrix provided by the MTx inertial measurement unit (Section 3.2.2), the gravity vector $(0 \ 0 \ 9.81) m/s^2$ is transferred into the sensor coordinate system. The upper Figure 3.3(a) illustrates the transformed components of the gravity vector in the sensor coordinate system while performing the activity *turn heating on*. In the first part, the values stay nearly constant. Between samples 100 and 120 (annotated activity illustrated as crosses), there is a high variance in the data which settles

again in the remainder of the displayed data. The variance is calculated over a sliding window of 10 samples of each curve.

Accordingly, those three values are added to obtain the magnitude of the variance. A high variance indicates a fast change of the arm's position which can be seen between samples 100 and 120 in Figure 3.3(b). A low variance acts as an indicator for a posture. The variance curve remains low at the beginning (start posture) and the end (end posture) of the annotated activity. A simple threshold for the variance separates phases of postures from those moving the arm. The lower Figure 3.3(b) contains the mentioned variance (dashed) and a visualization of the grouped regions (solid). If a region is a posture, the posture value is increased to 5 for visualization.

The experiments show that a threshold for the variance of 0.01 for opening and closing the hood and a threshold of 0.03 for all other activities results in sufficient segmentation. The threshold for opening/closing the hood has to be lower since the hood's pneumatic closing mechanism prevents the user from changing the arm's position very fast. A higher threshold would classify the slow movement between the start and end posture as a posture as well.

Segmenting the data

By means of the detected posture areas, the segments are built. Later, the activity classification will be performed on those segments. A segment starts at the end of one posture and ends at the beginning of another. Furthermore, segments for a specific activity have a minimal and maximal length shown in Table 3.1. A minimum length of 0.2 seconds and a maximum length of 0.7 for turning the heating on will lead to two segments in the illustrated example in Figure 3.3(b), one ranging from sample 94 to 123 and the other from 94 to 129. Both segments in the example in Figure 3.3(b) contain the movement of the activity *heating on*. Therefore, there is a necessity for handling overlapping segments.

Discarding overlapping segments

The example in Figure 3.3(b) illustrates that overlapping segments can contain the same activity. Before evaluating the performance of our method for activity detection in Section 3.4.2, it is helpful to keep only one of those overlapping segments. We define two segments S_1 and S_2 overlap if equation 3.1 holds true:

$$\frac{\text{length}(S_1 \cap S_2)}{\text{length}(S_1 \cup S_2)} > 0.5 \quad (3.1)$$

$S_1 \cap S_2$ represents the intersection of S_1 and S_2 . $S_1 \cup S_2$ means the union of S_1 and S_2 . This equation ensures that not only segments with a high common region overlap. Furthermore, the regions have to be of similar size.

If two segments overlap, the distance between the segments' postures and the corresponding learned postures of the specific activity in the training data is calculated. The segment that is closer to the training data will remain; the other one will be discarded.

In this section, we discussed the grouping of data into regions of movements and postures. Using those regions, data segments are built that are enclosed by two different postures. Subsequently, overlapping segments are deleted to ensure a correct evaluation in a later stage. This section provides a basis for the next set of experiments where we show that classifiers using features on the start and end posture of the segments obtain high recognition results for activity spotting in continuous data. The distinctive performance of start posture, end posture and the difference in orientation of both will be analyzed for our activities.

3.4.2 Classification using postures

The following experiments show that separate features built on top of start and end posture as well as the orientation difference between both lead to promising recognition rates. Classifiers using those features are applied on the segments as extracted in Section 3.4.1. The obtained results are better for each separate classifier compared to the classification using HMMs. We will show that a classifier combining the features obtains the highest recognition rates.

Classification using separate postures

This experiment evaluates three single classifiers using three different distance vectors:

- Posture 1: Distance of the segment's start posture to a mean start posture of the training data of each activity.
- Posture 2: Distance of the segment's end posture to a mean end posture of the training data of each activity.
- Diff Pos 1 & Pos 2: Distance of the segment's difference vector to a mean difference vector of the training data of each activity.

A segment is accepted as detection if the considered distance is lower than a threshold. To evaluate a classifier, the respective threshold for the distance is increased stepwise. Figure 3.4 illustrates the average precision-recall for the three single classifiers Posture 1, Posture 2 and Diff Pos 1 & 2 over all activities. The precision plot shows a clear difference between classifier 3 and the other two. Up to a recall of almost 0.9, classifier 3 has a precision of more than 0.6. Classifier 1 and 2 perform slightly worse. All posture classifiers outperform the results by hidden Markov models.

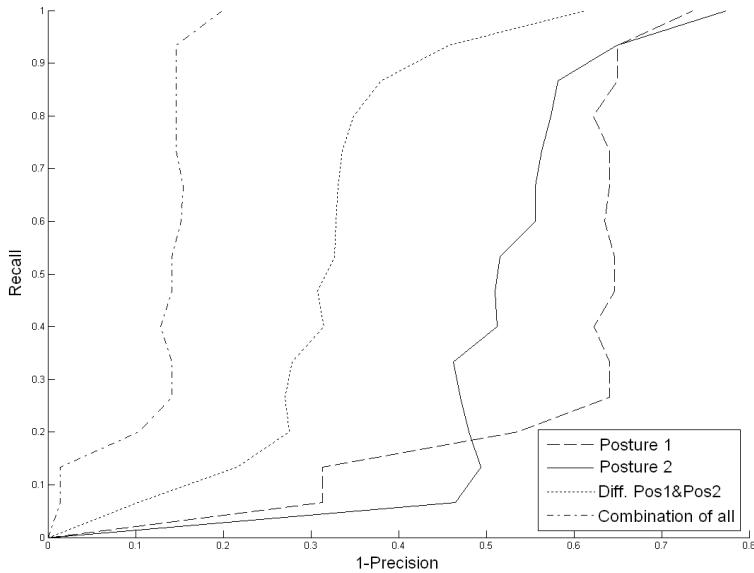


Figure 3.4: Average precision-recall results for classification with Posture 1, Posture 2, Diff Pos 1 & Pos 2 and the fusion of all settings.

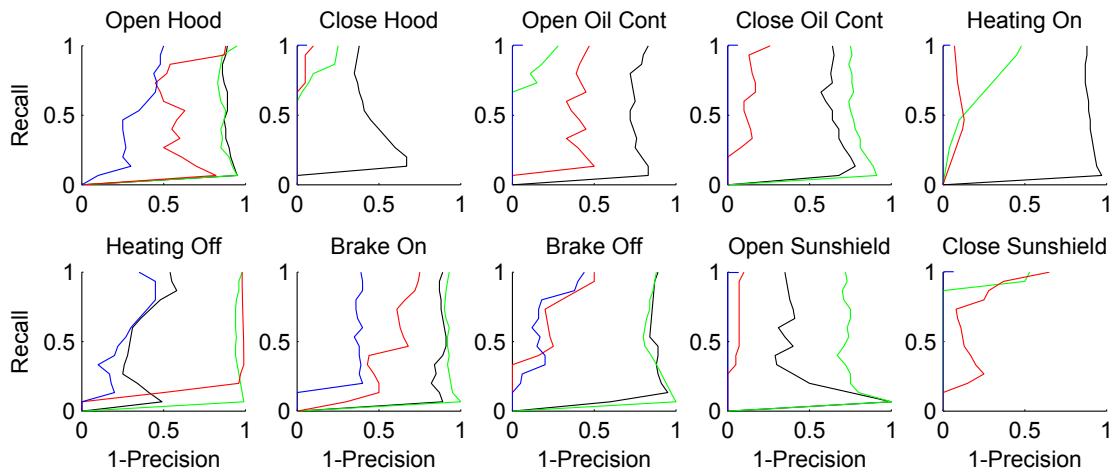


Figure 3.5: Precision-recall of all activities for classification with Posture 1 (black), Posture 2 (green), Diff Pos 1 & Pos 2 (red) and the fusion of all settings (blue).

We notice that each classifier models a different subset of activities in a sufficient way (see Figure 3.5). The strongest results for classifier 1 could be observed for the activities *open* and *close sun shield*, *heating off* and *close hood*. In contrast, the precision of *brake on/off*, *heating on* and *open hood* is low.

Classifier 2 recognizes the activities *close hood* or *sun shield* and *open oil container* with a high precision in average. Even for a quite high recall of almost 0.7, the precision is still 1. The precision drops to a minimum of 0.5 for a recall of 1.

Classifier 3 is selective for more activities: 6 out of 10 activities have a higher precision than 0.5 for a recall up to 1. For *heating on* and *open sun shield*, the precision remains higher than 0.9 for a full recall. Obviously, classifier 3 depends on the start and end posture. In the case that either the start or end posture has a high variance, the difference will not perform nearly as well. *Heating off* is an example where this effect occurs. Both classifiers 2 and 3 do not score a high precision when increasing the recall for classifier 2 and 3.

The experiments show that postures and the difference between postures qualify as features for a good classifier. Almost all activities in our set are modeled well by at least one classifier. We will define a generalized classifier that will fuse the three features and obtain better recognition results than classifiers using single features.

Classification with combined postures

The previous experiments demonstrated features on single postures and the difference of two postures as a strong feature to reduce the number of *false positives* in a continuous data stream. All three postures individually perform on average much better than the approach using HMMs on the segments or on a sliding window. Also, one can observe that depending on the activity, different postures can classify better than others. A combination of the three features by summing up the distances can improve the recognition significantly.

On average and for each activity, the combined classifier performs better than the three classifiers on single features. Figure 3.4 includes the average precision-recall curve of the combined classifier. The classifier obtains an average precision of 0.8 for a recall up to 1. Only for *open hood*, *turn heating off*, *pulling* and *release handbrake*, we do not find an optimal value (see Figure 3.5). *Open hood* performs worst. The precision is still higher than 0.5 for a recall of 0.9.

3.5 Conclusion

This chapter reports preliminary results of analyzing different methods for activity spotting in continuous data. The first set of experiments demonstrates that left-right HMMs trained on segmented training data model the segmented activity test sequences sufficiently. For the continuous case, we note that hidden Markov models cannot easily be transferred from a segmented to a continuous recognition. The experiment's results show that the models cannot discriminate the short activities from irrelevant data. More specifically, the number of false positives is too high. Section 6.5 will report on similar results applying HMMs on a different data set including an evaluation of different HMM parameters for modeling the number of states, transitions, and the type of HMM.

The proposed approach using postures to segment and classify the data obtains good results. Postures are suitable both for segmenting the data stream in smaller segments and

to reduce the number of analyzed segments. We provide an evaluation of three different features and a fusion of them. In particular, the postures already classify the regarded activities to a high degree. A classifier fusing the distance features of the start and end posture, as well as the distance between both, performs best on average.

Note that the work in this chapter is a preliminary study evaluating the segmentation method as well as the distinctiveness of start and end postures. It is a first step to overcome the challenge *Continuous data stream with large background*. While this approach is already able to sufficiently deal with little training data and simple data sets, the recognition is still user dependent and does not model the movement between the postures. In addition, many parameters and thresholds are set manually. Nevertheless, the findings in this work provide a basis for the remaining chapters of this thesis. In Chapter 4 we propose a method for user-independent gesture recognition in continuous data streams. The segmentation as introduced in the current chapter is extended. Furthermore, we also model the movement part between the postures and thereby allow for a recognition of more complex activities.

4

User-Independent Gesture Recognition in Continuous Data Streams

As computing devices become smaller, lighter and more powerful, more and more people are using small and wearable devices in mobile settings every day. However, interacting with these devices is often unsatisfying and tedious. In this chapter, we present a new approach to enable gesture recognition in continuous data streams. In addition, we show that the complexity of gestures affects their distinctiveness from gestures in daily life. Beyond the recognition aspects, we pay particular attention to the social acceptability of the evaluated gestures.

4.1 Introduction

Interacting with computing devices is particularly difficult while physically moving, i.e. walking, jogging, biking or driving. Imagine you are for example jogging or walking in a city and your cell phone starts ringing. Typically there are only a few actions that you would like to perform in this situation like picking up the phone, suspending the call, or ignoring it. Or, while you are cycling to work or casually walking around in the city you may want to skip to the next song without taking out your MP3 player. Also, during a presentation one could easily move forward and backward in the presented slides or start and stop a video device with a small set of commands. What makes interaction with current technology unsatisfying in these situations is the fact that users have to stop their primary task. Either users have to stop jogging or cycling in order to take out the devices from a belt bag, or a presenter has to physically walk to the computer or video recorder.

The starting point of this chapter is the observation that in many truly mobile scenarios a small set of commands for functionalities such as *Start*, *Stop*, *Pause*, *Forward* or *Backward* would already be very useful and may even fully satisfy the users' needs in these situations. In many situations, for example sitting in public areas or while jogging, speech recognition is not adequate because of noisy environments or because people may feel annoyed. We argue that the users' arm gestures are an interesting alternative to enable the user to explicitly issue a limited set of commands and therefore enable interaction

in truly mobile settings. We hypothesize that for many scenarios, less than ten gestures would meet the users' expectations. Ideally, those gestures should be almost similar for different scenarios reducing the users' training effort. In this thesis we refer to gestures for controlling a user interface as *control gestures*.

Control gestures have to fulfill two basic requirements. First, they should be distinguishable from gestures that occur in daily life. Since hand movements and gestures are a natural way of human interaction, the system must distinguish control gestures from common ones. In this chapter we evaluate the distinctiveness of seven control gestures with different complexity against daily gestures in the settings ranging from teaching (both university and school) to more complex settings such as jogging, cycling and driving a car. A second requirement is that the gestures have to be socially acceptable. Socially acceptable means that the gestures have to be unobtrusive both for the user performing them and for his/her environment.

The main contribution of the chapter is a new approach to enable gesture recognition in continuous data streams. We use turning points in users' arm movements to identify segments of interest in the continuous data stream. The recognition algorithm considers both the arm movements between turning points and the shape of the turning points for classification. Using the new method, seven gestures of different complexity are evaluated against a realistic background class of daily gestures in five different scenarios.

The rest of the chapter is organized as follows. Section 4.2 summarizes user interviews and the experimental setup including the used data sets and sensors. In Section 4.3, the features as well as the proposed approach for gesture recognition in continuous data streams is introduced. Section 4.4 evaluates how discriminant seven sample gestures are with respect to the collected background data in different scenarios. Section 4.5 concludes and summarizes the main contributions of the chapter.

4.2 Experimental Setup

Control gestures should be both socially acceptable and distinguishable. Section 4.2.1 *Collecting background data* argues for recording a long set of background data in different scenarios to analyze how distinctive control gestures are from gestures in daily life. Furthermore, we describe how we designed the data recordings to be as natural as possible to obtain realistic data. Section 4.2.2 *Defining and collecting the navigation gestures* summarizes the outcomes of user interviews that we performed to find out criteria of social acceptance. Furthermore, a set of seven gestures of various complexity is introduced based on the interview results.

4.2.1 Collecting background data

As previously mentioned, control gestures have to be distinctive from daily gestures. For an evaluation how control gestures differ from daily performed gestures, we need

to record background data covering the wide variety of gestures of daily life as much as possible. Already a small set of control gestures such as Start, Stop, Pause, Forward or Backward could facilitate human computer interaction with a cell phone or MP3 player in truly mobile scenarios like jogging or cycling. Ideally, similar gestures could be used both by lecturers (at school or university) to switch presentation slides or control video or audio devices and by people driving a car. Currently, drivers physically press buttons to control devices causing them to lose their attention to the traffic for a short time. Control gestures make it possible to constantly keep the eyes on the traffic since users do not have to focus on their wrist or hand while performing gestures.

Based on the observation that a restricted set of commands for functionalities would already be very useful and may even fully satisfy the users' needs in these situations, we focus on five scenarios with a large variability in daily gestures. The scenarios are *Professors at university*, *Teachers at elementary schools*, *People while jogging*, *People while cycling* and *People while driving*.

These scenarios are also some of the most interesting application scenarios for the recognition of control gestures. In 4.4.1, the control gestures will be evaluated against the whole background class. In the optimal case, control gestures will be reliably recognized against each other without causing false alarms in the background data at all. The more daily gestures are recorded, the better is the analysis of gestures' distinctiveness.

For the five scenarios we recorded 2250 minutes (or 37.5h) of data from 30 persons. 17 male and 13 female volunteers range between 22 to 60 in age. Per scenario, we recorded data of more than 5 persons for about 90 minutes in the analyzed scenarios. For data collection, the users are equipped with a small wristband sensor board (Section 3.2.2 *Hardware - MTx inertial measurement unit*) at their right wrist. A comfortable wristband to fix the sensor at the users' wrist and a wireless communication of the sensor with the recording computer allows the user to gesticulate and act naturally which is important to obtain a realistic recording of daily gestures.

4.2.2 Defining and collecting the navigation gestures

The second requirement for control gestures is social acceptability. Socially accepted in this context means that the gestures have to be unobtrusive both for the user performing them and for his/her environment. Beyond the background data of daily gestures described in the previous section, we need a set of unobtrusive control gestures performed by several people. This section summarizes how we choose the set based on user interviews. Subsequently, we describe how the training and test set of control gestures was recorded for later evaluation.

User Interviews We interviewed 80 persons in five different scenarios to find adequate navigation gestures for our scenarios. Performing the interviews, we wanted to get an overview what users consider to be socially acceptable as well as an appropriate control

gesture. Therefore, we asked the interviewees to perform different gestures of their choice while being video recorded. After performing the gestures, the users gave a first rating. Afterward they could adapt or revise their statement about social acceptability of different gestures while watching their video. According to their answers, gestures should be executed fast and in a natural arm position above the waistline. 70% of the users prefer vertical and horizontal movements since they are easy to perform and to remember. 90% of the interviewees rate complex gestures as not suitable for control gestures. Nevertheless, they do not see a problem in the complexity of gestures that are defined based on specific well known shapes, for example letters, figures or characters. The users believe that gestures based on shapes will considerably reduce training and concentration effort. Based on these interviews the following set of sample gestures (see Figure 4.1) was chosen for the evaluation of our method in this chapter. In the following discussion we will refer to the different gestures as *E*, *S*, *B*, *5*, *3*, *Square* and *Flank*. We could have chosen a different and a larger set of gestures based on the interviews. An important insight from the experimental results in Section 4.4 however is that the gestures should have a certain complexity to enable reliable recognition in continuous data streams.

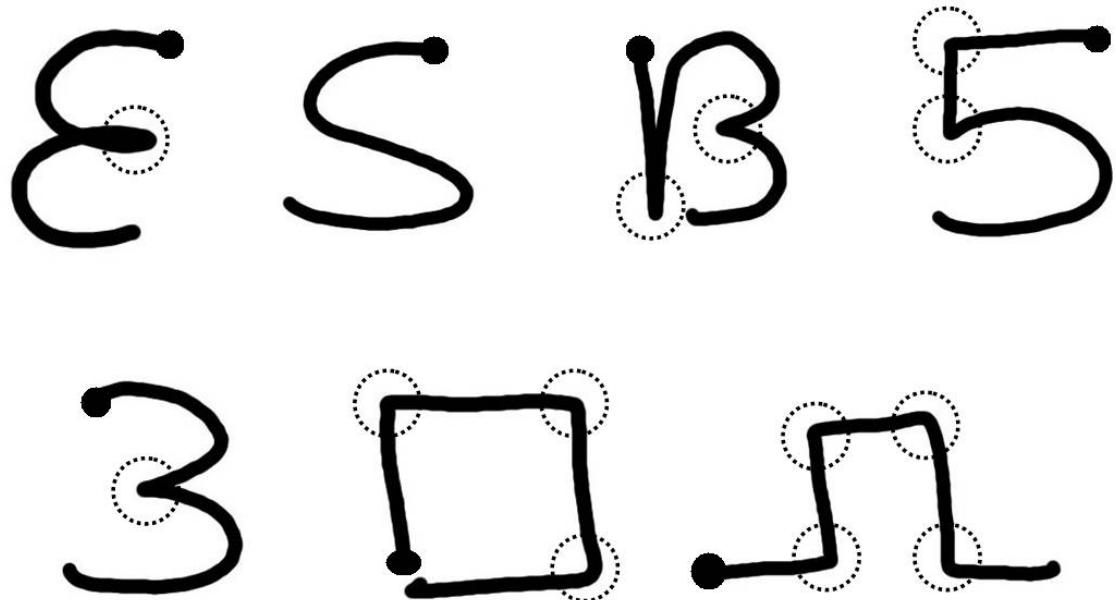


Figure 4.1: Set of seven gestures of different complexity that are evaluated against a realistic background class of daily gestures in five different scenarios.

The set contains three letters, two numbers and two geometric shapes of different complexity. Whereas gesture *S* is easy to perform, the *E* and *3* are already more complex. Between the two arcs of the latter gestures the users have to turn the direction of the arm movement (illustrated by a dashed circle). From now on we call those points *turning points*. Areas between turning points will be called *gesture primitives*. For gestures *B* and *5*, two turning points are illustrated in Figure 4.1. The most complex gestures are *Square* and *Flank* with three or four turning points. Turning points occur either at sharp

edges (for example up and down while performing gesture B) or when bending narrowly (for example between the two arcs of gesture B). Later, we will aim to identify these turning points in the data and use them to segment the continuous data stream. We will also analyze how the number of turning points affects the recognition of the respective gesture.

Collecting Control Gestures A second data set includes instances of the seven displayed gestures (see Figure 4.1) of five different people. To keep annotation effort manageable we aimed to record many gestures in a short time. At the same time we wanted to obtain a continuous recording with naturally performed gestures for which we proceeded as follows. After briefly practicing the gestures in advance, the users performed the seven gestures in random order multiple times. Between the individual gestures the users could do whatever they wanted like talking to other people, walking around, etc. For all persons, the data set contains at least 10 instances of each gesture. While performing the gestures, the users are recorded on video for later annotation. Both the gestures and the respective turning points are annotated using the video recordings.

4.3 Gesture Segmentation and Classification in Continuous Data Streams

This section describes our novel approach for gesture recognition. First (Section 4.3.1), we introduce a new segmentation procedure based on the detecting of potential turning points of arm movements in continuous data streams. As we will see in the experiments, this novel segmentation procedure results in a significant data reduction even though the number of turning points and segments between those turning points is still large. Importantly however, the segmentation procedure is capable to return all turning points of our gestures as defined in Section 4.2.2. The second step of the algorithm calculates two types of features for gesture classification (Section 4.3.2). The first feature is a shape histogram of the turning points themselves and the second feature is a direction histogram on the segments between two turning points. Section 4.3.3 then describes that the segmentation and the feature extraction enable the definition of a simple yet effective algorithm for gesture recognition. Section 4.4 experimentally analyzes the performance of this novel algorithm.

4.3.1 Segmentation of the continuous data stream

Segmentation of the continuous data stream may be seen as a filter for subsequent classification. Ideally the result of this filtering step are segments that are both easier to classify than the continuous data stream and that coincide with the turning points of the control gestures. We describe how turning points of arm movements can be used for a reliable

segmentation of a continuous data stream. As previously mentioned, two examples for such a turning point can be observed while performing a gesture B as defined in Figure 4.1. Turning the direction between moving the arm down and up leads to a first turning point. Bending the movement narrowly between the two arcs of gesture B defines the second turning point. We will illustrate with gesture B as an example how the method works. Similarly, turning points can be identified for all introduced gestures.

While performing arm gestures the movements can be described by a sequence of arm orientations. These 3D-orientations can be extracted from the XSens (Section 3.2.2 *Hardware - MTx inertial measurement unit*) if the sensor is placed on the user's wrist. It is robust against different arm twists while performing the gesture. Figure 4.2 displays a sample sequence of 3D-arm orientations on the upper right. Similar to Section 3.4.1, we calculate the variance over the arm's orientation. The lower picture on the right illustrates the variance of this orientation. Clearly, the variance of the orientation (over a small time interval) will be lower at turning points since the user has to slow down the speed of movement while turning. While performing a gesture the variance will increase for segments between two turning points. Therefore, we detect local minima within the variance to split the continuous data stream into segments which are illustrated as colored and numbered areas at the base line of the plot. Note that simple thresholding as applied in Section 3.4.1 is often not enough to identify all turning points because of high variance. The detection of local minima is sufficient (i.e. for our gestures introduced before). We find that there is a coherence between the detected minima and the turning points.

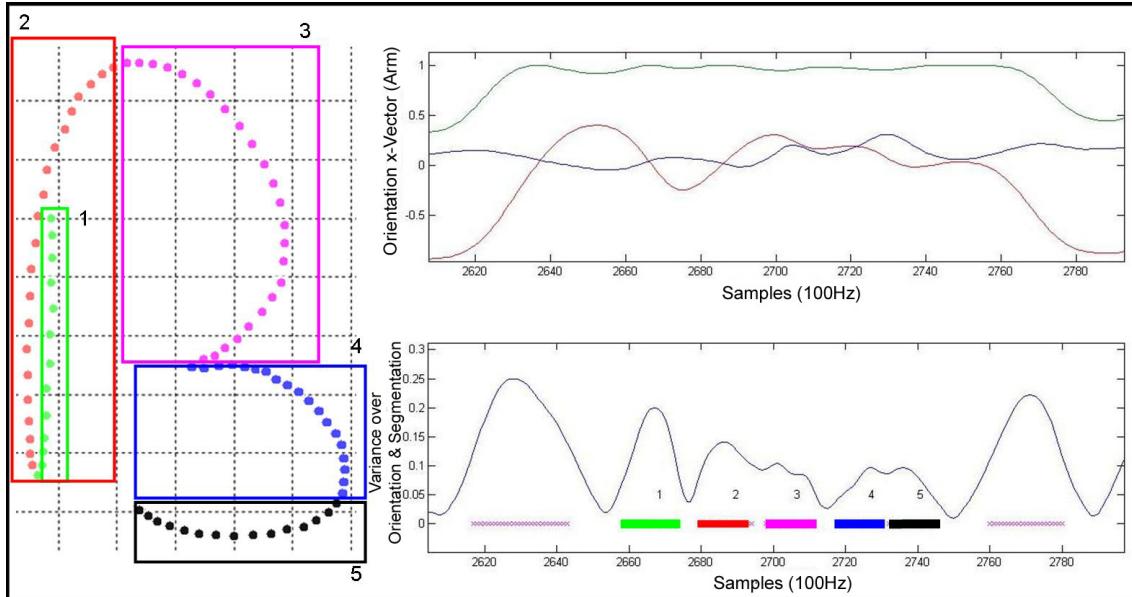


Figure 4.2: Sequence of arm orientations for a gesture B projected (for illustration purposes only) onto a plane (left), Orientation of the arm, variance over the orientation and segmentation for the same gesture B (right).

Figure 4.2 shows an example gesture (B) and the corresponding segmentation achieved with the above algorithm. Plotting the values of the arm's orientation results in data points

on a unit-sphere. Rather than showing this sphere Figure 4.2 (left) shows the projection of the orientations onto a single plane (not used for classification). In the same figure on the right, we can observe that there are several local minima in the variance of the arm's orientation. These local minima are used to extract a sequence of segments of which segments one to five correspond to different parts of the gesture B shown on the left. The segments and the corresponding parts of the gesture are numbered and color-coded. In this example there is a strong correlation between the local minima and the turning points. Particularly, the first turning point of gestures B is between segment one and two and the second turning point is enclosed by segments three and four.

To quantify the quality of the segmentation algorithm we evaluated the performance on the entire gesture set as well as on the background set. Overall the gesture set contains 425 instances of different gestures. Depending on our definition of turning points, 1228 points should be extracted from this set. The proposed algorithm detects all 1228 turning points but also detects a few more. Overall 1770 local minima are detected. The second question is how many turning points are detected in the background set. Within the 37.5h (2250 min) of background data, the algorithm detects 17,166,666 minima for later evaluation. These numbers show that the algorithm is capable to find all relevant turning points and segments. At the same time however the algorithm has to extract discriminant features from the data to enable reliable gesture recognition. We will introduce two types of features for this purpose.

4.3.2 Calculation of features

The previous section introduced a procedure to segment a continuous data stream into a sequence of turning points and segments between turning points. In the following discussion, we introduce two types of features for gesture recognition. The first feature is calculated on segments between two turning points (*Direction histograms on segments between turning points*) and the second feature is a shape histogram of the turning points themselves (*Shape histograms on turning points*).

To enable the calculation of these features, we introduce the following preprocessing step. As mentioned before, the values of the arm orientations result in data points on a unit-sphere. In order to be independent of the absolute arm orientation in which the gesture is performed, we want to subtract the mean orientation. For this we calculate the mean orientation $\vec{\mu}$ of the arm's movements over an appropriate time interval. For an entire segment the corresponding time interval is given by the two enclosing turning points. A time interval for turning points is a local neighborhood around the turning point which will be defined later. Then we project the data points of that time interval onto the plane defined by vector $\vec{\mu}$ as the normal. We define the first basis vector \vec{b}_1 as the cross product of $\vec{\mu}$ and the gravity \vec{g} . \vec{b}_2 is the cross product of \vec{b}_1 and $\vec{\mu}$. These projected data points are used to calculate the two different feature types discussed below.

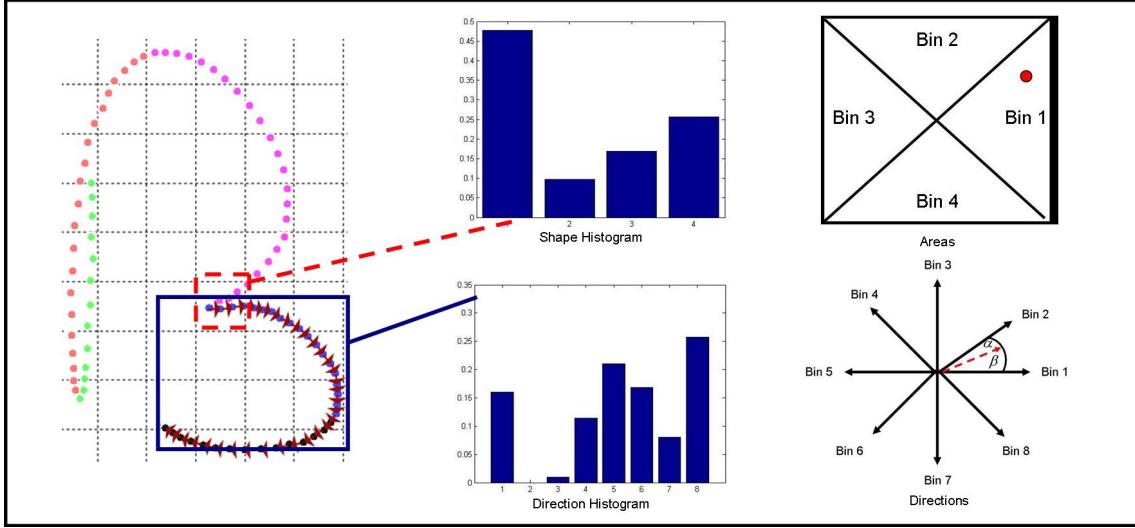


Figure 4.3: Gesture B in 2D (left), Shape Histogram of 2nd Turning Point and Direction Histogram of the third part (middle), Area and Direction Definition for the Histograms (right).

Direction histograms of segments between turning points For the segments (enclosed by two turning points) we propose to calculate direction histograms as feature. For illustration, we will again use the previous example of a gesture B. First we calculate the directional vectors of succeeding data points between two turning points. The directional vectors from the second turning point to the end of the gesture (see blue solid box in the left picture of Figure 4.3) are displayed as arrows. We define eight directions in the two dimensional plane as illustrated in the lower right picture. The directional vectors of the segment are then assigned to eight bins of a histogram represented by eight directions. This assignment leads to the direction histogram illustrated in the middle lower picture of Figure 4.3. We can observe high values for *bin five* and *bin eight* and minimum values for *bin two* and *bin three*. Visually, this distribution is a good representation for the arc shape surrounded by the blue solid box in the left picture of Figure 4.3.

Note that the gesture primitive as illustrated in the blue solid box encloses two segments as illustrated in Figure 4.2. While primitives are not always divided into two (or sometimes even more) segments such a deviation has to be expected. Therefore, we will have to test more segments and combinations of segments in the subsequent gesture classification (Section 4.3.3).

To make the feature robust to speed-difference, the histogram is normalized by the overall length of the considered segments between two enclosing turning points. Also, we use soft assignment to the two neighboring bins where the assignment is proportional to the angle of the directional vectors. An example is illustrated in Figure 4.3 by the red dashed arrow in the lower right figure. Here the direction lies in between the directions of *bin 1* and *bin 2*. Therefore, it is assigned with percentage $\frac{\alpha}{45}$ to *bin 1* and with percentage of $\frac{\beta}{45}$ to *bin 2*.

Shape histograms of turning points We represent the local shape of the turning points using shape histograms. In the left picture of Figure 4.3 a local neighborhood of the second turning point is surrounded by a red dashed box. This time, we split the area into four spatial bins defined by the axes that can be seen in the upper right picture of Figure 4.3. We consider 30% of the data between the current to the previous and next turning point for the local neighborhood of a turning point. Depending on the position of the data sample, these data points are assigned to a four bin histogram. Analogous to the previous histogram, we use soft assignment for the four bins. The middle upper picture of Figure 4.3 illustrates the resulting shape histogram for the second turning point of the sample gesture *B*. The resulting histogram has a high peak in *bin 1*.

Note the difference between the shape histogram and the direction histogram. While the direction histogram for segments codes the relative directions across a segment, the shape histograms for turning points describe the local shape around the turning points. Therefore, different information is coded by these features and as we will see in the experiments the two features complement each other well. In the experiments the combination of both features is clearly superior to the individual features.

4.3.3 Gesture classification

The last step of our algorithm consists of using the segmentation as well as the direction and shape histograms for gesture classification. Based on the features defined in the previous sections, we build a classifier to evaluate the specified gestures.

We already mentioned that control gestures in the training data and the corresponding turning points are hand annotated. Therefore, we can directly calculate the direction histograms on the given gesture primitives as well as the shape histograms on the turning points. Calculation of the histograms for the test data is more complex. As previously analyzed in Section 4.3.1, we can automatically detect all turning points in the continuous data stream. However, the number of turning points that we detect is larger than the number expected from the definition in Section 4.2.2. We have seen that a gesture primitive as defined in Section 4.2.2 can be composed of several segments. Therefore, the following method distinguishes between gesture primitives and segments. From the training data we extract the maximal number of segments that correspond to a single primitive. This maximal number of segments per primitive is taken as the maximum during testing. For straight primitives, for example primitives of the gestures *Square* or *Flank*, the maximum number is 2. More complex primitives including arcs have a maximum number of 3 segments.

When calculating the histograms on segments and turning points in the test data, all possible combinations of segments with an equal or smaller number than the maximum based on the training data have to be analyzed. For all combinations, the direction histograms as well as the shape histograms can be calculated. Next we apply histogram intersection to calculate the distance between shape and direction histograms on the test

data to the corresponding histograms on the training data considering the nearest neighbor. Histogram intersection [M.J.Swain and D.H.Ballard 1991] adds up the minimum values between each pair of corresponding bins as defined in $I(H_a, H_b) = \sum_{i=1}^n \min(H_a[i], H_b[i])$. Two similar histograms will result in a large intersection value. Subsequently, we approximate a probability for a match of two histograms by a flat sigmoid function calculated from the intersection values of the training data.

In the experiments the overall probability for a gesture given segments in the test data is calculated in three ways. In the first setting the probabilities are calculated considering the direction histograms only whereas in the second setting only the shape histograms are considered. In the last setting both feature types are used to calculate the probabilities. Depending on the number of primitives, in all three settings multiple probabilities are obtained. These are combined in a naive Bayes fashion (see Section 2.3) by multiplication where we assume independence of the features. While the features are clearly not independent the naive Bayes classifier obtained highly promising results.

4.4 Experiments and Results

This section reports on two types of experiments. In Section 4.4.1 *Evaluation of control gestures against background*, we analyze how distinctive each control gesture is with respect to gestures of daily life (i.e. the background). In Section 4.4.2 *Confusion between control gestures*, we summarize the confusion between the control gestures among themselves.

We perform leave-one-user-out cross-validation. As we have data from five different users for our control gesture set we effectively perform five-fold cross validation across the five users.

4.4.1 Evaluation of control gestures against background

In this section we evaluate the distinctiveness of each control gesture separately with respect to the collected background data. Therefore, in every cross-validation round we add the background data of daily gestures to the test set of control gestures. First, the probability for each control gesture in the test set is calculated. Next, we calculate the probabilities of all possible combinations of segments in the background data for the respective evaluated gestures. Below we report how the number of false positives grows while increasing the recall for the actual control gestures in the current test set. In the following the three settings are evaluated and summarized. Ideally both precision and recall are 100%. However, in the envisioned scenarios already a small number of false alarms might be annoying so that we are particularly interested in the percentage (=recall) of control gestures that can be recognized without causing any false positives on our 37.5h long background data set.

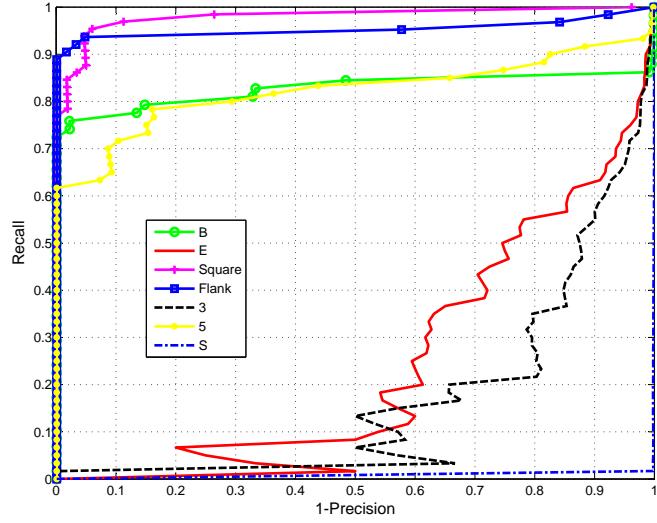


Figure 4.4: Precision-recall results for the seven gestures against the background when using direction histograms only.

Figure 4.4 shows the recognition results when using direction histograms only. Clearly the gesture *Flank* performs best with a recall of almost 0.9 without any false alarms in the background data. Gesture *Square* follows with a recall around 0.75 without any false alarms while gestures *B* and *5* still obtain a recall of more than 0.6. The remaining gestures *E*, *3* and *S* cause too many false detections using direction histograms only.

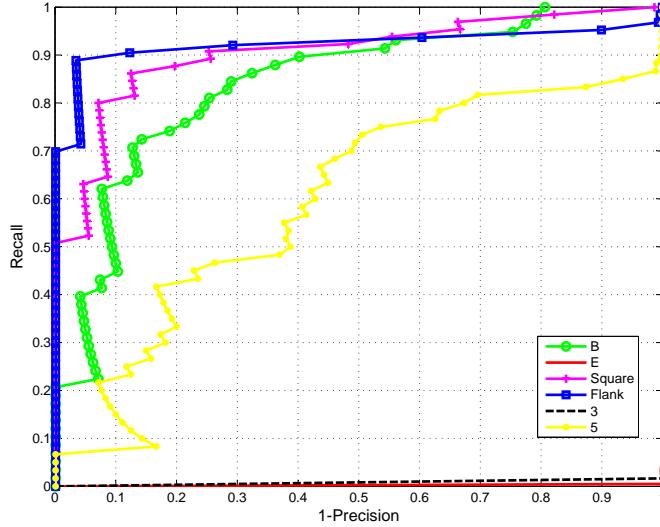


Figure 4.5: Precision-Recall results for the seven gestures against the background when using shape histograms only.

Figure 4.5 illustrates the results of the second feature set, namely shape histograms. It needs to be mentioned that the gesture *S* cannot be recognized using this feature since it does not contain a turning point. Similar to the first results, the gestures *Flank* and

Square show best results. All other control gestures involve too many false positives in the background data.

As can be seen in Figure 4.6, four gestures obtain a high recall of more than 0.8 without returning any false positive in the background data for the combined classifier. Due to the obtained results, the four gestures *B*, *Square*, *Flank* and *5* can be considered to be suitable and distinguishable gestures. In fact, the gestures *3* and *E* perform better for the combined classifier, but we consider the performance not high enough. As expected the combined classifier performs best.

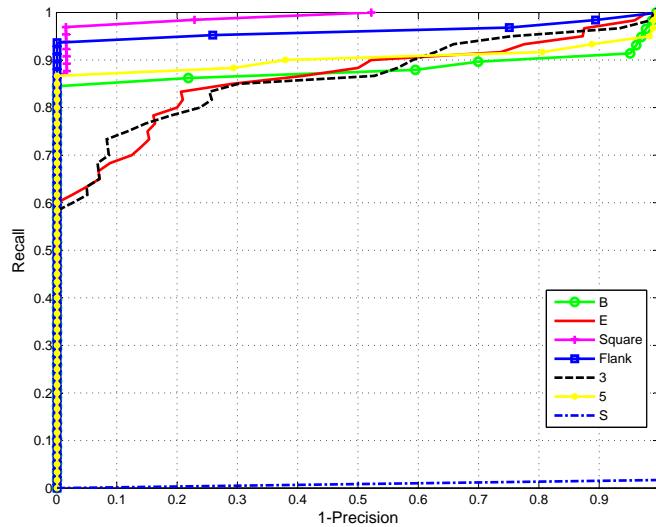


Figure 4.6: Precision-recall results for the seven gestures against the background when combining direction and shape histograms.

It is interesting to note that the obtained ordering of gestures is highly correlated to the number of primitives and turning points per gesture. While *Flank* and *Square* have 4 or more primitives, the next best performing gestures (*B* and *5*) have 3 and the remaining have only 2 or 1 primitive.

Table 4.1 illustrates the dependencies between false positives and the number of primitives in a gesture. Gesture *S* performs the worst. Already for a low recall of 0.3, more than 10,000 false positives are detected. While evaluating the gestures *E* and *3* for a recall of 0.8, more than ten false positives are detected. Broadly speaking, a recall lower than 0.8 is likely to be unacceptable for users. Using this criteria, none of the gestures in our set with two or less primitives is suitable for gesture recognition. Good results are obtained for the remaining gestures. These observations confirm that sample gestures composed of four or more primitives are most suited and can obtain a promising recall of 0.9 and more with nearly no false positives. Analyzing the number of false positives for subparts of gesture *B* more closely shows the strong inverse correlation between number of primitives and false positives. Just considering one primitive for classification, more than 4,000 false positives are obtained at a low recall of 0.3. The number of false positives

False Positives in Background	Recall				
	0.3	0.5	0.8	0.9	1
Flank	0	0	0	0	140113
Square	0	0	0	1	71
5	0	0	0	33	13324
E	0	0	12	59	2925
3	0	0	15	78	6694
S	10329	25095	112050	165472	196347
B(3 Segm.)	0	0	0	1038	11502
B(2 Segm.)	0	5	309	49019	124250
B(1 Segm.)	4552	8095	29497	318100	572053

Table 4.1: False Positives for the seven gestures of different complexity in dependence on recall. An analysis of the number of false positives for subparts of gesture B shows the strong inverse correlation between number of primitives and false positives.

drops when including features of the second primitive and reaches best performance when combining all three primitives. The relatively large number of false positives for gesture *Flank* at recall of 1 results from the bad performances of one single user. Figure 4.7 illustrates the worst instances that are all taken from a single user. In order to accept those instances lots of other daily gestures had to be accepted before those could be recognized. Clearly the user did not perform the gesture with lots of caution and the user would have to train to perform this gesture more carefully.

4.4.2 Confusion between control gestures

We have shown that four of the proposed seven gestures are suitable for gesture recognition in continuous data. We now evaluate the confusion between the remaining four control gestures. Again, we calculate the probability of all possible combinations of segments per primitive depending on the analyzed gesture, this time in the continuous gesture stream. We evaluate all gestures against the obtained combinations. Once a gesture instance in the test set is not rejected and classified as a wrong gesture, we observe a confusion.

Table 4.2 shows that no confusion are obtained for the best four gestures. Only three instances of gesture *Flank* are rejected because of bad execution as illustrated in Figure 4.7.

4.5 Conclusions

In this chapter, we present a new approach to enable gesture recognition in continuous data streams. Turning points in users' arm movements are used in order to identify segments of interest in the continuous data stream. The recognition algorithm considers both

		Classified Activity					Sum	Recall
Ground Truth	B	58	0	0	0	0		
	Square	0	65	0	0	0	65	100,0%
	Flank	0	0	60	0	3	63	95,2%
	5	0	0	0	60	0	60	100,0%
	Sum	58	65	60	60	3	246	
	Precision	100,0%	100,0%	100,0%	100,0%			

Table 4.2: Confusion Matrix of the gestures B, 5, Square and Flank.

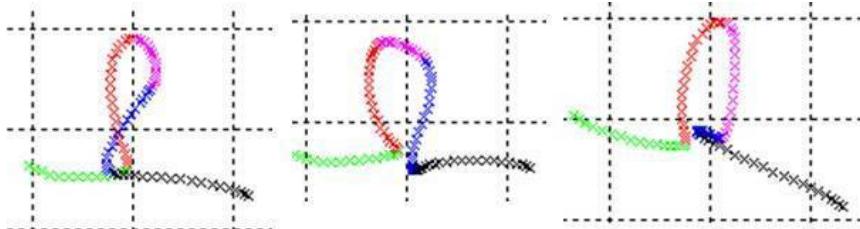


Figure 4.7: Worst instances of gesture Flank that are all taken from a single user.

the arm movements between turning points and the shape of the turning points for classification. Using the new method, seven gestures of different complexity are evaluated against a realistic background class of daily gestures in five different scenarios. The novel segmentation extracts all turning points as defined in Section 4.2.2. Since the proposed method results in a significant data reduction, classification becomes feasible. Direction histograms on gesture primitives and shape histograms on turning points have proven to be suitable features for gesture recognition. We have shown that the complexity of gestures affects their distinctiveness from gestures in daily life. Gestures composed of three or more primitives can often be considered as suitable control gestures. Beyond the recognition aspects, we pay particular attention to the social acceptability of the evaluated gestures. We performed user interviews in order to find adequate control gestures for the five scenarios. Although highly complex gestures will not be accepted by users, the users believe that gestures based on shapes, letters and other known forms will reduce the training and concentration effort and therefore be accepted as control gestures.

This chapter is an important step in addressing the challenge *High variance in performance and user independence*. Nevertheless, the number of considered gestures is still small. Furthermore, the recognition of explicit gestures is obviously less complex than the recognition of activities because explicit gestures do not vary as much in performance. The following chapter transfers the current results to activity recognition. A novel and robust model-based approach using body-model derived primitives addresses the challenges *Multiple types and diversity of activities* and *High variance in performance and user independence*.

5

Multi Activity Recognition based on Body-Model-Derived Primitives

This chapter presents a model-based approach to activity recognition using body-model derived primitives. Joint boosting enables the automatic discovery of important and distinctive features ranging from motion over posture to location. In experiments we show the feasibility of the approach presenting user-dependent and across user results for a previously published data set. The specific scenario that we study is composed of 20 activities in quality inspection of a car production process.

5.1 Introduction

Most previous approaches to activity recognition [Deng and Tsui 2000, Ward *et al.* 2006, Ogris *et al.* 2007, Stiefmeier *et al.* 2006, Kallio *et al.* 2006, Mäntylä *et al.* 2000, Pylvänäinen 2005] rely on the use of *signal-oriented features* (such as mean, variance, and FFT-coefficients) and state-of-the-art machine learning techniques (such as HMM and SVM). However, the success of activity recognition in general is still far from being satisfactory in realistic and challenging real-world scenarios. Even state-of-the-art approaches are typically challenged by the recognition of short and non-repetitive activities, by the recognition of a large number of activities in a user-independent manner, and by spotting activities in large and continuous data streams. While previous work has aimed to address some of these challenges at least individually [Lester *et al.* 2005, Ogris *et al.* 2008, Stiefmeier *et al.* 2007], we argue that the sole use of *signal-oriented features* limits the applicability of most state-of-the-art approaches to realistic scenarios.

The first main contribution of this chapter is that we follow a model-based approach where high-level primitives are derived from a human body-model. This is in contrast to most previous work that typically relies on signal-oriented features only. As human activities are composed of various sub-actions we derive various motion primitives such as *move the hands up*, *turn the torso* or *turn the arm*. Since these primitives are based on a human body-model they are more robust or can be even invariant to the variability of performing various activities. Besides motion primitives, we also calculate posture features

and use location information. The second main contribution of the chapter is the simultaneous recognition of multiple activities. For this we employ the joint boosting framework [Torralba *et al.* 2007] as summarized in Section 2.3.2. This framework allows to select discriminant features from a large pool of features that are shared across different activity classes. This makes the classification scheme not only more efficient but also allows to recognize multiple activities in a user-independent manner. The third contribution of the chapter is a further extension of the segmentation procedures introduced in Chapter 3 and 4. The primary goal is the segmentation of activities in a continuous data stream thereby reducing the search space considerably. We will show in the experiments that the segmentation procedure enables efficient as well as effective recognition of multiple activities. Finally, we apply the proposed method to a car-quality control data set provided by [Ogris *et al.* 2008].

The chapter is structured as follows. In Section 5.2 we describe the segmentation procedure. Section 5.3 describes the various motion primitives and posture features derived from a human body-model. Section 5.4 explains how joint boosting can be used for multi activity recognition. Section 5.5 summarizes the experimental results. Finally Section 5.6 concludes and discusses the contributions of this chapter.

5.2 Activity Segmentation in Continuous Data Streams

The following sections describe our novel approach for activity recognition. We first introduce our human body-model based on inertial sensors (Section 3.2.2 *Hardware - MTx inertial measurement unit*) placed at several positions of the subject's body (Section 5.2.1). Based on this body-model, we derive our novel segmentation procedure generalizing the work presented in Section 3.4 and Section 4.3.1. As we will see in the experiments, this segmentation procedure results in a significant data reduction. The next step of the algorithm calculates multiple features and primitives ranging from motion over posture to location (Section 5.3). Section 5.4 then describes how we apply joint boosting (Section 2.3.2) to reliably detect 20 different activities. Section 5.5 experimentally analyzes the performance of this novel algorithm on a previously published data set [Ogris *et al.* 2008].

5.2.1 Body-model

Human activity recognition is directly linked to human motion and movement analysis. We are interested in the recognition of activities such as *open trunk* or *check hood gaps*. It is important to note that the exact execution of these activities may vary greatly between subjects and will often vary substantially even for the same individual due to personal preferences, fatigue and other reasons. As a result, signal-oriented features such as FFT-coefficients of body-worn sensors will also vary substantially. Any subsequent classification using such signal-oriented features only will be difficult and might be even

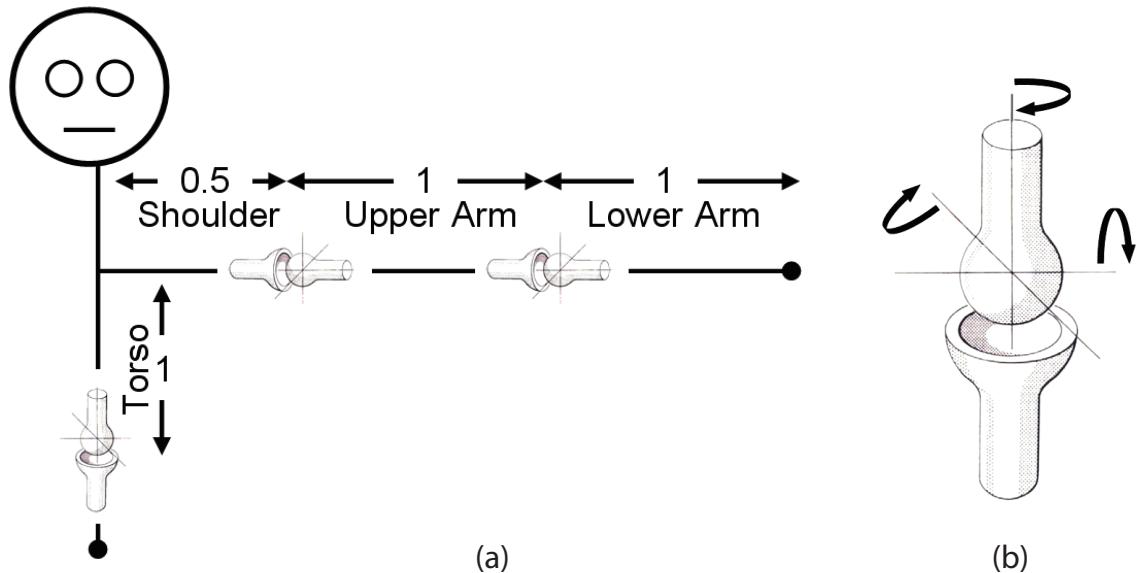


Figure 5.1: Geometrical properties of the proposed body model (a) and ball-and-socket joint (b)

impossible. By using a human body-model we can calculate more general motion primitives such as *moving hands up* irrespective of the exact orientation of the hands and the exact execution of the movement. As a result, the subsequent classification becomes much easier. In the following we describe the calculation of various such movement primitives as well as posture features.

Figure 5.1 (a) contains a sketch of the human upper body frame (only left body part is illustrated) divided into five major limbs, namely the torso including the shoulders, the upper arms and the lower arms. We regard our model as an open-chain collection of rigid limbs. With this modeling, the limbs are connected to each other by ball-and-socket joints (see Figure 5.1 (b)). Therefore each limb has three rotational degrees of freedom (DoF) relative to its adjacent. The entire upper body model thus has a total of 15 ($5 * 3$) degrees of freedom, and five joints. To calculate the direction of each individual body limb, we consider the 3D orientation information of five inertial sensors (Section 3.2.2 *Hardware - MTx inertial measurement unit*) located at the user's upper and lower arms and the torso. As illustrated in Figure 5.1 (a), we estimate the geometrical and inertial properties of the human frame. For the torso, the upper and the lower arm, we presume a length of 1 unit. The torso width (right and left shoulder) is hypothesized as 1 unit. As we will show in the following, the introduced body model allows a richer representation of human movements although it is still very simple.

However one could imagine more sophisticated models taking into account both translation and rotation of body limbs or additional body joints and constraints. In addition, more accurate numerical values for the geometrical properties of the human frame and its major segments (e.g. user dependent) could improve the body model and results. More elaborate approaches (e.g. [Zatsiorsky 1997, Moven 2009]) also consider the whole body

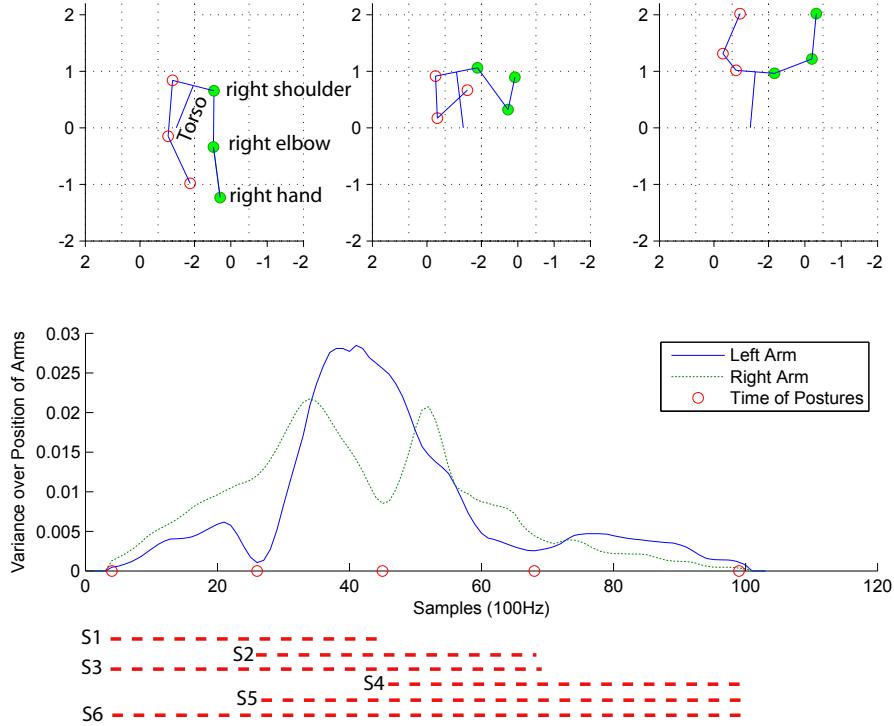


Figure 5.2: Upper body-model (top) and variance over hand positions (bottom) including segmentation while opening a hood based on 5 XSens inertial sensors.

and the fact that a human body and its joints cannot be modeled as a pure kinematic chain with well-defined joints such as hinge-joints and ball-and-socket-joints. Here, orientation and position changes of the body segments are continuously updated by using a biomechanical model of the human body giving physical constraints for the body segments' constellation.

Note that not all degrees of freedom are considered at once when calculating the body-model based features (Section 5.3). Whereas the calculation of twist primitives takes into account the twist of the lower left or right arm (1 DoF each), bending primitives only depend on the torso's bending (3 DoF). Height and push-pull primitives and direction histograms are calculated based on the hands' position in the 3D reference system (11 DoF: discard twist of the right/left and lower/upper arm).

Figure 5.2 illustrates three snapshots of the resulting 3D body-model (depicted as stick figures) while opening the hood of a car from a back perspective. For illustration, the left figure is labeled with the corresponding body extremities where the user moves down both hands to grasp the hood. Lifting both arms in the middle figure, the user reaches the final hand positions above the head illustrated in the right figure. While body-models have been proposed for motion capturing by [Moven 2009] and have been used for activity recognition in computer vision [Ryoo and Aggarwal 2006], we are not aware of similar work in the area of activity recognition using body worn sensors.

5.2.2 Segmentation

Segmentation of the continuous data stream may be seen as a filter for subsequent classification. Ideally the result of this filtering step are segments that are easier to classify than the continuous data stream. While performing activities, the movements (trajectory) can be described by a sequence of upper body postures as described in Section 5.2.1. We observed in Section 3.4 that many activities in a continuous data stream are enclosed by short but fixed positions of the hands (abbreviated SFP in the following sections). SFPs often occur at the beginning and end of activities. We also observed in Section 4.3.1 that turning points (TP in the following sections) in hand movements can help to segment activities. For both SFPs and TPs the variance in the hand position over a small time interval will be lower since the user slows down the speed of movement.

Next we illustrate this segmentation procedure using *Open Hood* as sample activity. Figure 5.2 displays snapshots of the trajectory while opening the hood. We calculate the variance over the hand positions in the 3D body-model. As previously mentioned, the variance of the hand positions will be lower for SFPs and TPs in arm movements. Between SFPs and TPs, the variance will increase. The lower picture illustrates the variance of the user's right hand positions as a dotted green and the corresponding left hand positions as a solid blue line in the course of the activity.

Note that calculating the variance over the lower arms' orientation as applied in Section 4.3.1 is often not enough to identify all SFPs and TPs, for example when pulling the arm. Here, the orientation of the lower arm does not change and therefore does not indicate a SFP or TP of the hand. Therefore this novel body-model based segmentation procedure effectively generalizes the previous works of Section 3.4 and Section 4.3.1.

For the segmentation we detect local minima within the variance of the hand positions separately for both hands. Each local minimum gives us a potential SFP or TPs enclosing an activity (*cf.* red circles in lower Figure 5.2). In the subsequent activity classification (Section 5.4), we will have to test on segments of a specific minimum and maximum length enclosed by pairwise combinations of the detected local minima. As can be seen in the figure (red dashed lines), the displayed time interval contains six segments (S1-S6) complying with the time constraints that are evaluated in the subsequent classification step.

To quantify the quality of the segmentation algorithm, we evaluate the performance on the entire activity set as well as on the background set. Overall, the data set described in Section 5.5.1 contains more than 280 minutes. The mean length of the analyzed activities is 1.4 seconds with a standard deviation of about 0.95 seconds. As the segmentation procedure is used as a filter for the subsequent activity recognition step, it is important that all activities are contained within these segments. This is the case as the average distance between the annotated start and end times of activities and a detected SFP or TP is less than 0.15 seconds. The distance is actually lower than the expected variance in annotation accuracy.

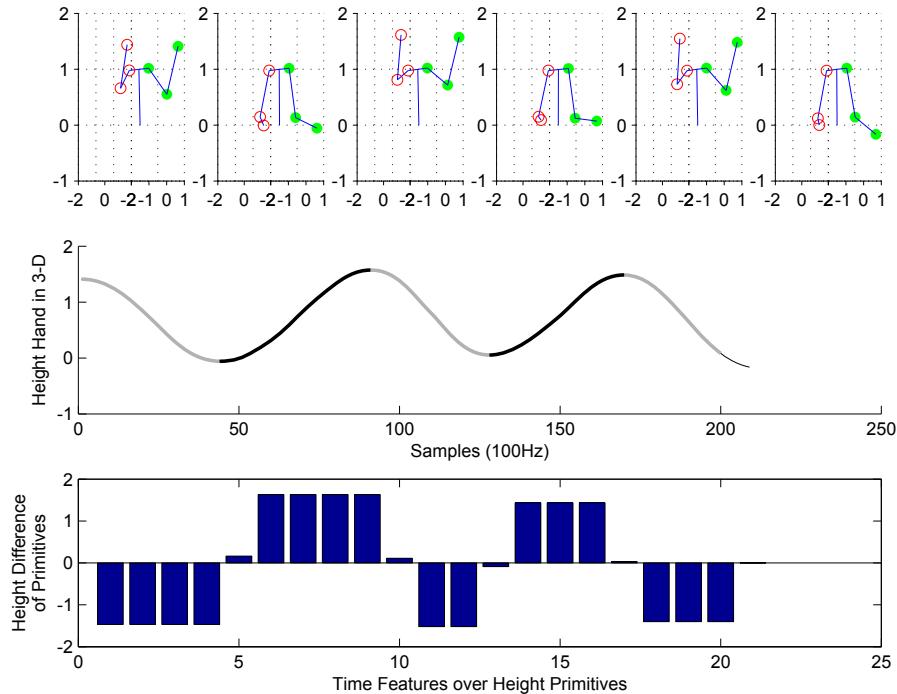


Figure 5.3: Illustration of height features: Snapshots of the activity *Check Trunk* (Top). Right hand's height calculated using the 3D model (Middle) including areas of up (black) and down (gray) movements. Time features over primitives (Bottom).

These numbers show that the algorithm is capable of finding the relevant SFPs or TPs and segments. Furthermore, the segmentation procedure results in a significant data reduction. Within the data (1,691,418 samples), less than 34,000 segments have to be evaluated. For each of these segmentations, the algorithm extracts various primitives and features to enable reliable activity recognition. The following section introduces three types of features for this purpose.

5.3 Calculation of Primitives and Features

The previous section introduced a method to segment a continuous data stream considering SFPs and TPs. This section describes how we extract discriminant features from the data to enable reliable activity recognition. The analyzed activities can be performed in various ways. The speed is not the only factor that has an impact on the execution of an activity. A user who for instance opens a car door can use both hands in various ways. Considerable variability also occurs when opening or closing the hood or the trunk, where the user is not constrained to use a specific arm. In addition, a change of the user's position to the object will influence the motion trajectory. The main objective of Section 5.3.1 is the identification of general movement primitives, such as moving the arms up or down (for example while opening the hood) or pulling the arm toward the body or turning the

torso (for example while opening a car door). These primitives are based on the 3D body-model as presented in Section 5.2.1. The second feature set considers postures within the segments of interest (Posture Features). The last features are calculated on location data using an ultra-wide band (UWB) system from Ubisense [Ubisense 2009] attached to the user's chest (Location Features).

5.3.1 Motion Primitives

Taking into account the 3D body-model from Section 5.2.1, we are now looking for primitives characterizing basic arm movements like up or down (Height Primitives), push and pull (Push-Pull Primitives), the bending of the body forward and back (Bending Primitives) and arm twisting (Twist Primitives). Additionally, a fixed car position leads to similar directions of the arm movements for different people and instances of the same activity. Therefore, we also calculate histograms on the movement direction of both hands (Direction Histograms).

Height Primitives

Figure 5.3 illustrates a series of snapshots of the activity *Check Trunk* where both hands are recurrently moved up and down in order to test the trunk's hinges. As an example, the middle plot displays the right hand's height calculated using the 3D model (left hand analog). By applying a sequential minimum-maximum search, we divide the segment into areas of up (black) and down (gray) movements as color-coded in the figure. We detect five height primitives in this example.

We calculate features on the detected primitives as the next step. A temporal coding (time features over primitives) of a fixed length is illustrated in the lower part of Figure 5.3. The segment is divided into twenty equally spaced bins. Each bin is assigned the height difference of the associated height primitive (20 features). Furthermore, we add the number, average, maximum and minimum of up and down primitives for the analyzed segment (8 features). Finally, a histogram of the primitives' length, normalized by the segment's overall length, is included (5 features). All together, 66 height features for both hands are included in the subsequent training and classification steps.

Push-Pull Primitives

The calculation of push-pull features is similar to the height features. To enable the calculation of these features we introduce the following preprocessing step. We project the data points of the 3D body-model onto the plane defined by the gravity vector as the normal.

Figure 5.4 illustrates a resulting sequence of snapshots for the recurrent pushing and pulling of the right arm from the bird's-eye view, for example, as it occurs for the activity

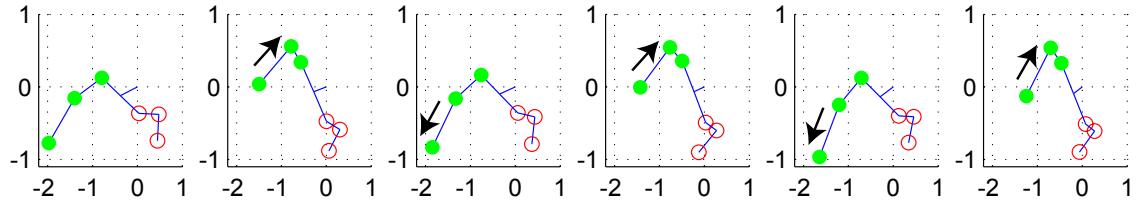


Figure 5.4: Illustration of a resulting sequence of snapshots for the recurrent pushing and pulling of the right arm from the bird's-eye view, for example, as it occurs for the activity *Check Right Door Lock*. The black arrows emphasize the movement's direction of the right hand at each point in time.

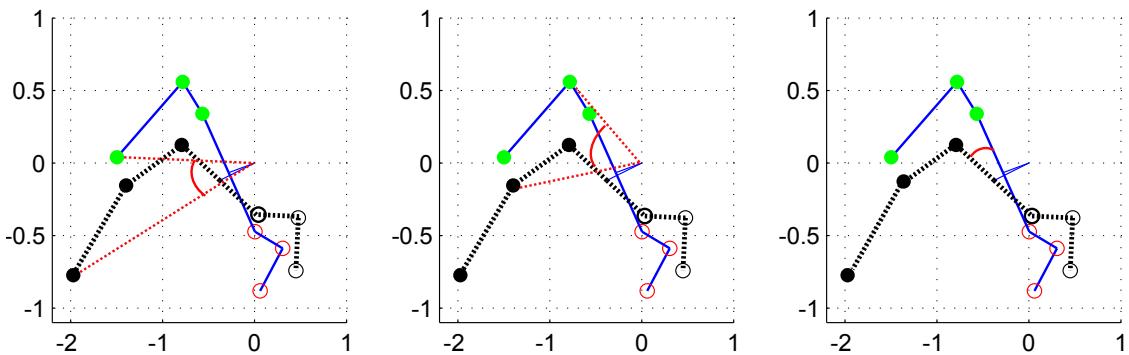


Figure 5.5: Start (dotted black) and end (solid blue) postures while pulling the arm towards the body. Primitives are extracted based on hand (left) and elbow (middle) movements or torso turn (right)

Check Right Door Lock. The black arrows emphasize the movement's direction of the right hand at each point in time. As previously mentioned, various motions can indicate the illustrated hand movement. First, the positions of the hand and elbow towards the torso change. Similarly, a turning of the torso is a sign for the push and pull movements.

Figure 5.5 depicts the start (dotted black) and end (solid blue) postures while pulling the arm towards the body. The first push-pull primitive is calculated on the angle included by the connection lines of the torso to the right hand at the beginning of the segment and the current point in time (*cf.* left figure 5.5). Obviously, this angle will increase when pulling the arm and decrease again while pushing.

The upper figure 5.6 represents the angle at each point in time in the push-pull segment (*cf.* figure 5.4) including three times pulling and two times pushing. Analogously to the height primitives, we apply minimum-maximum search to segment five primitives as color-coded in Figure 5.6. Similarly, we approach the second kind of push-pull primitives. This time, we calculate the angle included by the connection lines of the torso to the right elbow at the beginning of the segment and the current point in time (see middle plot of Figure 5.5). The resulting primitives can be seen in the middle of Figure 5.6. As motivated before, the last push-pull primitive is the turning of the torso as given by the angle between the body axis at the beginning of the segment and the current time (see Figure 5.5 right).

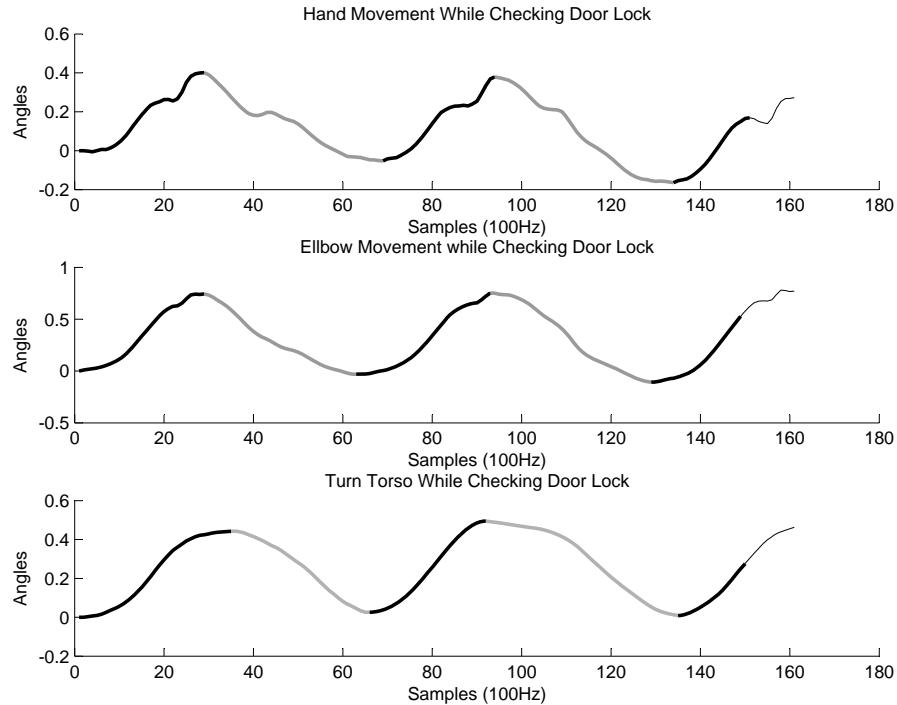


Figure 5.6: Segmented push-pull primitives by torso turn, hand and elbow movements

As before, several features are calculated on the detected primitives (compare to Section *Height Primitives*). Beyond a temporal coding (time features over primitives) of a fixed length and a normalized histogram of the primitives' length, we add the number, mean, maximum and minimum value for each push-pull primitive. 165 push-pull features (66 hands, 66 elbows, 33 torso) are added to the feature set.

Bending Primitives

In addition to the use of arms and the turning of the torso, one can observe that users often bend the torso to support the activity, for example when opening the hood or opening two doors at once. To calculate primitives describing the forward and backward bending of the torso, the minimum-maximum search is applied to the angle included by the torso's direction and the gravity. Clearly, this angle increases/decreases when bending forward/backward. The features are calculated analogically to the height and push-pull features (33 features).

Twist Primitives

Twist primitives are motivated by activities that cause the user to twist one or two arms, for example when turning a knob. They are not calculated using the body-model like the previous primitives, but directly on the integrated gyroscope data of the lower arms.

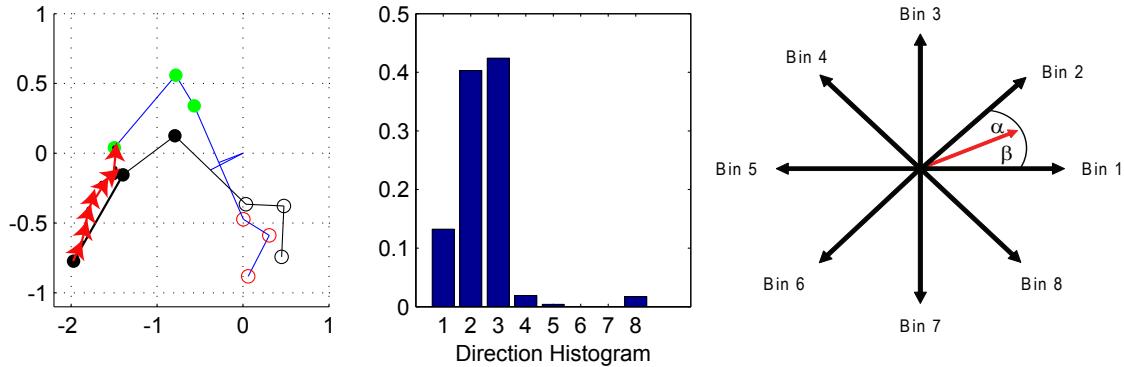


Figure 5.7: Pull movement (left), direction histogram of the right hand movement (middle), direction definition for the histograms (right).

A rotation of a lower arm in a clockwise direction leads to an increasing sum, whereas a counterclockwise turn causes a decrease. Minimum-maximum search over the sum yields the twist primitives. The features are calculated analogically to the primitives before (66 features for left and right arm).

Direction Histograms

A fixed car position leads to similar directions of the arm movements for different persons and instances while performing an activity. Therefore, we finally calculate histograms on the movement direction of both hands (Direction Histograms). For illustration, we will again use the previous example of a pull movement. In a first step, we calculate the directional vectors of succeeding hand positions. The directional vectors for the pull activity are displayed as arrows in Figure 5.7 (left). We define eight directions in the two dimensional plane as illustrated in the right picture. The directional vectors of the segment are then assigned to eight bins of a histogram represented by eight directions. This assignment leads to the direction histogram illustrated in the middle picture of Figure 5.7. We observe high values for bin two and three. Visually this distribution is a good representation for the performed movement in the left picture of Figure 5.7.

To make the feature robust to speed-difference the histogram is normalized by the overall length of the current segment. Also, we use soft assignment to the two neighboring bins where the assignment is proportional to the angle of the directional vectors. An example is illustrated in Figure 5.7 by the red short arrow in the right figure. Here the direction lies between the directions of *bin 1* and *bin 2*. Therefore, it is assigned with a percentage of $\frac{\alpha}{45}$ to *bin 1* and with a percentage of $\frac{\beta}{45}$ to *bin 2*. The resulting 16 direction features for both hands are added to the feature set.

5.3.2 Posture Features

As seen in the last section, activities can be decomposed into motion primitives. One can now ask the question whether discriminant features over body postures can help improve the recognition results. Looking at the activity *Fuel Lid*, it is clear that all users have to keep the right hand in a similar height to contact the fuel knob. It should also be noted that this height level will not change much while performing the activity. This observation is different from activity *Open Hood* where the arm height is changing very quickly. While writing, both hands remain in a similar height and orientation towards gravity. Here, one hand holds the notepad and the other writes with a pen. In addition to motion primitives we will consider postures that can help to distinguish between activities and the background. In the experiments described below, we consider the following postures: the arms' orientation towards gravity (6 dimensions), the distance between the two hands (1 dimension) and the hands' height (2 dimensions), as well as the torso's relative direction to the car (2 dimensions). We add the minimum, maximum, mean and variance over the postures (44 features) to the feature set.

5.3.3 Location Features

The worker's relative position to the car body is estimated using an ultra-wide band (UWB) system by Ubisense. Sensor data of four tags is smoothed with a Kalman filter before the four sensor sources are combined. Despite the use of multiple sensors, the data becomes patchy at certain times. In a preprocessing step, we estimate location values for these gaps by linearly interpolating between known values in time. Based on these location values, two different location representations are calculated and added to the feature space. First, the median of the enclosed x and y-values of a specific segment are added (2 features). Second, we calculate the angle included by the first basis vector and the connection of a specific location point to the car center M. The median of this angle for a segment is also added to the features (1 feature).

Figure 5.8 illustrates the accuracy of the preprocessed location data for the twenty annotated activities. While location is a good indicator which subset of activities might be performed, it is clear that location alone will not be sufficient to distinguish all activities. Especially behind and on the sides of the car are various overlapping areas for different activities. In the case of the activity *Writing* (see black diamonds), location will not be helpful as the activity is spread over the entire area.

5.4 Multi-Activity Recognition using Joint Boosting

The final step of our algorithm classifies the different segments into multiple activities using the motion, posture and location features introduced in the previous section. We employ joint boosting as it allows to learn a runtime efficient classifier by sharing features

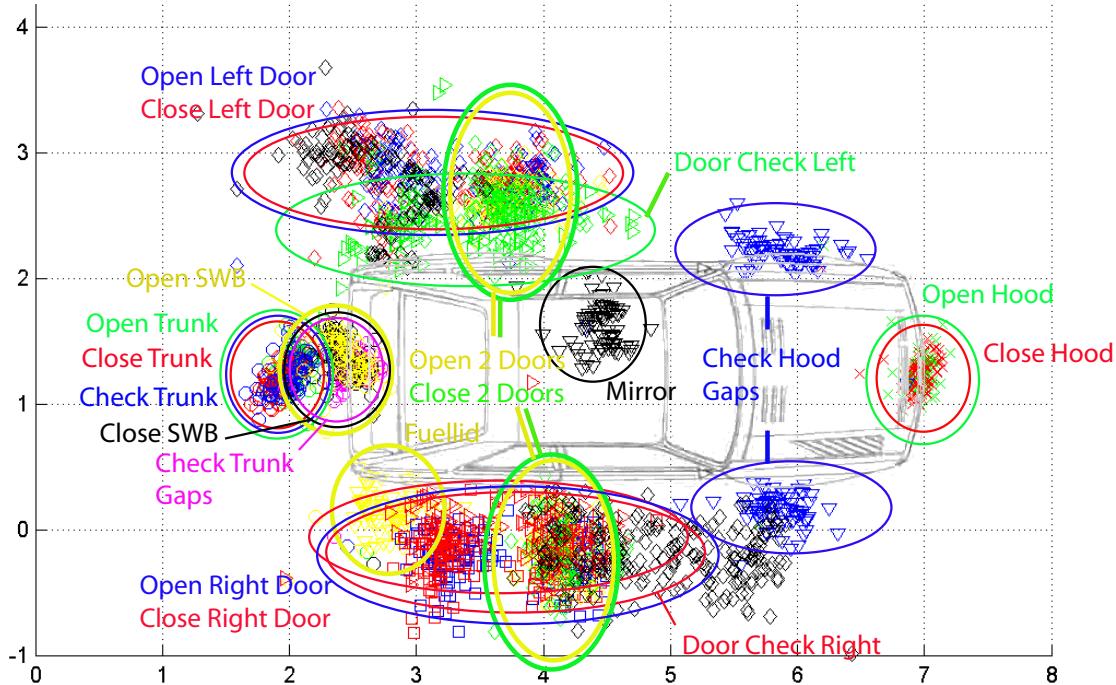


Figure 5.8: Illustration of the users' location while performing the 20 activities after preprocessing. For Writing (black diamonds), the location is spread over the entire area.

across multiple activities. See Section 2.3.2 for an introduction of the joint boosting algorithm.

Our experiments (Section 5.5.2) indicate that groups of similar activities are separated during the first rounds and are disambiguated among each other in later boosting rounds.

To further reduce the number of required rounds we counted mis-classification on background samples as error for the activity classifiers but did not learn a classifier for the background class. In order to handle the highly unbalanced training sample distribution among classes we adapted the weight initialization such that for each activity the class specific instance weights for positive samples as well as for negative samples sum up to 0.5.

Calculation of test and training vectors Activities in the training data are annotated. Therefore, we can directly calculate all features (Section 5.3) on the positive training segments. The resulting feature vectors can be used as direct input for the training phase of the joint boosting algorithm.

Calculation of the features for the negative training segments and the test data is more complex. As previously discussed in Section 5.2, we can automatically detect short fixed arm positions and turning points in the continuous data stream. However, the number that we detect is larger than the number needed to segment the activities from the background stream (see S1-S6 in Figure 5.2).

From the training data we extract one minimal and one maximal length over all considered activities and allow for $\pm 30\%$ variance of these values. When calculating the features in the test data and the training data (negative instances), all possible combinations of segments with length between the minimum and maximum length are analyzed. For all resulting segments, the features are calculated. Note that a negative instance on the training data is only used as input for the training phase of the joint boosting algorithm if it does not overlap with any activity instance. All test segments are classified by the boosting algorithm. As the writing activity is considerably longer than all other activities, we split the annotation of activity *Writing* into sequences shorter than four seconds. The advantage is that the overall maximal length of all activities becomes shorter and the overall number of segments that need to be classified drops significantly. The only drawback is that we obtain multiple segments classified as writing for a single continuous writing activity which have to be merged in a post-processing step.

5.5 Experiments and Results

The specific scenario that we study is composed of 20 activities in quality inspection of a car production process. The next paragraph shortly introduces the previously published data set [[Ogris et al. 2008](#)]. Subsequently, we evaluate the distinctiveness of each activity separately with respect to the collected background data and the remaining activities.

5.5.1 Car-quality control data set

The data were collected in a large industrial project on the use of wearable technology in production environments [[Stiefmeier et al. 2008](#)]. During the experiments, data of a wearable system composed of 7 motion sensors (Section [3.2.2 Hardware - MTx inertial measurement unit](#)), 16 force sensing resistors (FSR) for lower arm muscle monitoring and 4 ultra-wide band (UWB) [[Ubisense 2009](#)] tags for tracking user position were collected. In this chapter and in Chapter [6](#), we use the data of 5 motion sensors (located at the user's upper and lower arms and torso) and the 4 UWB tags for activity classification. A list of the twenty activities can be seen in Figure [5.9](#).

5.5.2 Evaluation of the model-based approach to activity recognition

This section evaluates the distinctiveness of each activity separately with respect to the collected background data and the remaining activities. In each validation round, we calculate the probability for all detected segments. As proposed in [[Ogris et al. 2008](#)], we evaluate each segment as follows: if a considered segment overlaps more than 50% with the annotation of a specific activity, it will be counted as a true positive. Clearly the number of false positives grows while increasing the recall for the actual activity.

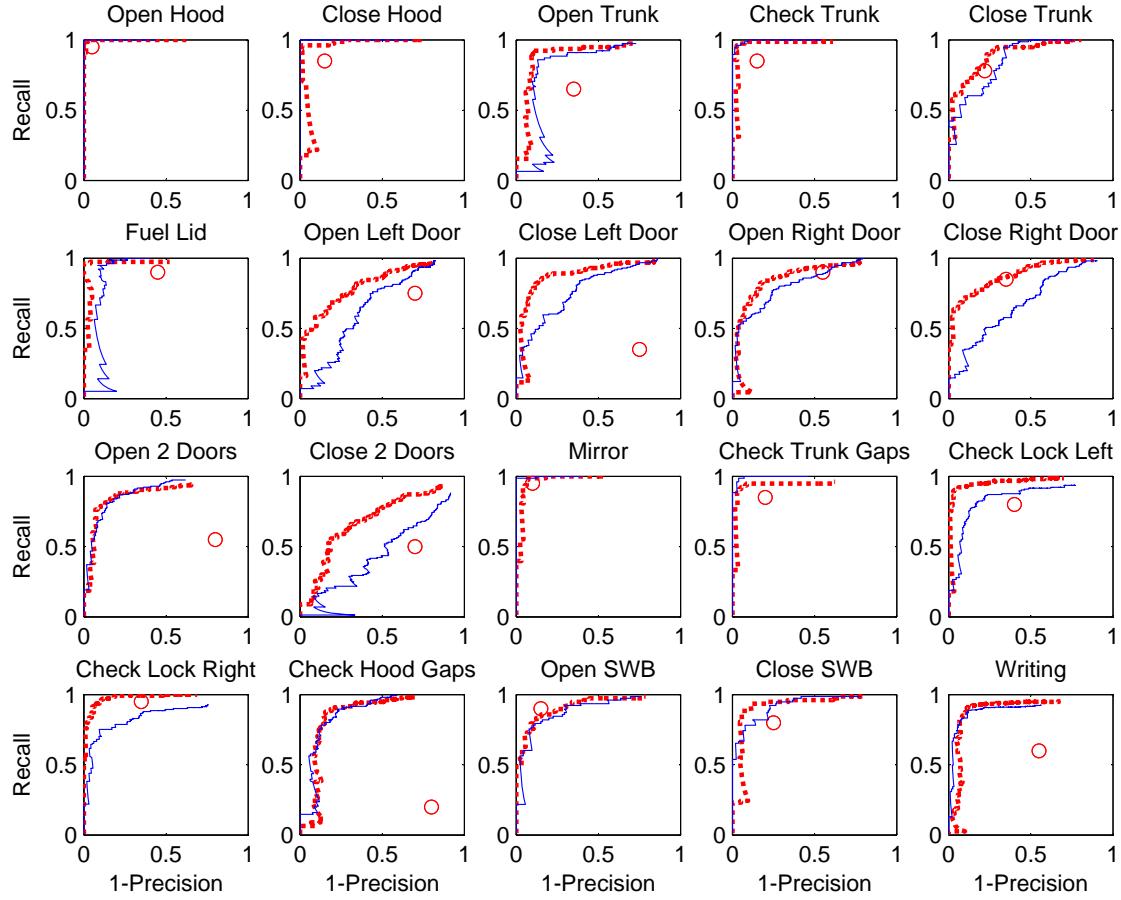


Figure 5.9: ROC curves for the 20 activities (dotted red: user-dependent, solid blue: across-user). The red circles are the user-dependent results of Ogris et al. [Ogris et al. 2008] (SWB = spare wheel box)

Ideally both precision and recall are 100%. The following paragraphs report on two types of experiments. In the *User-dependent experiments*, we analyze the performance of the recognition system for user-dependent training and evaluation. This is the same setting as used in [Ogris et al. 2008]. In *Across-user experiments*, we report for user-independent training and testing. In *General results*, we summarize general conclusions and analyze in more detail how joint boosting works for our activity classes.

User-dependent experiments

As mentioned before, each user performed the activity sequence 10 times. We perform leave-one-instance-out cross-validation for each user. The red dotted plot in Figure 5.9 summarizes the results for each activity.

For ten out of twenty activities, the system achieves very good performance with a recall and precision greater than 0.9. For six of those activities, an almost optimal

performance can be observed. The activities *Check Hood Gaps* and *Writing* follow closely with a recall of 0.85 and a precision of 0.85. Most door related activities and *Open Spare Wheel Box* still obtain a recall of 0.75 for a precision of 0.75. *Close Two Doors* performs worst with a recall 0.6 for a precision of 0.75.

As our experiments show, the new recognition method yields significant performance improvements compared to the user dependent approach introduced by [Ogris *et al.* 2008] (see red circle in the figure) in 14 out of 20 activities. For five activities our system performs similarly. Only for activity *Open Spare Wheel Box* (Open SWB), our system is marginal worse as can be seen in the dotted red plot.

Across-user experiments

Beyond user dependent experiments, we perform leave-one-user-out cross-validation to analyze the system performance across users (see blue plot in Figure 5.9). As we have data from eight different users for our activity set we effectively perform eight-fold cross validation across the eight users. Interestingly, the novel approach proposed in this chapter still achieves good detection rates across users. Whereas the system achieves similar results for ten out of twenty activities, the precision-recall ratio drops for door related activities. Considering the difficulty of the dataset, the results show that the proposed features and primitives derived from a human body-model make activity recognition robust even across user. In three cases, the results of the across-user experiments are better than the user-dependent (*Close Hood, Mirror, and Check Trunk Gaps*) because of more representative data and a larger training set. Interestingly, the across-user results of our system still outperform the user-dependent results introduced by [Ogris *et al.* 2008] for 13 of the 20 activities while performing similarly for three.

General results

Many of the activities with lower recognition performance are door related. Visual inspection of the corresponding body-models indicates that magnetic disturbances occur for these activities. This might be a reason for the decreased performance. Additionally, opening and closing car doors allows for a greater degree of variability during execution than most other activities.

To gain insight into how joint boosting works on this data set, it is helpful to examine which features are selected and when. As previously mentioned, joint boosting shares weak classifiers across multiple classes.

Figure 5.10 shows the final set of features selected in the first 25 rounds and the sharing matrix that specifies how the different features are shared across the 20 activity classes. Each row corresponds to one feature and each column shows the features used for each activity class. An entry (gray box) in cell (j, i) means that activity i uses feature j . The

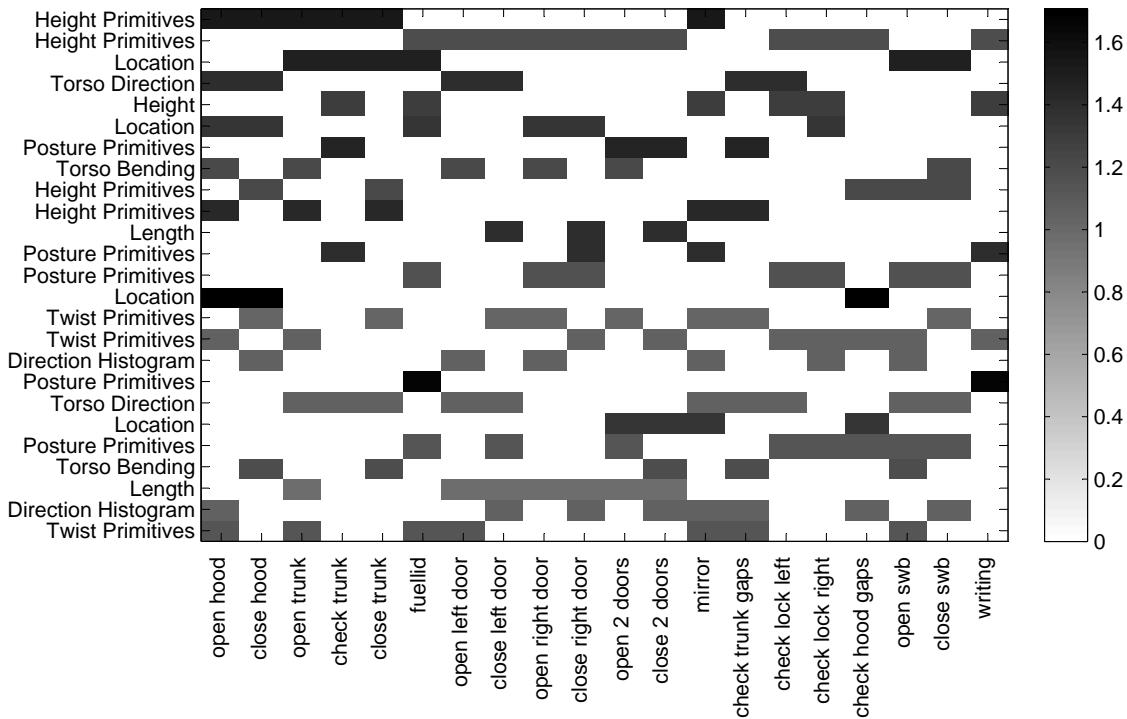


Figure 5.10: Matrix that relates features to classifiers from top to bottom, which shows which features are shared among the different activity classes. These features were chosen from a pool of 393 features in the first 25 rounds of boosting.

darker the entry, the stronger does the corresponding feature vote for the class. Lighter entries are weighted less.

In the first round, the algorithm selects height primitives to separate activity classes 1 to 5 and 13 from the remaining classes. The selected activities share distinctive height primitives while interacting with the hood, trunk or mirror. The second round characterizes activities with low height primitives. In rounds number three and four, boosting considers more features shared by many classes. Whereas the first location feature is shared by all activities in the back of the car, the torso direction helps to characterize activities in the front and on the front left side of the car.

The figure illustrates that the features are shared between classes in a way that is not tree-structured. A hierarchical partitioning is therefore not possible. However, the figure also shows how joint boosting reduces the computational complexity, by finding common features that are shared across several classes. Already after less than five rounds, features shared by all classes are considered. Furthermore, all features used in the first ten rounds are shared by at least four classes.

In the last paragraph we described how boosting is able to reduce the weighted squared error on the training data within a few rounds by finding common features shared across several classes. The features selected by joint training in the first rounds are generic

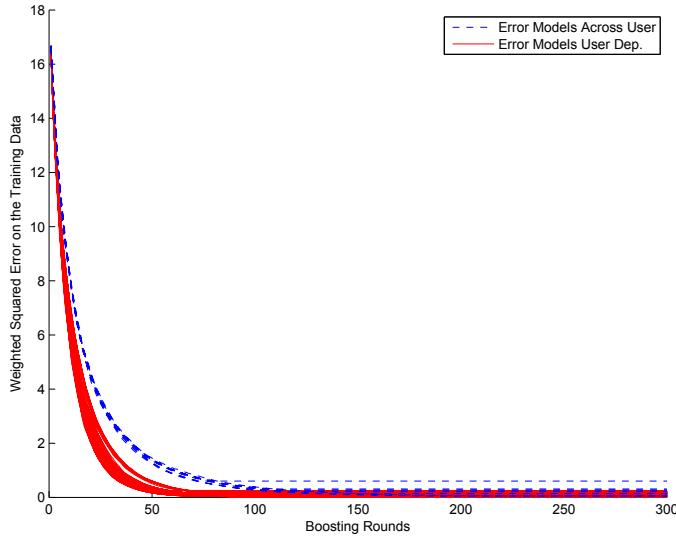


Figure 5.11: Weighted squared error on the training data (y-axis) as a function of the boosting rounds (x-axis). The solid red curves correspond to the eight user-dependent models converging after 40-60 rounds. In case of across-user training (blue curve), boosting requires more classifiers (100-120) to converge.

location and height features, whereas the features chosen in the later rounds tend to be more activity specific.

To gain a little more insight into the obtained classifier we discuss several classes in more detail. Quite interestingly, most introduced features and all primitives are used. The activities *Open/Close Hood*, *Open/Check/Close Trunk*, *Mirror*, *Open SWB*, *Close SWB* and *Check Trunk Gaps* can be characterized by height primitives. In addition, boosting finds bending primitives to be discriminant for those activities. Furthermore, it is not surprising that twist primitives strongly describe activity *Fuel Lid*. Characteristic twist primitives over the entire activity's duration can also be observed for *Open Hood/Trunk*, *Mirror* and *Check Trunk Gaps*. For most door-related activities and *Check Hood Gaps*, the push-pull primitives and the direction histograms become prominent. These activities are mainly characterized by a planar movement of the hand. To support the activities, the users bend the body forward and backward. Posture primitives are mainly used for activities *Check Trunk*, *Check Trunk Gaps*, *Writing*, *Fuel Lid*, *Open Two Doors* and *Close Two Doors*. Beyond the mean and variance of the hands' height (distinctive against background), the distance between the two hands is characteristic for the same activities. Whereas *Open/Close Two Doors*, *Check Trunk* and *Check Trunk Gaps* necessitate a broad distance between the hands, the user keeps the hands close to each other for *Writing* and *Fuel Lid*. Finally, the orientation of the left and right arm towards gravity is specific for the activity *Writing*.

As previously mentioned, most of the features introduced in Section 5.3 are used by the boosting framework. One can wonder how many classifiers or rounds are needed

by boosting to obtain reasonable results for the twenty activities. Figure 5.11 plots the weighted squared error on the training data (y-axis) as a function of the boosting rounds (x-axis). The solid red curves correspond to the eight user-dependent models. As can be seen, the error on the trained models converges after 40-60 rounds. Not surprisingly, in case of across-user training, boosting requires more classifiers (100-120) to converge. This fact shows that activity recognition across user is a more complex task as the variability of activity performances is higher, and therefore boosting has to find more weak classifiers to obtain good results.

5.6 Discussion and Conclusion

This chapter presents a novel model-based approach to activity recognition using high-level primitives that are derived from a 3D human body-model. Using short but fixed positions of the hands and turning points of hand movements, the continuous data stream is segmented in short segments of interest. The method is successfully employed for a car-quality control setting of 20 different activities in a continuous data stream. In the experiments our approach shows superior performance with published results on the same data set (based on string matching).

Our representation of motion and posture primitives on segments has proven effective in both user dependent and across-user activity recognition. The introduced high-level features such as *move the hands up and down* or *turn the torso* or *arm* allow recognition even in the presence of large variability while performing activities.

Our results confirm the strength of the introduced features. Unlike signal-oriented features, body-model features are closely linked to motions that are required when performing an activity, for instance raising one or two arms when opening the hood. The exact execution is not of great importance but rather features like the end position of the hands described by the body-model. First results comparing signal-oriented versus body-model features confirm a superior performance of body-model derived features. Chapter 6 will report on a detailed analysis and comparison of body-model derived and signal-oriented features.

Beyond the modeling aspects, we pay particular attention to computational feasibility. Based on the segmentation, the evaluation can be calculated in real-time. The feature calculation using non-optimized MATLAB code for the data set of 280 minutes takes 230 minutes. The succeeding evaluation is performed in 7 minutes. Indeed, the runtime will increase when recognizing more activity classes. However since joint boosting is able to share features among the classes efficiently, the runtime will still be real-time.

This chapter makes a big step toward user-independent and multi activity recognition. The following chapter evaluates the benefits of body-model derived features compared to standard signal-oriented features.

6

An Analysis of Sensor-Oriented vs. Model-Based Activity Recognition

The previous chapter pursues user-independent and multi activity recognition. Therefore, high-level primitives are derived from a human body-model. Accordingly, joint boosting enables the automatic discovery of important and distinctive features. The primary goal of this chapter is to contribute a systematic and in-depth analysis and comparison of model-based with sensor-oriented activity recognition. In addition, results of incorporating location or utilizing different types of sensors as well as results of reducing the number of sensors are presented.

6.1 Introduction

In Chapter 5, we propose a model-based method for user-independent activity recognition. We first estimate a body-model from five inertial measurement units (IMUs) used to derive high-level primitives such as moving the arms up or down or turning the wrist. On a 20-activity dataset, the approach applying joint boosting achieves good results for both user-dependent and user-independent settings. One can now wonder about the real impact of body-model derived features compared to standard signal-oriented features.

The first goal of this chapter is therefore to answer the question if such model-based methods have indeed potential to advance the state-of-the-art in activity recognition. For this we systematically compare the model-based approach of the previous chapter to more traditional signal-oriented approaches. The experimental results in this chapter indeed indicate that model-based approaches enable more robust activity recognition than signal-oriented approaches and that their combination can further improve recognition performance. As the original approach requires the use of five relatively expensive and power-hungry IMUs, we extend the approach to reduce sensor requirements. To this end we propose an alternative model-based approach that does not require the use of IMUs but instead uses accelerometer sensors only. We explore the possibility to reduce the number of sensors required to merely two sensors attached to the wrists of the human. Last but not least we also analyze the importance of location information for activity recognition.

Section 6.2 describes additional evaluated features. Section 6.3 reflects the method applying joint boosting and introduces an approach using hidden Markov models to classify multiple activities. Section 6.4 describes the evaluation procedure and Section 6.5 summarizes the experimental results. Finally, the main contributions of the chapter are discussed in Section 6.6.

6.2 Additional Features and Adapted Body-Model

The primary goal of this chapter is an evaluation if the model-based method of Chapter 5 has the potential to advance the state-of-the-art in activity recognition. Section 6.2.1 briefly summarizes the body-model derived primitives and location features as presented in Section 5.3. In addition, features based on sensor data of IMUs for an evaluation of a reduced number of sensors are presented. Section 6.2.2 introduces and discusses a novel method to estimate a human body-model based on acceleration sensors only. By applying this method, the sensor requirements of the IMU based approach in Chapter 5 can be reduced. Section 6.2.3 recapitulates common signal oriented features as used for example by [Lester *et al.* 2005].

6.2.1 Body-model by IMUs (BM_IMU) and location features

To derive features or primitives like moving the hands up, turning the arm, or keeping the arm in a specific posture, Chapter 5 introduces a 3D human body-model (BM_IMU). Based on BM_IMU, several data streams for each point in time are calculated including the hands' height, the arms' twist, the torso's bending or twisting, the distance between the left and right hand as well as the arms' orientation towards gravity. Furthermore, the torso's global orientation as well as the angles included by the connection lines of the torso to the hands and elbows can be estimated. This data provides the basis for the subsequent extraction of motion and posture primitives. For a detailed description how motion primitives and posture primitives are calculated based on BM_IMU, we refer to Section 5.3.

We also evaluate these features using only two wrist-worn sensors in Section 6.5. Obviously not all features can be calculated. Whereas features on height primitives and rotation features as well as postures can be estimated considering only the wrist sensors, all other primitives (push-pull, bending, and twist) cannot be detected. Furthermore, the distance between the two hands as well as the torso's direction cannot be estimated. Instead, the average orientation of the two wrist sensors approximates the torso's direction when using two sensors only.

Location features as described in Section 5.3 are considered to analyze the importance of location information for activity recognition. For each segment, the means of the (x,y) co-ordinates provided by the Ubisense [Ubisense 2009] location system are calculated.

6.2.2 Body-model by acceleration (BM_ACC)

Chapter 5 shows how to use a body-model to extract abstract features that improve recognition results. Having access to precise data from inertial measurement units (IMUs), respectively to the global orientation of the sensor, it is straightforward to determine the angles of joints and hereby the configuration of the body. However, IMUs come with the price of power, are harder to embed into wearable items and are still expensive. While the results are less accurate than using IMUs, accelerometer-based approaches also allow to estimate the sensor's orientation [Mizell and Cray 2003] and have been used to create low cost and power efficient motion capture systems [Farella *et al.* 2007, Tiesel and Loviscach 2006, Slyper and Hodgins 2008]. In this section we introduce a novel method to estimate a human body-model from acceleration sensors only and then briefly discuss difficulties and drawbacks compared to the BM_IMU. Section 6.4 analyzes the impact of this novel model on recognition performance.

We use the effect of the earth gravity vector G on the 3D acceleration values as reference to estimate the orientation of each sensor individually. Additional acceleration caused by human movement obviously influence the estimate of the direction of G . To reduce this dynamic motion component we smooth the signal by calculating the mean of the acceleration on a sliding window of 120ms. The direction of the normalized acceleration vector is taken as estimate of the earth gravity vector. This vector is then used as the sensor orientation with respect to the ground plane.

Some of the remaining ambiguities can be resolved by adding simple constraints, before creating the kinematic chain of the human body. The following paragraphs describe the constraints that disallow unnatural poses.

Upper Arm When using acceleration only to estimate the orientation of a limb, unnatural postures of the upper arm are allowed. An example is illustrated as dashed line in Figure 6.1 (a). Dynamic motion affects the estimate of the orientation around G and might lead to those postures. To disallow these unnatural poses we simply mirror the arm vertically back into a hemisphere as shown in Figure 6.1 (a) .

Lower Arm Bending Figure 6.1 (b) shows the allowed forearm postures. We obtain the lower arm's direction toward gravity by observing the axis along the elbow, shown as a blue arrow in the figure. If the arm is directed towards the ground, the full earth gravity affects the sensor. The higher the arm moves, the less the sensor is affected by gravity. Hereby we constrain the lower arm to bend to the front perpendicular to the shoulder.

Lower Arm Rotation To represent the rotation along the elbow, we use the two axes perpendicular to the lower arm. This works best on the ground plane and is less precise the closer the rotation occurs around the axis of gravity due to the fact that rotation around G does not change the acceleration.

After applying the constraints we concatenate the obtained orientation vectors as before, starting from the torso to the hand, to estimate the body configuration. As a result,

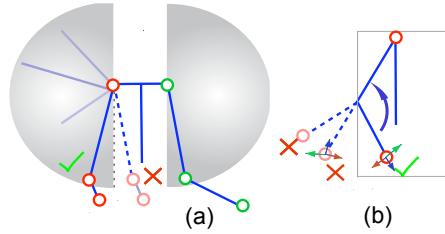


Figure 6.1: (a) The upper arm rotational space. (b) The lower arm constraint: The rotational space of the lower arm around is constrained to the front.

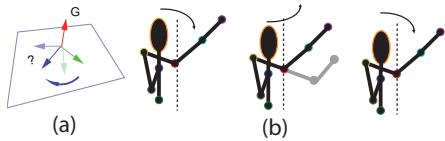


Figure 6.2: (a) Rotation around the gravity vector G leaves vectors on the ground plane undetermined. (b) As consequence the push-pull-primitive using the BM_ACC on horizontal plane cannot be determined. The gray limbs denote the actual posture. The black lines the posture of the BM_ACC

using acceleration as basis we can achieve a similar kinematic chain as using IMUs. One drawback using this approach is its limitation to activities with large and fast movement. Fast motion strongly influences the acceleration values and hereby the estimate of the gravity vector.

Lacking the rotational information around G , we introduce a systematic error. As seen in Figure 6.2 (a) rotating on a plane perpendicular to G does not change the gravity field, i.e., it does not affect the acceleration. Thus, the orientation cannot be precisely estimated. This affects the body-model as follows. Figure 6.2 (b) shows a sequence of the push and pull primitive. Hardly any rotation around the shoulder is measured, if the upper arm is aligned to the horizontal plane.

In this section the BM_ACC was described. To enable direct comparison of this approach with BM_IMU, we use the same motion and posture primitives as described in Section 6.2.1.

6.2.3 Signal Features

An evaluation of body-model features compared to signal-oriented features was motivated in Section 6.1. Whereas sections 6.2.1 and 6.2.2 introduce features based on two body-models, we briefly explain common signal-oriented features in the following of this section. Most research on activity recognition successfully base their feature calculation directly on a sensor signal [Bao and Intille 2004, Ward *et al.* 2006, Kela *et al.* 2006]. Typically, the features are calculated separately for each dimension.

Using 5 IMUs, features for 45 dimensions are calculated (3D orientation, 3D acceleration, 3D gyroscope per sensor). Using 5 acceleration sensors, only 15 dimensions will be considered. Note that for the evaluation *Benefit of Location*, two more dimensions will be added. In case of *Reducing Number of Sensors*, the dimensions will decrease.

For each dimension, for example a sensor's first acceleration dimension, a set of features in frequency and time domain is calculated. First, the Fast Fourier Transform (FFT) maps the incoming signal into the frequency spectrum. We group the FFT-coefficients into five logarithmic bands (5 features). The resulting absolute and real values are added to the feature set. Furthermore, we calculate the cumulative energy of the Fourier series. In addition, 10 cepstral coefficients are calculated modeling the spectral energy distribution. The spectral entropy (1 feature), which gives a cue about the complexity of the signal is also added. As features in the time domain we calculate the mean and variance of the signal (2 features). Hence we obtain 19 features per dimension. Given 5 IMUs and the additional 2D location, the feature calculation yields a total number of 865 features.

6.3 Multi-Activity Recognition using Joint Boosting and HMMs

In the following of this chapter, we apply two different methods to classify activities based on the features introduced in the previous section and Section 5.3. Method number one is our body-model based approach using joint boosting as presented in Chapter 5. The second method is based on hidden Markov models (see Section 2.3.1). As hidden Markov models (HMMs) allow to capture temporal regularities they are often used for activity recognition. Section 6.3.1 therefore describes different HMM methods evaluated in Section 6.4.

For evaluation, we use the car-quality control data set introduced in Section 5.5.1. Activities are hand annotated and all features can directly be calculated on the positive training segments. Boosting uses directly the features as described in Section 6.2 and the features for HMMs are explained in Section 6.3.1. The respective feature vectors are then used to train joint boosting and the HMMs. Calculation of feature vectors for the negative training and test data is similar to Section 5.4 applying a segmentation of the continuous data stream. For all resulting segments, features are calculated and they are either used as negative training samples for joint boosting or classified in the test case with joint boosting or the HMMs.

6.3.1 Hidden Markov models (HMMs)

HMMs (see Section 2.3.1) have been successfully used in modeling different types of time series [Lester *et al.* 2006, Ward *et al.* 2006, Zappi *et al.* 2008]. Later we will evaluate different algorithms and also different types of HMMs. In particular we are interested

to analyze how different types of HMMs may capture temporal regularities of short and non repetitive activities in our scenario. Since activities in the training data are hand annotated, we can train an HMM for each activity. Regarding the topology of the HMM, we use fully-connected and left-right models with 15 states. Note that experimental results with less than 10 states did not improve the results. Additionally we consider two types of observation sequences. Section 6.2.1 introduces 15 data streams that can be calculated from the BM_IMU, for example the hands' height or the arms' twist. Furthermore, we consider the directional vectors of succeeding hand positions. Within those data streams, movement primitives are extracted and transformed into temporal features over primitives in a similar manner to Section 6.2. Note that both the features of the boosting algorithm and the HMMs are based on the same primitives. We also train HMMs directly on the continuous data stream.

Due to computational reasons we did not evaluate all possible test segments with all 20-dimensional HMMs. Instead we also adopted the segmentation procedure introduced before to reduce the computation time considerably. Using boosting to rank the different segments we choose the best 20% segments of the test set. Note that these segments contain all activities of the test data and that the HMMs therefore can achieve a recall of 100% using this segments only. The 20 HMM likelihoods are then used to rank the remaining segments.

6.4 Evaluation

To evaluate the features and algorithms presented in the previous section we have chosen the car-quality control data set introduced in Section 5.5.1.

Evaluation Procedure We evaluate the distinctiveness of each activity individually with respect to the collected background data and the remaining activities. As a result the number of false positives increases while increasing the recall for an activity. We perform leave-one-user-out cross-validation to enable user-independent activity recognition. In each cross-validation round, we calculate the probability for all detected segments. A segment T will be counted as true positive if the ground truth segment A has the same activity label and if the following equation 6.1 holds true:

$$\text{start}(A) \leq \text{center}(T) \leq \text{stop}(A) \quad (6.1)$$

with $\text{start}(A)$ and $\text{stop}(A)$ correspond to the begin and end times of the ground truth segment A and $\text{center}(T)$ indicates the central time of segment T . This ensures that the analyzed activities are spotted at the right time location. Only if the central time of a segment intersects with the annotated activity, the segment is counted as a true positive. Ideally both precision and recall are 100%. Typically however, the precision decreases when increasing the recall for a particular activity. For brevity we use a single point of the precision-recall curve namely the commonly used equal error rate (EER) where recall

and precision are equal. Additionally, we report the mean equal-error-rate for each setting in our evaluation.

As mentioned earlier the main focus of this chapter is a systematic evaluation of different aspects and their impact on activity recognition performance. In order to make the results as comparable as possible we use the same segmentation procedure introduced in Section 5.2 as a pre-filter of all algorithms. Please note that the remaining segments contain all annotated activities so that all algorithms can obtain 100% recall. Although other segmentation procedures based on two sensors (acceleration or IMUs) exist (see Section 3.4 and Section 4.3.1), the same segmentation procedure for all algorithms is used to make the results directly comparable for the purpose of this chapter.

6.5 Experimental Results

As motivated before, this chapter contributes a systematic evaluation of various aspects of activity recognition algorithms. The presentation is structured in four parts. The first (*Benefit of Body-Model*) compares results from body-model based features to signal-based features. Here we consider two body-models (BM_IMU and BM_ACC) that are either based on IMUs or acceleration sensors only. Results of incorporating location (*Benefit of Location*) are presented in the second part and the third part reports on the results of reducing the number of sensors (*Reducing Number of Sensors*). Finally, we also compare the discriminative joint boosting approach to a generative approach using hidden Markov models (HMM). We conclude with a discussion.

Benefit of a Body-Model

Section 6.2 introduced two body-models either using five IMUs or five acceleration sensors. The top three rows in Table 6.1 contain a comparison of the algorithm for

		with location																			
		w/o location																			
BM_IMU + signal, inertial	.93	.99	1.00	.92	.97	.92	.96	.87	.92	.94	.88	.94	.82	1.00	.97	.92	.89	.92	.97	.94	.95
BM_IMU signal, inertial	.92	.99	.99	.91	.99	.92	.94	.86	.89	.94	.87	.88	.77	1.00	.96	.89	.86	.92	.97	.91	.92
	.88	.95	.97	.86	.91	.82	.92	.74	.85	.86	.88	.87	.76	.99	.99	.88	.90	.77	.95	.90	.93
BM_ACC + signal, acceleration	.64	.65	.59	.72	.79	.59	.63	.44	.44	.44	.38	.65	.30	.95	.82	.71	.69	.71	.79	.62	.94
BM_ACC signal, acceleration	.61	.78	.47	.64	.77	.62	.68	.43	.35	.40	.35	.62	.28	.88	.81	.69	.65	.56	.65	.58	.93
	.57	.56	.44	.46	.46	.59	.71	.32	.44	.28	.24	.88	.24	.88	.82	.78	.70	.72	.44	.27	.91
BM_IMU + signal, inertial	.92	.97	.96	.94	1.00	.94	.96	.85	.89	.92	.83	.92	.76	.99	.97	.88	.87	.93	.97	.94	.95
BM_IMU signal, inertial	.92	.99	.99	.94	1.00	.94	.94	.86	.90	.92	.81	.90	.81	.97	.99	.90	.83	.95	.96	.91	.94
	.90	.96	.96	.90	.95	.82	.91	.83	.88	.90	.90	.88	.75	.99	.95	.84	.82	.92	.94	.91	.94
BM_ACC + signal, acceleration	.81	.94	.97	.87	1.00	.88	.90	.71	.60	.60	.50	.76	.54	.97	.90	.78	.79	.89	.92	.79	.95
BM_ACC signal, acceleration	.81	.96	.99	.86	.99	.88	.83	.73	.58	.62	.46	.77	.56	.99	.91	.78	.71	.88	.91	.74	.95
	.71	.85	.72	.81	.99	.68	.86	.51	.55	.46	.42	.65	.40	.99	.90	.80	.78	.88	.54	.46	.95
Average EER																					
open hood																					
close hood																					
open trunk																					
check trunk																					
close trunk																					
fuel lid																					
open left door																					
close left door																					
open right door																					
close right door																					
open two doors																					
close two doors																					
mirror																					
check trunk gaps																					
lock check left																					
lock check right																					
check hood gaps																					
open swl																					
close swl																					
writing																					

Table 6.1: Top group rows show EER for activity recognition without using location. The bottom group incorporates location. For each case we use body-model, signal-oriented or their combination either based on IMU-sensor data or on acceleration

BM_IMU derived features with signal-oriented features for the IMU sensors. On average,

BM_IMU performs better on body-models with an EER of 0.92 than signal-based features with 0.88. Only for four activities, the signal-oriented approach is marginally better. The best result is achieved by combining both approaches which improves the average EER to 0.93. In combination 18 activities are recognized better than using signal-oriented features only. Solely for one activity (*check trunk gaps*), the signal-oriented features perform marginally better.

Row four to six in Table 6.1 show results using the BM_ACC and signal-based features on acceleration sensors only. An EER of 0.57 is obtained using signal oriented features. The BM_ACC outperforms the former with 0.61. Again the combination of the two types of features performs best with an EER of 0.64. For five out of twenty activities, the signal-based approach is slightly better.

From the results we can conclude that using body-models does indeed improve results by about 4% in both cases with respect to signal-oriented features alone. Combining sensor-oriented features with the body-model features further increases performance. A larger difference in performance however is observed between the precise IMUs and the acceleration only approaches, where the performance drops substantially (from 0.93 to 0.64, using the combination of signal-based features and the body-model).

Benefit of Location

In a second setting we incorporate location information as described in Section 5.3.3. The results are given in rows seven to twelve of Table 6.1. On average, the BM_IMU performs better than the signal-based approach on the IMU with an EER of 0.92 respectively 0.90. Only for three activities, the signal-oriented approach is marginally better. Combining the two feature types performs similar to the body-model only approach with an average EER of 0.92. Only for two activities, the signal-oriented features perform slightly better.

For the acceleration-based approach, we obtain an EER of 0.71 using signal-oriented features. The approach using BM_ACC again outperforms the signal-oriented approach with 0.81. A combination of both yields no significant improvement in average remaining at an EER of 0.81. For three out of twenty activities, the signal-based approach is slightly better.

For the IMU-based approach using location information does not have a significant effect. In case of acceleration sensors however integrating location information helps to improve the results narrowing the difference of the EER to the IMU-based approach to about 10%.

Reducing Number of Sensors

In a third step we reduced the number of sensors from 5 to 2 sensors worn at the left and right wrist. Using the IMU approach the recognition drops about 7% from 0.93 (5 sensors) to 0.86 (2 sensors). All activities are better recognized with 5 sensors. The largest drop in performance of about 30% can be observed for the activities *opening/closing* 2

location with w/o	BM_IMU + signal, inertial	.86	.99	.99	.88	.95	.90	.92	.75	.85	.88	.81	.75	.51	.96	.94	.89	.86	.86	.82	.81	.94
	signal, inertial	.82	.97	.95	.88	.92	.88	.86	.68	.78	.77	.79	.67	.35	.95	.99	.86	.89	.63	.90	.71	.94
	signal, acceleration	.43	.28	.44	.51	.79	.45	.67	.23	.32	.35	.00	.35	.03	.90	.54	.70	.73	.35	.00	.09	.91
Average EER	open hood																					
	close hood																					
	open trunk																					
	check trunk																					
	close trunk																					
	fuel lid																					
	open left door																					
	close left door																					
	open right door																					
	close right door																					
	open two doors																					
	close two doors																					
	mirror																					
	check trunk gaps																					
	lock check left																					
	lock check right																					
	check hood gaps																					
	open swl																					
	close swl																					
	writing																					

Table 6.2: EER for activity recognition with 2 sensors (both IMUs and acceleration sensors) with and without location

doors. The reason behind this are magnetic disturbances caused by the moving doors, which heavily influence the orientation estimation of IMUs.

Reducing the number of acceleration sensors, the performance also drops significantly from 0.81 (5 sensors) to 0.61 (2 sensors). Whereas the IMU only loses 7%, the decrease using acceleration sensors is about 20%. Using two IMUs without location information achieves 0.86 EER whereas using two acceleration sensors achieves only 0.61 EER together with location.

Joint Boosting vs. Hidden Markov Models

HMM	leftright, on primitives	.26	.74	.86	.45	.00	.41	.00	.00	.03	.25	.12	.12	.00	.40	.82	.00	.00	.00	.71	.29	.01
	fullyconnected, continous data	.59	.61	.44	.54	.90	.41	.67	.35	.72	.66	.73	.70	.32	.59	.66	.58	.65	.44	.61	.54	.72
	leftright, continous data	.56	.48	.50	.49	.91	.31	.71	.36	.62	.47	.76	.42	.19	.78	.83	.52	.70	.30	.78	.43	.70
JointBoosting	Average EER	.92	.99	.99	.94	.100	.94	.94	.86	.90	.92	.81	.90	.81	.97	.99	.90	.83	.95	.96	.91	.94
	open hood																					
	close hood																					
	open trunk																					
	check trunk																					
	close trunk																					
	fuel lid																					
	open left door																					
	close left door																					
	open right door																					
	close right door																					
	open two doors																					
	close two doors																					
	mirror																					
	check trunk gaps																					
	lock check left																					
	lock check right																					
	check hood gaps																					
	open swl																					
	close swl																					
	writing																					

Table 6.3: EER for different types of HMMs vs. Joint Boosting

In addition to our experiments using joint boosting as discriminant classifier, we also evaluated a generative approach using HMMs. We experimented with different topologies of HMMs, more specifically a fully-connected topology (transitions are allowed from any state to any state) and a left-right topology (the transitions are constrained to one direction only). We tested both types with 2 different feature sets. First, we provide as input the sequence of primitives, which are also used by joint boosting. As HMMs have the property to capture continuous time series well, we also evaluated directly on the continuous values. The results are given in Figure 6.3. It can be seen that the discriminant approach outperforms the generative approach using HMMs for all settings by a significant margin.

Discussion

In this section the effects on the recognition performance regarding different aspects, namely the *Benefit of Body-Model*, *Comparison between IMUs and Acceleration*, the *Benefit of Location* and the *Reduction of Number of Sensors* are reported. Incorporating a body-model improves consistently the results for all settings. As the analyzed activities are still relatively simple we expect that for more complex activities the margin between

signal based approaches and body-model based approaches will become even more pronounced.

Replacing IMUs with acceleration sensors always results in a significant drop in performance. This is an interesting result in itself as most systems using body-worn sensors rely on accelerometer data only. By incorporating additional information (such as location and combining signal and body-model based features) in the best case the performance difference between using IMU-sensors versus only accelerometer sensors is still 10%. For activities such as *writing*, *check trunk*, *mirror* we obtain constantly good recognition rates for the BMU_ACC.

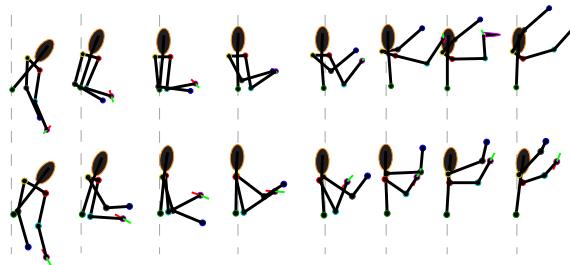


Figure 6.3: The acceleration-based sequence (top) approximates the IMU-based sequence (bottom) very well for the opening hood activity

On rotation changes perpendicular to earth gravity which can be found in activities like *opening hood*, *closing hood*, *opening trunk*, *closing trunk* the acceleration based approach works expectedly well as the orientation towards ground can be estimated accurately using accelerometers. Figure 6.3 shows a sequence of snapshots of an *opening hood* activity. The user lowers himself to grab the hood and lifts it above his/her head. It shows that the BM_ACC (top) approximates the BM_IMU (bottom) visually well for this activity.



Figure 6.4: Writing. (Left) The wrong acceleration based body configuration. (Right) IMU-based body configuration.

As we limit the lower arm to rotate to the front, we may obtain wrong horizontal positions for the hands, like illustrated in Figure 6.4 – the left side illustrates *writing* using the acceleration based approach, the right side the more accurate IMU-based approach. Though the lack of full orientation information using acceleration leads to wrong postures, it might not hurt the classification task, as long as the posture and its features stay consistent within the same activity and distinctive enough between different activities. The good results of recognizing *writing* support this assumption.

Without location information the acceleration based approach worsens significantly. Similar activities, for example *open/close hood*, *open/close trunk* get confused, as these

are similar in motion and thereby distinguishable only by location. The IMU-based approach does not profit from location. As IMUs yield a global unique orientation and hereby encode the orientation of the wearer with respect to the car, it contains enough information to distinguish for instance *opening hood* or *opening trunk*. However as the sensor fusion of IMUs includes magnetometers BM_IMU suffers from magnetic disturbances found in activities like *closing/opening the door* and *checking the locks*. This influence is more intense using the wrist sensors only.

Obviously, there is a correspondence between the user's orientation with respect to the car and his/her absolute location in this specific scenario. This fact raises the question if the improved performance using IMUs instead of acceleration sensors only results from the global orientation given by the IMU sensors. To this end, we evaluated the setting combining BM_IMU and signal-oriented features for five and two sensors with a restricted feature set without global features. Table 6.4 shows the results applying five and two sensors.

5 senors	.81	.86	.86	.81	.87	.77	.85	.71	.81	.79	.79	.83	.72	.87	.86	.77	.79	.76	.85	.79	.84
2 sensors	.70	.82	.72	.77	.87	.71	.85	.63	.66	.58	.60	.60	.33	.83	.81	.77	.79	.54	.54	.58	.92
Average EER																					
open hood																					
close hood																					
open trunk																					
check trunk																					
close trunk																					
fuel lid																					
open left door																					
close left door																					
open right door																					
close right door																					
open two doors																					
close two doors																					
mirror																					
check trunk gaps																					
lock check left																					
lock check right																					
check hood gaps																					
open swl																					
close swl																					
writing																					

Table 6.4: EER for activity recognition with 2 and 5 sensors (BM_IMU + signal) without global features

Discarding all global features of the BM_IMU approach combined with signal-based features with five IMUs obtain an average EER of 0.81. In fact, the results are slightly worse compared with the global approach (0.93). They still outperform the approach combining BM_ACC with signal-oriented features on acceleration data only (average EER of 0.64). Regarding two sensors, the performance drops from 0.86 to 0.69 (0.43 for acceleration sensors only). From the results we can conclude that using IMUs instead of acceleration sensors only, we can still improve the results significantly without considering global features.

By exploiting the feature selection property of joint boosting, a combination of all features (BM_ACC, BM_IMU and signal-based) achieves a minor improvement to an average EER of 0.94.

6.6 Conclusion

This chapter provides a systematic analysis regarding different activity representations (model-based vs. signal oriented) and sensor settings such as type and number of sensors. It is shown that model-based approaches enable more robust activity recognition than signal-oriented approaches. While the improvement can be observed in all considered

settings based on body-models, hidden Markov models do not improve the results. Interestingly, promising results can be obtained using two wrist-worn IMUs only without any additional information. Whereas additional location turns out to be important information for activity recognition with low-cost and power-efficient acceleration sensors, the benefit of location for the IMU based approach is limited. As shown, different sensor requirements do not always lead to a high performance difference. Depending on the activities and the scenario, a prior effort to choose a feasible sensor setting is crucial for successful activity recognition.

The analysis presented in this chapter basically addresses the challenges *High variance in performance and user independence* (Section 1.1.1) and Usability (Section 1.1.2), in particular the impact of different sensor settings on the recognition performance. The following chapter proposes a novel method for multi-level activity recognition. Temporal constraints encoded in UML diagrams enable reliable recognition of composed activities or high-level tasks without requiring large amounts of training data. In addition, the recognition of low-level activities as considered in this and previous chapters can significantly be improved.

7

Hierarchical Activity Recognition Using UML Diagrams

The previous chapters have reported promising results for recognizing ‘atomic’ activities (such as drilling, handshaking, or open a door). However, in many applications the analysis and recognition of high-level and longer-term activities is an important component. This chapter proposes a novel method for multi-level activity recognition. Temporal constraints, for example encoded in UML diagrams, enable reliable recognition of composed activities or high-level tasks without requiring large amounts of training data.

7.1 Introduction

In the presence of sufficient training data, activity recognition methods based on state-of-the-art machine learning techniques have proven to be appropriate to train classifiers for ‘atomic’ activities. In this chapter we argue that the proposed methods for recognizing ‘atomic’ activities (see previous chapters) do not scale well to the recognition of high-level tasks that are composed of multiple activities. The main reason is that a prohibitive amount of training would be required to cover the high variability and the large number of possibilities to execute high-level tasks.

This chapter starts with the observation that for many relevant activities and high-level tasks a task-description already exists or is relatively easy to obtain from domain experts (for example in scenarios such as maintenance or cooking). These task descriptions or task models might come in different forms such as manuals, recipes, or UML diagrams. What is common to these descriptions is that they detail the workflow of sub-activities that constitute the high-level task. In order to enable the recognition of such high-level tasks this chapter proposes an approach that uses such task-descriptions effectively as prior knowledge, thereby reducing the required amount of training data significantly.

The main contribution of this chapter is therefore a novel hierarchical approach for activity recognition that allows to leverage prior knowledge contained in task models such as UML diagrams. The lowest level of our hierarchy is concerned with the recognition

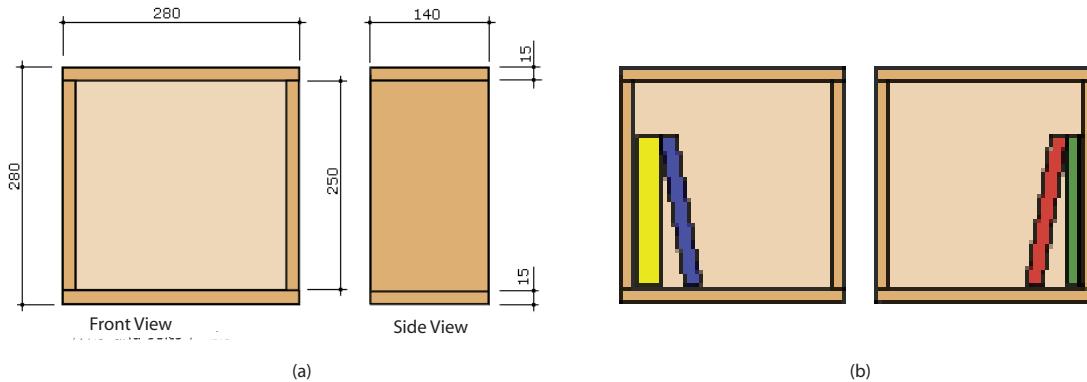


Figure 7.1: Illustration of boxes as they are included in the manual that can be found on the web. (a) A box displayed from front and side view highlights the final dimensions as well as the planks' adjustment. (b) The layout example of two boxes gives an impression how several boxes can be combined and used to store books.

of ‘atomic’ activities (called level-1 activities in the following). This level uses state-of-the-art machine learning to enable user-independent recognition of a wide variety of activities that are typically shared by several ‘composite’ activities. The second level then enables the recognition of activities that are composed of multiple level-1 activities. The focus of this level is on composite activities (called level-2 activities in this chapter) that have a clear temporal order and that are typically executed contiguously. The third level then uses the above mentioned prior knowledge extracted from task models to enable the recognition of complete tasks. At this level partial ordering constraints are modeled therefore allowing a high variability of the temporal ordering of the constituent composite activities.

Besides the above mentioned main contribution, this chapter also contributes a new and challenging data set for high-level task recognition. In this data set the recognition of level-1 (=‘atomic’) activities is not sufficient as these are both ambiguous and shared among various level-2 (=‘composite’) activities. The chapter also contributes an experimental analysis on this data set and demonstrates the ability of the proposed approach to recognize highly variable activities even based on a rather small amounts of training data.

The chapter is structured as follows. Section 7.2 motivates the proposed data set and discusses the challenges involved in high-level task recognition. The level-1 activities are recognized based on our approach applying joint boosting on body-model derived primitives as introduced in Chapter 5. Section 7.3 describes our hierarchical approach to recognize activities including three different levels. Section 7.4 summarizes the experimental results on our data set and Section 7.5 concludes and discusses the contributions of the chapter.

7.2 Data Set

As motivated in the introduction of this chapter, we want to advance the state-of-the-art in activity recognition by considering composite activities and high-level tasks. This section therefore introduces a new data set that is realistic in the sense that it contains real-world challenges such as the high variability in executing high-level tasks and the ambiguity of level-1 activities. This data set is used later to evaluate the effectiveness of our novel hierarchical approach for activity recognition.

To support our argument that current techniques do not scale well to the recognition of realistic and variable tasks, we followed a top-down procedure to design our data recordings in the following way. First, we started from a real high-level task namely the construction of two wooden book boxes as shown in Figure 7.1. For this task, necessary work steps (or activities) are extracted from an existing manual on how to build these book boxes. As it is usual for such a task description it gives a subject a large number of possibilities to execute the overall task as well as the different activities. This extraction of the task description from the manual results in a realistic and challenging set of level-1 activities which in general cannot be recognized perfectly. The task description also defines several level-2 activities composed of various level-1 activities but with a clear temporal order.

Finally, we asked ten independent people to perform the overall task of building two book boxes.

Task description based on a manual/UML diagram As previously mentioned, we propose to use high-level knowledge contained in UML-diagrams to improve the recognition of level-2 and level-1 activities. To support our argument that current techniques do not scale to the recognition of variable tasks, we prompt users to build two book boxes as illustrated in Figure 7.1(a). The visualization of the book box from a front and side view as well as the UML diagram [UML 2009] in Figure 7.2(a) are extracted from a manual as found on the web. Whereas building one box would result in a single sequential sequence of level-2 activities, building two boxes already gives a subject a large number of possibilities to execute the activities. As can be seen in the UML-diagram, the user starts by creating a template. The subsequent building of the two boxes can be performed concurrently. Reasonable orders can be extracted from the UML diagram. In order to cover the large variability of task execution and to make the data set more interesting and challenging, the coarse grained temporal order of composite level-2 activities was specified and given to the different subjects in advance. The fine-grained temporal order was neither discussed nor influenced. Figure 7.2(b) shows the different paths of ten subjects covered in this data set. Whereas subject number one first builds and hangs-up one box before proceeding to the second one, user eight parallelizes the necessary working steps. Note that seven different sequences can be noticed in the data for level-2 activities. Obviously, for a reasonable training of the high level model, a lot of training data would be

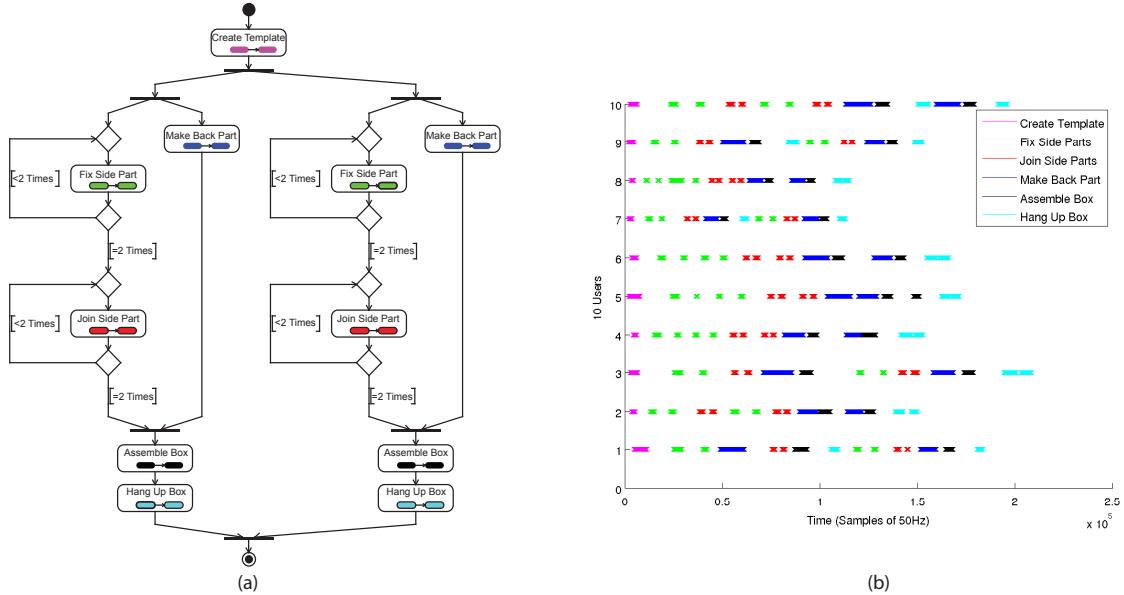


Figure 7.2: (a) UML diagram illustrating possible paths when building 2 book boxes. This diagram can be extracted from existing manual descriptions of the specific high-level task. (b) User dependent paths in the data set when building 2 book boxes. Seven different sequences can be noticed in the data for level-2 activities.

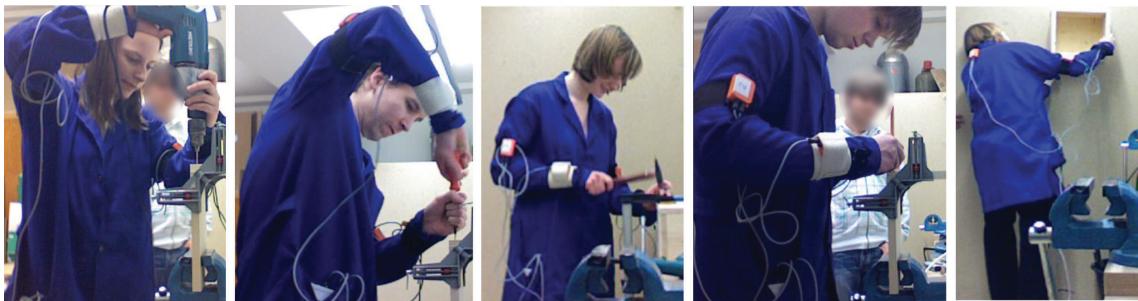


Figure 7.3: Diverse class complexity: drilling, screwing, hammering, mark, hang up.

required. This fact supports our claim that considering information in UML diagrams is an important step toward activity recognition in realistic scenarios.

Challenges in recognizing level-1 activities Beyond giving a subject a large number of possibilities to execute the overall task and the necessary activities, we'd like to emphasize some difficulties involved in recognizing level-1 activities. The right column of Table 7.1 lists all level-1 activities in this data set. In the following challenges in recognizing different level-1 activities are motivated. In many scenarios, selected activities differ significantly in their constitution. Often, activities are characterized by repetitive movements like turning the arm when screwing or moving the arm up and down while



Figure 7.4: Intra-class variability when performing activity screw (fix side part). Although the subjects perform the same activity, a high variance in execution can be observed.

hammering. Beyond these activities of longer duration (>10 seconds), very short activities (<3 seconds) like drilling, marking holes or hanging up boxes are of interest. Not only the short duration complicates the detection of activities. In addition, short activities often do not contain discriminant arm movements. Whereas hammering or turning screws can be identified by noticeable arm movements, the arm position hardly changes for activities like cutting the paper template or marking holes for drilling. The analyzed data set of this chapter includes activities of diverse complexity as illustrated in Figure 7.3. In addition to repetitive activities like sawing, hammering or screwing, a recognition of short activities like drilling, fixing a back support, marking, cutting or hanging up the boxes are of major importance.

Beyond the difficulty to find discriminant characteristics of level-1 activities, the execution often differs between subjects considerably. Figure 7.4 illustrates four different people while *screwing side parts*. Although the subjects perform the same activity, a high variance in execution (intra class variability) can be observed. A rotation of the screw driver can be enforced either by hand or turning the whole arm (see subject in the left images). Whereas the subject in the third picture uses his left hand, the last subject clasps the screw driver in a different way than the three other subjects.

In realistic scenarios different activities are often ambiguous due to their similarity to each other. In the maintenance area, activities like hammering, drilling and screwing occur frequently. Here, the activities' execution only differs slightly when performed in diverse working steps. Figure 7.5 illustrates drilling in two different working steps (subject one/two in setting one, subject three/four in setting two). The arm postures, as well as the movements while drilling, are similar for different classes (inter class similarity).

Since the recognition of level-1 activities is the input for level-2 activity recognition it is clear that we need to use a state-of-the-art method to obtain best performance. However, due to the above mentioned difficulties such as intra class variability and inter class similarity, it is unrealistic to expect perfect recognition of level-1 activities. As we will see in the later section, our three-level hierarchy can successfully deal with this problem:



Figure 7.5: Inter-class similarity while drilling. 2 right pictures (drilling to fix side parts), 2 left pictures (drilling to join side parts). The arm postures, as well as the movements while drilling in different work steps are similar.

Composite Activity	Sub Activities
Fix Side Parts	Mark Holes, Drill, Screw (Fix Side Parts)
Join Side Parts	Mark Holes, Drill, Screw (Join Side Parts)
Make Back Part	Sawing, Drill, Screw, Fix Back Support
Assemble Box	Mark Positions, Hammering
Hang Up Box	Mark Holes, Drill, Screw, Hang-Up Box
Create Template	Mark Template, Mark Holes, Cut Template

Table 7.1: 6 level-2 and the corresponding 19 level-1 activities necessary to build 2 book boxes.

While recognition of level-1 activities is not perfect using only the lowest level of the hierarchy, the higher levels of the hierarchy enable robust recognition of almost all level-1 activities.

Definition of level-2 activities composed of level-1 activities In addition to level-1 activity recognition, a main focus of this chapter is to recognize composite activities. Table 7.1 summarizes the six level-2 activities defined by the overall task as well as the related sequential level-1 activities as given in the UML diagram. In our model, we assume that level-1 activities belonging to the same level-2 activity have a clear temporal order. They are typically completed contiguously. While it might be theoretically possible to switch even within these sub-activities, their execution is unnatural and inefficient because box components have to be recurrently fixed and unfixed with the vise and the mitre square (see Figure 7.4).

From Table 7.1, one can observe at first glance that some composite activities are highly similar. This fact makes the recognition challenging also on the second level. Both for *fix side parts* as well as for *join side parts*, hole positions have to be marked first. After drilling the holes, the screws can be used to fix wooden parts together. Whereas the components are fixed using the vise when fixing the side parts, a vise cannot be used for the level-2 activity *join side parts* (see Figure 7.5). Here, recognizing drilling only at at

level one is not sufficient. A good detection system should be capable of distinguishing similar activities (for example drilling or screwing) embedded in level-2 activities such as *fix or join side parts*.

To complete the entire workflow, *join* and *fix side parts* have to be executed four times while building two boxes. Whereas *create template* with sub-activities *mark template*, *mark holes* and *cut template* is performed only once in the beginning, *hang up box*, *assemble box* and *make back part* are performed twice (see also Figure 7.2 (a)).

Independent subjects performing the high level task To show the effectiveness of our approach, we recorded ten persons while building two book boxes each. In a first step, the subjects create a template to ease and accelerate the subsequent marking and drilling. Next, the boxes have to be built. The study ends when both boxes are hanging on the wall. To create a representative data set, we selected ten independent subjects without a deeper knowledge in activity recognition. Seven male and 3 female subjects in age between 23 and 37 participated in the data recording. The users' height varied between 1.63m and 1.85m. Furthermore, none of the users has advanced experience using workshop tools or working with wood.

7.2.1 Sensor setup and annotation

The subjects were provided with five inertial measurement units as introduced in Section 3.2.2. The five sensors were mounted at the users' lower and upper arms as well as the torso. To reduce the risk of data loss, the sensors were directly connected to a laptop using an XBus system [XSens 2009]. In total, our data set consists of about 10 hours of recording while building 20 book boxes. In this recording, the sensors deliver data at a rate of 50Hz.

To analyze the effectiveness of our approach, we aimed for fine-grained annotations. While performing the activities, the subjects are recorded on video for later annotation. Both level-1 as well as the composite level-2 activities are annotated using the video recordings.

7.3 Hierarchical Activity Recognition using UML

This section describes our method to classify activities using prior knowledge extracted from UML diagrams. The method uses a three level hierarchy. For level-1 activity recognition, we apply our approach using joint boosting on body-model derived primitives to train a discriminant classifier as introduced in Chapter 5. As motivated before, additional temporal information is often encoded in manuals or UML diagrams. Considering the work-flow and temporal order of sub-activities, we then describe how to recognize composite level-2 activities based on the output of level-1 recognizers. Finally, on level 3,

additional partial ordering constraints derived from UML diagrams are used to enable the recognition of entire tasks composed of multiple and repeating level-2 activities. Figure 7.6 schematically illustrates our approach on three activity levels.

7.3.1 Level-1: Multi-activity recognition using joint boosting

The first level of the hierarchy is concerned with user-independent recognition of level-1 activities. Similar to Chapter 5, we calculate features on segments that can subsequently be used as input for the training and the test phase of joint boosting. Here, we once more split the annotation of activities with long duration into multiple segments of maximum length of four seconds. As a result, the overall number of segments that need to be classified drops significantly.

Joint boosting gives a classifier score for a specific activity at each time instance. For activity spotting, we are interested in locating maxima in the streams of scores. Therefore we apply the mean shift algorithm with an activity specific window length. Here, the window length is chosen to be the mean duration of each activity on the training data. In the following $a_i, i \in 1, \dots, l_1$ indicates a maxima for activity i (l_1 is the number of level-1 activities). We are fitting a sigmoid function to the scoring values to obtain probabilities $p(a_i)$ for all activities.

7.3.2 Level-2: Sequential constraints given by UML diagram

The second level of the hierarchy calculates probabilities of composite activities $c_j, j \in 1, \dots, l_2$. For example, *fix side parts* is composed of three sub-activities *mark holes (side parts)*, *drill holes (side parts)*, and *screw (side parts)*. The individual probabilities of these sub-activities as well as the temporal relation among each pair is considered when calculating the probability for a composite activity. Let $d(a_{i_1}, a_{i_2})$ map two maxima of level-1 activities to their temporal distance. A Gaussian model is learned for the distance of each pair of sub-activities of c_j on the training data. In the following $s_{j,k}$ is the k -th sub-activity of c_j and m_j denotes the total number of sub-activities of c_j . Note that some level-1 activity a_i might occur multiple times within the same sequence of sub-activities of c_j . For all tuples $c_j = (s_{j,1}, \dots, s_{j,m_j})$, a probability for the composite activities c_j can be calculated as follows:

$$p(c_j) = \frac{1}{Z} \prod_{k=1}^{m_j} p(s_{j,k}) \prod_{\substack{(i_1, i_2), i_1 < i_2, \\ i_1, i_2 \in \{1, \dots, m_j\}}} p(d(s_{j,i_1}, s_{j,i_2})) \quad (7.1)$$

Note that in order to calculate the normalization factor Z we would need a background model that represents all other sequences of sub-activities not corresponding to any of our composite activities c_j . As learning such a background model would require significant

amounts of training data we opt to implicitly approximate the background model by fitting a sigmoid to obtain probabilities $p(c_j)$. As we will see in the evaluation, this approximation enables recognition of most ‘composite’ activities.

7.3.3 Level-3: Activity recognition considering temporal constraints of UML diagrams

On level 3 of our hierarchical approach, we aim to recognize high-level tasks that may be composed of multiple and repeated level-2 activities c_j . Let w_n be the total number of level-2 activities that make up a high-level task A_n . For each tuple $(c_{t_1}, \dots, c_{t_{w_n}})$, a function $p(c_{t_1}, \dots, c_{t_{w_n}}) \rightarrow \{0, 1\}$ considers temporal constraints of the UML diagram and maps the corresponding tuple selection to a probability of 1 or 0. Other mapping functions are possible, for example when some ordering is more likely or more appropriate than other orderings the probability would be mapped onto a number between 0 and 1. As UML diagrams only give partial ordering information we cannot weight the different options using those diagrams alone. One could imagine consulting domain experts to give such relative weightings. The total probability $p(A_n)$ for each tuple $(c_{t_1}, \dots, c_{t_{w_n}})$ is calculated as follows:

$$p(A_n(c_{t_1}, \dots, c_{t_{w_n}})) = p(c_{t_1}, \dots, c_{t_{w_n}}) \prod_{i=1}^{w_n} p(c_{t_i}) \quad (7.2)$$

At this level, we are coding the order and time of level-2 activities while performing the overall process. One can imagine that there are many possible mappings of hypothesized level-2 activities to the complete task model. In the presented scenario, the function $p(c_{t_1}, \dots, c_{t_{w_n}})$ can be used to restrict the search space dramatically. Activities that do not meet the constraints given by the task model can immediately be discarded.

7.4 Experiments and Results

As discussed before, the main contribution of this chapter is to enable the recognition of high-level tasks based on a hierarchical approach. The following section first describes the evaluation procedure. The subsequent presentation of the results is structured in two parts. The first (results for level-1 activities) compares the results of the 19 level-1 activities for each of the three hierarchy levels as discussed in Section 7.3. The second part then compares recognition of composite level-2 activities as obtained from the second and third level of the hierarchy.

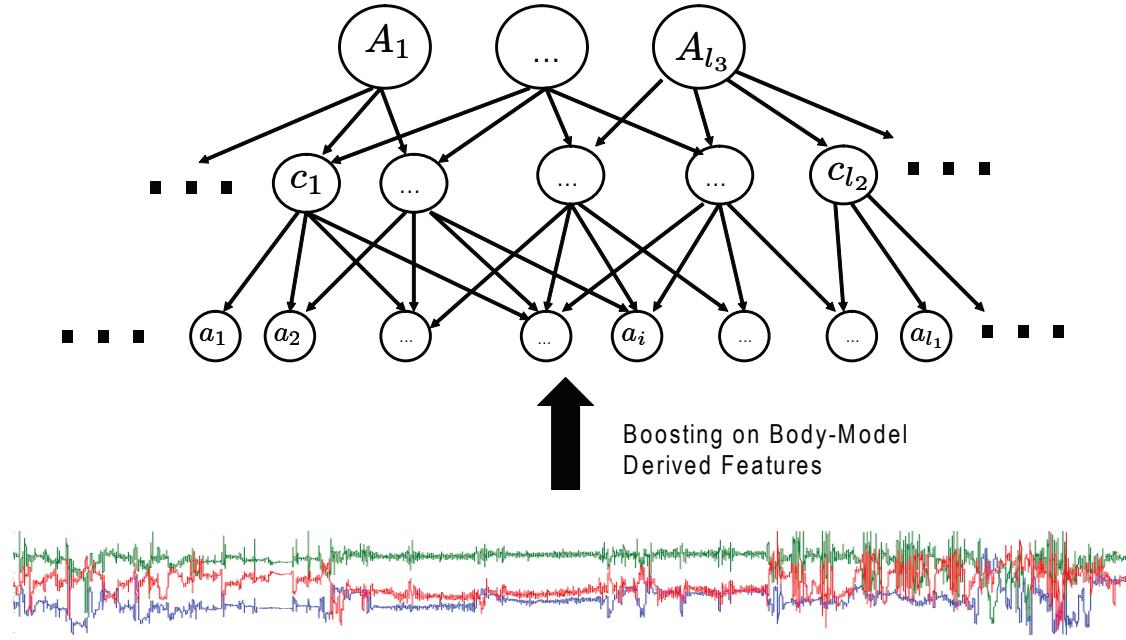


Figure 7.6: Schematics of the approach on three activity levels

7.4.1 Evaluation procedure

The approach presented in the previous sections is evaluated on the data set introduced before. In this data set ten different people are building two book boxes each. For each activity, we report precision and recall of each activity individually with respect to the collected background data and the remaining activities. For each activity we vary an activity-specific threshold defining the borderline for acceptance. We perform *leave-one-user-out* cross validation to evaluate user-independent activity recognition. In each cross-validation round, the probabilities for all segments is calculated. A segment S is counted as true positive if the annotated segment A has the same activity label and if the following equation holds true:

$$\text{start}(A) \leq \text{center}(S) \leq \text{stop}(A) \quad (7.3)$$

Note that $\text{start}(A)$ and $\text{stop}(A)$ comply with the begin and end times of the ground truth segment A and $\text{center}(S)$ specifies the central time of segment S . The equation ensures that the analyzed activities are spotted at the right time position. Only if the central time of a segment intersects with the annotation, the specific segment is counted as a true positive. Typically, the precision decreases when increasing the recall for a particular activity. Ideally, both precision and recall are 100%. The evaluation is performed both on the 19 level-1 activities and the six composite level-2 activities.

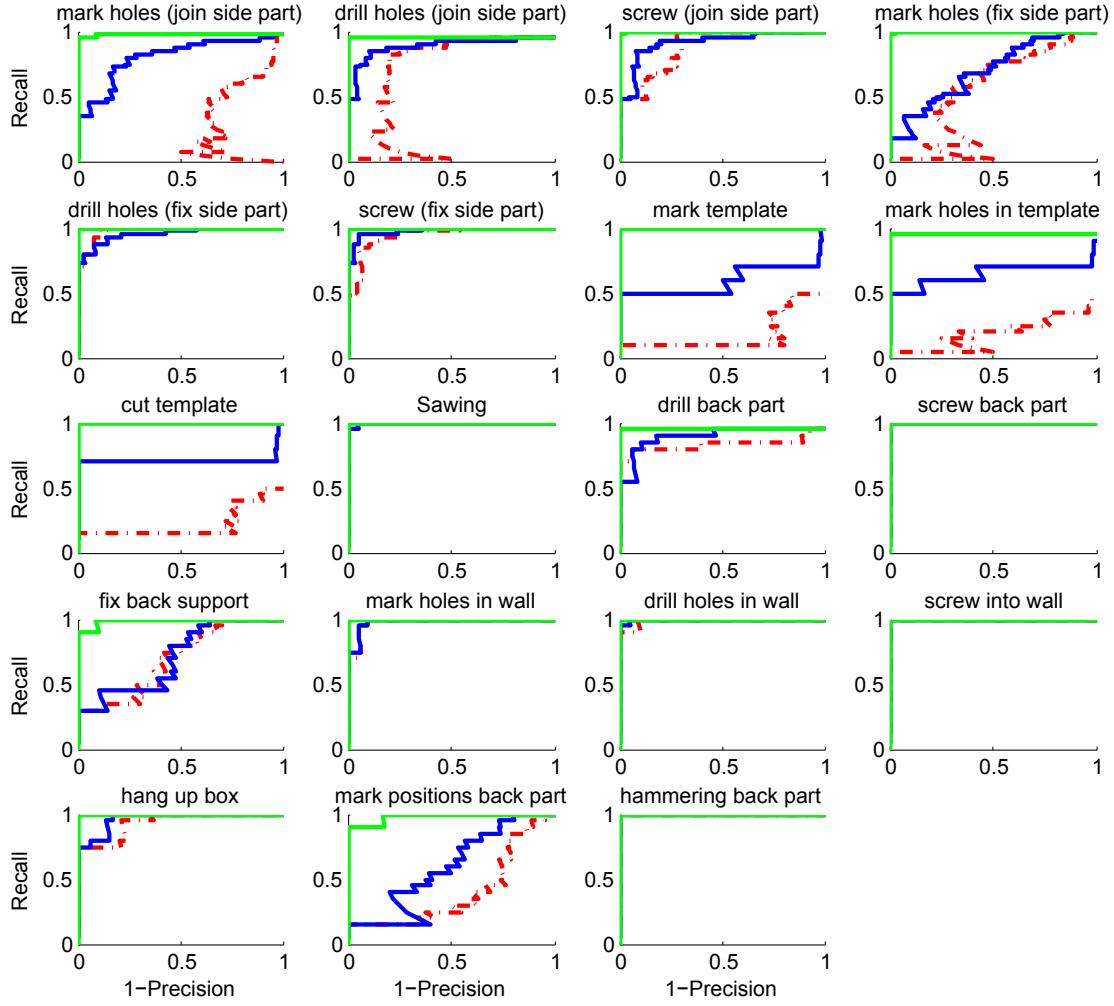


Figure 7.7: Precision-recall of the 19 level-1 activities. Level-1 recognition (red). Recognition as inferred by level-2 recognition (blue). Recognition as inferred from level-3 processing (green).

7.4.2 Results for level-1 activities

The following analysis compares recognition results as obtained by the three levels of our hierarchical approach. Whereas level one applies joint boosting, level two considers the temporal relations of sequential level-1 activities. Finally, the evaluation on level three also includes partial ordering constraints among the composite level-2 activities.

Results of level-1 recognizers The red curves in Figure 7.7 illustrate how the joint boosting approach performs on the 19 level-1 activities in the proposed data set. Four out of 19 activities (*hammering back part*, *screw into wall*, *screw back part* and *sawing*) are perfectly recognized with a precision and recall of 1. These four activities are characterized by repetitive movements over a longer period of time. Also, the subject's posture

while screwing into the wall is discriminant. The next best activities still perform well with a precision and recall higher than 0.8: *hang up box* and *drill holes in wall* and *drill holes (fix side parts)*. As before, those activities are mainly characterized by repetitive movements for screwing and scarce posture changes: while hanging up the box, the subject keeps two arms up; while drilling, a specific position of both hands is required to operate the drilling machine. The activities *screw* and *drill (join side part)* and *drill back part* obtain lower precision and recall. A high recall larger than 0.9 is not achieved caused by a large number of false positives. As motivated before in Section 7.1, a high similarity of different kinds of drilling activities as well as a high variance in the postures causes the performance drop for those activities. As one expects from purely level-1 recognition, a significant number of activities are not well recognized. Here, the following seven out of 19 activities are not recognized well: *Mark holes (fix and join side parts)*, *mark template*, *mark holes in template*, *cut template*, *fix back support* and *mark positions back part*. All those activities are very short and almost no arm movements are involved when performing the activities. Furthermore, executing those activities allows for a high variance, for example when cutting the template.

Inference of level-1 activities from level-2 recognizers In a second setting we incorporate temporal relation of sequential level-1 activities. The blue precision/recall curves in Figure 7.7 clearly show that the described level-2 recognition significantly improves results. Whereas the activities with good recognition results on level one still perform well, one can observe an improvement for short activities like *mark holes (fix side part)*, *mark the template*, *mark holes in template* and *cut template*. Those activities benefit both from the sequential ordering constraints and from a confident recognition of other sub-activities that are part of the same composite activity. In most cases, a combined probability by sequential activities improves the results. Only for activity *sawing*, the results on level two are slightly worse than on level one. The high and discriminant probability of activity *sawing* on level one is slightly decreased by a lower combined probability of the corresponding composite level-2 activity.

Inference of level-1 activities from level-3 recognition The third level considers all temporal constraints given by the UML diagram. The green curves in Figure 7.7 illustrate the superior performance compared to level one and two. For 13 out of 19 activities, the optimal recall and precision of 100% is achieved. For activities *mark holes (join side part)*, *fix back support* and *mark positions back part*, one or two false positives are detected before approaching a recall of 100%. Only for activities *drill holes (join side part)*, *mark holes in template* and *drill back part*, one instance is never detected. The specific segments are discarded caused by a low probability.

While the recognition of level-1 activities is indeed difficult with a state-of-the-art recognition method, the proposed hierarchical approach including prior knowledge extracted from UML diagrams significantly and effectively improves the recognition results.

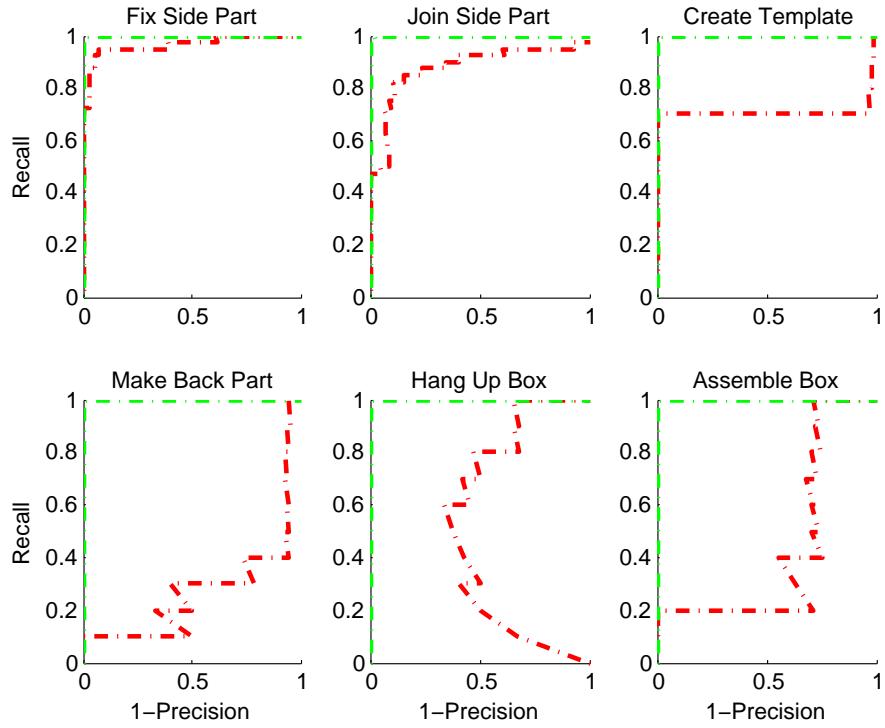


Figure 7.8: Precision-recall of 6 level-2 activities. Level 2 (red) models sequential sub-activities, level 3 (green) includes all temporal constraints.

7.4.3 Results for composite level-2 activities

As motivated in the introduction of this chapter, there is a high interest in recognizing composite activities. This section reports the results for the recognition of the six composite level-2 activities as recognized by level two and as inferred from level three recognition.

Results of level-2 recognizers The red curves in Figure 7.8 illustrate the results when recognizing the introduced composite activities using level two of our hierarchy. Whereas a recognition of *fix side part* and *join side part* proves satisfactory, the results for *create template* is much lower. Crossing a recall of 0.75, the precision promptly drops to almost zero. For the remaining activities, many false positives are recognized. There are two reasons for this low performance. Adequate results for *hang up box* and *assemble box* are not feasible because of the bad performance of sub-activities *fix back support* and *mark positions back part*. For *make back part*, a modeling of the temporal relation among the sub-activities turns out to be difficult. As a result, the combined probability is drawn down. Although the results on level two are only partially successful, the result of this level is essential for the subsequent filtering on level three.

Inference of level-2 activities from level-3 recognition As for the 19 level-1 activities, the results for composite level-2 activities significantly improve when performing recognition on level three. Considering the partial and temporal constraints as given by the UML diagram, an optimal recall for a precision of 100% is observed for all composite level-2 activities. Although some instances of sub-activities will always be discarded (for example *drill holes (join side part)*), all composite activities are successfully detected. This is possible because a minimal probability is considered even for false negatives, therefore allowing the detection of composite level-2 activities.

From the above results we can conclude that not only level-1 activities are successfully recognized and disambiguated but that also composite and level-2 activities can be recognized robustly. This is quite remarkable as the novel data-set is challenging, the used training data is rather limited, the recognition has been done in a user-independent manner and not all sub-activities have been detected.

7.5 Conclusion

In this chapter we have introduced a novel hierarchical approach for activity recognition that allows leveraging prior knowledge contained in task models such as UML diagrams. Using a realistic and challenging data set of 10 subjects building two book boxes each, we showed that the approach leads to highly promising results both for level-1 and level-2 activity recognition. As expected, the application of a state-of-the-art method based on machine learning is not powerful enough to allow reliable recognition of all level-1 activities. The information modeled and used on the second and third level of our hierarchy based on prior knowledge encoded in task models and enables the recognition of the entire tasks without requiring prohibitive amounts of training data. As a result, propagating the knowledge of the overall tasks from the third to the second level improves the recognition of both level-1 and level-2 activities significantly.

In conclusion, we believe that the proposed approach is highly attractive to enable scalable activity recognition. Especially in the light of less training data, the integration of prior knowledge from task models is a promising tool to enable reliable recognition of composite activities and complex high-level tasks in a user-independent manner.

Given the state-of-the-art in activity recognition and given the involved challenges of our data set we did not expect such promising results. We therefore argue and strongly believe that hierarchical approaches like ours also modeling prior knowledge about task models are essential to further advance the state-of-the-art in activity recognition.

8

Conclusion and Outlook

This thesis has investigated one important aspect of context-aware computing, namely the recognition of activities with wearable sensors. In the following we summarize our main insights and contributions. We also give an outlook on essential future work.

8.1 Conclusions

Segmentation as a filter for subsequent classification One of the main challenges in activity recognition (see Section 1.1.1) is the extraction of relevant segments from potentially large amounts of background data. In this thesis, we have developed an efficient segmentation method using short but fixed poses of the hands and turning points in arm movements to find candidate start and end points of activities in a continuous data stream. Several experiments show that this novel segmentation procedure results in a significant data reduction. The filtered segments are easier to classify than the continuous data stream.

Body-model derived primitives allow for a robust low-level activity recognition The experiments in this thesis point out that human activities can differ in various ways. Our representation of motion and posture primitives on segments has proven effective in both user dependent and across-user activity recognition. Experiments on two data sets confirm the strength of the introduced features that are more robust against for example speed differences and the user's position to the object when performing an activity. The presented body-model based approaches show superior performance compared with signal-oriented approaches both using inertial measurement units and acceleration sensors only.

Leveraging prior knowledge in task models enables high-level activity recognition This thesis has taken the first step towards modeling and recognizing high-level activities from body-worn accelerometers. Our method for multi-level activity recognition considers temporal constraints encoded in UML diagrams thereby enabling reliable recognition

of high-level tasks without requiring large amounts of training data. We show the applicability of the approach to model and recognize highly variable activities and tasks across users with a realistic and challenging data set. Propagating the knowledge of the overall high-level tasks to lower levels improves the recognition of low-level activities significantly. In conclusion, we believe that the proposed approach is highly attractive to enable scalable activity recognition. Especially in light of a small amount of training data, the integration of prior knowledge from task models is a promising tool to enable reliable recognition of complex high-level tasks in a user-independent manner.

Successful activity recognition requires a selection of feasible sensor modalities Replacing inertial measurement units (IMUs) with acceleration sensors and reducing the number of sensors results in a significant drop in performance for the considered data sets in this thesis. Interestingly, location affects the recognition results of acceleration and IMU sensors differently in the car-quality control data set (Chapter 6). Whereas the acceleration based approach significantly profits from the used location, the IMU-based approach does not. In this specific scenario, there is a correspondence between the user's orientation (as given by the IMUs) with respect to the car and his/her absolute location in this specific scenario. Other scenarios might not feature this correspondence. In summary, several factors have an impact on the recognition results depending on the activities and the scenario. A prior effort to choose a feasible sensor modality is crucial for successful activity recognition.

Joint boosting enables automatic and efficient discovery of important and distinctive features To accomplish a good representation of activities that differ significantly in their constitution, the feature space in this thesis is high dimensional. Joint boosting enables the automatic discovery of important and distinctive features ranging from motion over posture to location. The method shares weak classifiers across multiple classes while the strong classifiers are learned jointly. Especially in the given experiments with about 20 activities, joint boosting reduces the computational complexity by finding common features that are shared across several classes. In Chapter 7, the approach is easily transferred to the wood shop data set with similar promising results.

Gesture Recognition Arm gestures are an interesting alternative to enabling the user to explicitly issue a limited set of commands and therefore enable interaction in a truly mobile settings. For many scenarios, less than ten gestures meets the users' expectations. Gestures must be distinguishable from gestures that occur in daily life and need to be socially acceptable, i.e., the gestures have to be unobtrusive, both for the user and for his/her environment. In this thesis we have shown that the complexity of gestures affects their distinctiveness from gestures in daily life. Gestures composed of three or more primitives can often be considered as suitable control gestures. The number of false detections for less complex gestures is too high. Although highly complex gestures will not be accepted

by users, users believe that gestures based on shapes, letters and other known forms reduce the training and concentration effort and therefore be accepted as control gestures.

Discriminant approach outperforms hidden Markov models In this thesis, we evaluate the feasibility of hidden Markov models to capture continuous time series on two different data sets. In Chapter 3, the first experiment demonstrates that left-right HMMs can model segmented activities in test sequences sufficiently. However we observe that hidden Markov models cannot easily be transferred from a pre-segmented to a continuous data stream. The experiment's results show that the models cannot discriminate the short activities from irrelevant data. In Chapter 6, we additionally experiment with different topologies of HMMs, more specifically a fully-connected topology (transitions are allowed from any one state to any other state) and a left-right topology (the transitions are constrained to one direction only). It can be seen that the discriminant approach outperforms the generative approach using HMMs for all evaluated modalities by a significant margin. In addition, the proposed discriminative approach is computationally more efficient than HMMs in the high dimensional feature space. Many dimensions will have noisy data also impeding the HMMs to converge. From the results we can conclude that HMMs cannot directly be applied to detect gestures and activities in continuous recognition tasks. A high modeling effort to tune different HMM parameters (for example number of states or Gaussians and input dimensions) or a combination of HMMs with boosting as proposed by [Lester *et al.* 2005] might lead to better results to some extent.

8.2 Outlook

Reference data sets for high-level recordings An important step toward the evaluation of high-level activity recognition, for which we lacked the time and resources in the course of this thesis, is to recognize different high-level tasks under realistic conditions. Especially in supervised settings, obtaining such data together with sufficiently detailed annotations is tedious and time-consuming. Cooperation with related research groups for publication of reference data sets seems to be essential. Reference data sets could help researchers compare the performances of their approaches. However, a high variety of applied sensors and the lack of standard evaluation procedures make the recording of a commonly used data set a challenging task.

Quality of performance In various domains, for example sports medicine, or the maintenance area, there is a high interest in not only knowing whether an activity has been performed, but also how well it has been performed. Automatically recognizing the quality of an activity is a challenging problem and an open research question which could enable a range of interesting applications. Such a computer system could allow coaches and sports medicine practitioners to quantitatively measure relevant aspects of player performance, or educate maintenance workers in performing specific work tasks. Reliable

activity detection, as addressed in this thesis, is not sufficient for this purpose. Additionally, future research has to approach the more complex problem of activity segmentation and definition of quality measurements based on continuous sensor data.

Usability issues of the hardware The success of activity recognition in general is still far from being satisfactory in realistic and challenging real-world scenarios. Especially in the area of wearable computing, researchers must consider trade-offs between a system's usability and the quality of recorded data. In particular, the sensors' size, number, placement, as well as the runtime strongly influence the users' subjective perception of a wearable system. It is crucial to focus on architecture and experience reports assisting the study of usability in wearable computer applications in future work.

Exploiting a recognition of high-level activities further We believe that the use of prior knowledge as proposed in Chapter 7 is very promising for the discovery and modeling of high-level activities, and has interesting properties that we have not exploited yet. Considering only one high-level activity as in the wood shop data set, an exhaustive search is still feasible. A step toward multiple high-level activities necessitates more sophisticated machine learning algorithms such as Markov Chain Monte Carlo methods [[Andrieu et al. 2003](#)].

Activity recognition entails privacy critical risks Knowledge about performed activities as well as the quality of performance entails privacy critical risks. In [[Zinnen et al. 2008](#)], we investigate the impact of activity recognition in an E-learning environment on user's privacy. The specific system stores information about the user's performed tasks. In consultation with several companies' departments of Work Council, Corporate Legal and Data Protection & Privacy, a privacy policy was specified to fulfill the seven principles of Notice, Purpose, Consent, Security, Disclosure, Access and Accountability as stated in the OECD's [[OECD 2009](#)] recommendation for protection of personal data. The main outcome of the study is that users always must be informed and explicitly agree on data collection. In addition, process and storage of data allowing inference on the user's competence is not allowed. Although the findings of this work are preliminary and based on a research project, they should raise awareness and provide some basic insights that need to be considered when developing realistic activity recognition systems in the future. A key factor for the acceptance is the tradeoff between the system's benefit for a user and the risk of misusing private data for employees' performance control.

List of Figures

2.1	List view (left) and document view (right) of the provided touchless document browser as applied in the ward round scenario. Actions like open/close or scroll up/down are triggered by keeping the mouse pointer on a specific area for a fixed period of time. By keeping the mouse pointer for a fixed time on those areas, the browser performs the corresponding action (scroll up/down, close).	22
2.2	Left: Activation of the gesture interface using face detection. When the doctor's line of sight intersects with the screen (square), the system is activated. Middle: Preliminary results of the active area of the face detection. Right: Wireless sensor with 5m range can be worn as a wristband.	23
3.1	Start and end postures surrounding the activity <i>turning on the heating</i> . Considering these postures, the activities can be separated into three segments. The later classification is performed on segments enclosed by postures.	26
3.2	Log-likelihood and smoothed log-likelihood for activity open hood over the continuous data stream. The left-right HMM is directly applied to sliding windows. In addition, three example evaluation windows are illustrated.	31
3.3	Segmentation of activity <i>turn on the heating</i> based on the variance of the current sensor orientation. The upper figure shows the annotation and the sensor's orientation. The lower figure contains the variance of the sensor orientation and the detected postures.	33
3.4	Average precision-recall results for classification with Posture 1, Posture 2, Diff Pos 1 & Pos 2 and the fusion of all settings.	36
3.5	Precision-recall of all activities for classification with Posture 1 (black), Posture 2 (green), Diff Pos 1 & Pos 2 (red) and the fusion of all settings (blue).	36
4.1	Set of seven gestures of different complexity that are evaluated against a realistic background class of daily gestures in five different scenarios.	42

4.2	Sequence of arm orientations for a gesture <i>B</i> projected (for illustration purposes only) onto a plane (left), Orientation of the arm, variance over the orientation and segmentation for the same gesture <i>B</i> (right).	44
4.3	Gesture <i>B</i> in 2D (left), Shape Histogram of 2nd Turning Point and Direction Histogram of the third part (middle), Area and Direction Definition for the Histograms (right).	46
4.4	Precision-recall results for the seven gestures against the background when using direction histograms only.	49
4.5	Precision-Recall results for the seven gestures against the background when using shape histograms only.	49
4.6	Precision-recall results for the seven gestures against the background when combining direction and shape histograms.	50
4.7	Worst instances of gesture <i>Flank</i> that are all taken from a single user.	52
5.1	Geometrical properties of the proposed body model (a) and ball-and-socket joint (b)	55
5.2	Upper body-model (top) and variance over hand positions (bottom) including segmentation while opening a hood based on 5 XSens inertial sensors.	56
5.3	Illustration of height features: Snapshots of the activity <i>Check Trunk</i> (Top). Right hand's height calculated using the 3D model (Middle) including areas of up (black) and down (gray) movements. Time features over primitives (Bottom).	58
5.4	Illustration of a resulting sequence of snapshots for the recurrent pushing and pulling of the right arm from the bird's-eye view, for example, as it occurs for the activity <i>Check Right Door Lock</i> . The black arrows emphasize the movement's direction of the right hand at each point in time.	60
5.5	Start (dotted black) and end (solid blue) postures while pulling the arm towards the body. Primitives are extracted based on hand (left) and elbow (middle) movements or torso turn (right)	60
5.6	Segmented push-pull primitives by torso turn, hand and elbow movements	61
5.7	Pull movement (left), direction histogram of the right hand movement (middle), direction definition for the histograms (right).	62
5.8	Illustration of the users' location while performing the 20 activities after preprocessing. For Writing (black diamonds), the location is spread over the entire area.	64

5.9	ROC curves for the 20 activities (dotted red: user-dependent, solid blue: across-user). The red circles are the user-dependent results of Ogris et al. [Ogris <i>et al.</i> 2008] (SWB = spare wheel box)	66
5.10	Matrix that relates features to classifiers from top to bottom, which shows which features are shared among the different activity classes. These features were chosen from a pool of 393 features in the first 25 rounds of boosting.	68
5.11	Weighted squared error on the training data (y-axis) as a function of the boosting rounds (x-axis). The solid red curves correspond to the eight user-dependent models converging after 40-60 rounds. In case of across-user training (blue curve), boosting requires more classifiers (100-120) to converge.	69
6.1	(a) The upper arm rotational space. (b) The lower arm constraint: The rotational space of the lower arm around is constrained to the front.	74
6.2	(a) Rotation around the gravity vector G leaves vectors on the ground plane undetermined. (b) As consequence the push-pull-primitive using the BM_ACC on horizontal plane cannot be determined. The gray limbs denote the actual posture. The black lines the posture of the BM_ACC	74
6.3	The acceleration-based sequence (top) approximates the IMU-based sequence (bottom) very well for the <i>opening hood</i> activity	80
6.4	<i>Writing.</i> (Left) The wrong acceleration based body configuration. (Right) IMU-based body configuration.	80
7.1	Illustration of boxes as they are included in the manual that can be found on the web. (a) A box displayed from front and side view highlights the final dimensions as well as the planks' adjustment. (b) The layout example of two boxes gives an impression how several boxes can be combined and used to store books.	84
7.2	(a) UML diagram illustrating possible paths when building 2 book boxes. This diagram can be extracted from existing manual descriptions of the specific high-level task. (b) User dependent paths in the data set when building 2 book boxes. Seven different sequences can be noticed in the data for level-2 activities.	86
7.3	Diverse class complexity: drilling, screwing, hammering, mark, hang up.	86
7.4	Intra-class variability when performing activity screw (fix side part). Although the subjects perform the same activity, a high variance in execution can be observed.	87

7.5	Inter-class similarity while drilling. 2 right pictures (drilling to fix side parts), 2 left pictures (drilling to join side parts). The arm postures, as well as the movements while drilling in different work steps are similar.	88
7.6	Schematics of the approach on three activity levels	92
7.7	Precision-recall of the 19 level-1 activities. Level-1 recognition (red). Recognition as inferred by level-2 recognition (blue). Recognition as inferred from level-3 processing (green).	93
7.8	Precision-recall of 6 level-2 activities. Level 2 (red) models sequential sub-activities, level 3 (green) includes all temporal constraints.	95

List of Tables

3.1	Minimum and maximum times for the considered activities in the training data. The training data contains ten performances of all activities by one subject.	28
3.2	Confusion matrix for ten short activities on hand segmented data using fully-connected HMMs with three states and single Gaussian distribution. The average recognition rate is 80%.	29
3.3	Confusion matrix for ten short activities on hand segmented data using left-right HMMs with three states and single Gaussian distribution. The average recognition rate is 88%.	30
4.1	False Positives for the seven gestures of different complexity in dependence on recall. An analysis of the number of false positives for sub-parts of gesture <i>B</i> shows the strong inverse correlation between number of primitives and false positives.	51
4.2	Confusion Matrix of the gestures B, 5, Square and Flank.	52
6.1	Top group rows show EER for activity recognition without using location. The bottom group incorporates location. For each case we use body-model, signal-oriented or their combination either based on IMU-sensor data or on acceleration	77
6.2	EER for activity recognition with 2 sensors (both IMUs and acceleration sensors) with and without location	79
6.3	EER for different types of HMMs vs. Joint Boosting	79
6.4	EER for activity recognition with 2 and 5 sensors (BM_IMU + signal) without global features	81
7.1	6 level-2 and the corresponding 19 level-1 activities necessary to build 2 book boxes.	88

Bibliography

- [Abowd 2000] Gregory D. Abowd. Classroom 2000: An experiment with the instrumentation of a living educational environment. *IBM Systems Journal*, 38, 2000. *cited on pp. 10*
- [Ali and Aggarwal 2001] Anjum Ali and J. K. Aggarwal. Segmentation and recognition of continuous human activity. *IEEE Workshop on Detection and Recognition of Events in Video*, 2001. *cited on pp. 16, 17*
- [Amft *et al.* 2005] Oliver Amft, Holger Junker, and Gerhard Tröster. Detection of eating and drinking arm gestures using inertial body-worn sensors. In *ISWC*, 2005. *cited on pp. 14, 16*
- [Andrew *et al.* 2007] Adrienne Andrew, Yaw Anokwa, Karl Koscher, Jonathan Lester, and Gaetano Borriello. Context to make you more aware. *IWSAWC*, 2007. *cited on pp. 11*
- [Andrieu *et al.* 2003] Christophe Andrieu, Nando de Freitas, Arnaud Doucet, and Michael I. Jordan. An introduction to mcmc for machine learning. *Machine Learning*, 50:5–43, 2003. *cited on pp. 100*
- [Anid Dey and Kodama 2006] Stef Streng Anid Dey, Theo Sohn and Jus Kodama. CAP: Interactive prototyping of context-aware applications. In *PERVASIVE*, May 2006. *cited on pp. 10*
- [Anliker *et al.* 2004] U. Anliker, J.A. Ward, P. Lukowicz, G. Tröster, F. Dolveck, M. Baer, F. Keita, E.B. Schenker, F. Catarsi, L. Coluccini, A. Belardinelli, D. Shklarski, M. Alon, E. Hirt, R. Schmid, and M. Vuskovic. Amon: a wearable multi-parameter medical monitoring and alert system. *IEEE Transactions on Information Technology in Biomedicine*, 8:415–427, 2004. *cited on pp. 11*
- [Antifakos *et al.* 2002] Stavros Antifakos, Florian Michahelles, and Bernt Schiele. Proactive instructions for furniture assembly. In *UbiComp*, 2002. *cited on pp. 17*
- [Apple Inc. 2009] Apple Inc. The Apple iPhone, 2009. Available from: <http://www.apple.com/iphone/> [cited April 29, 2009]. *cited on pp. 12*
- [Argyros and Lourakis 2006] A.A. Argyros and M.I.A. Lourakis. Vision-based interpretation of hand gestures for remote control of a computer mouse. In *CVHCI*, 2006. *cited on pp. 13*

- [Backman *et al.* 2006] A. Backman, K. Bodin, G. Bucht, LE Janlert, M. Maxhall, T. Pederson, D. Sjölie, B. Sondell, and D. Surie. easyadl - wearable support system for independent life despite dementia. In *CHI 2006 Workshop on Designing Technology for People with Cognitive Impairments*, 2006. *cited on pp. 11*
- [Bao and Intille 2004] Ling Bao and Stephen S. Intille. Activity recognition from user-annotated acceleration data. *Pervasive*, 2004. *cited on pp. 18, 74*
- [Bardram and Christensen 2007] Jakob E. Bardram and Henrik B. Christensen. Pervasive computing support for hospitals: An overview of the activity-based computing project. *IEEE Pervasive Computing*, 6, 2007. *cited on pp. 11*
- [Begole *et al.* 2003] James Begole, John C. Tang, and Rosco Hill. Rhythm modeling, visualizations and applications. In *UIST*, 2003. *cited on pp. 4*
- [Bennett *et al.* 1994] Frazer Bennett, Tristan Richardson, Andy Harter, and Cb Qa. Teleporting - making applications mobile. In *Proceedings of Workshop on Mobile Computing Systems and Applications*, 1994. *cited on pp. 9*
- [Borkar *et al.* 2001] Vinayak Borkar, Kaustubh Deshmukh, and Sunita Sarawagi. Automatic segmentation of text into structured records. *SIGMOD Rec.*, 30, 2001. *cited on pp. 12*
- [Bourke *et al.* 2007] AK Bourke, JV O'Brien, and GM Lyons. Evaluation of a threshold-based tri-axial accelerometer fall detection algorithm. *Gait & Posture*, 26(2), 2007. *cited on pp. 11*
- [Brand and Kettnaker 2000] Matthew Brand and Vera Kettnaker. Discovery and segmentation of activities in video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8), 2000. *cited on pp. 12*
- [Brashear and Starner 2003] Helene Brashear and Thad Starner. Using multiple sensors for mobile sign language recognition. In *ISWC*, 2003. *cited on pp. 15*
- [Brown 1996] P. J. Brown. The stick-e document: a framework for creating context-aware applications. In *Proceedings of EP'96, Palo Alto*, 1996. *cited on pp. 10*
- [Cakmakci *et al.* 2002] Ozan Cakmakci, Joelle Coutaz, Kristof Van Laerhoven, and Hans-Werner Gellersen. Context awareness in systems with limited resources. In *ECAI*, 2002. *cited on pp. 16*
- [Choudhury *et al.* 2006] T. Choudhury, M. Philipose, D. Wyatt, and J. Lester. Towards Activity Databases: Using Sensors and Statistical Models to Summarize People's Lives. *IEEE Data Eng. Bull*, 29, 2006. *cited on pp. 11*
- [Clarkson and Pentland 1999] B. Clarkson and A. Pentland. Unsupervised clustering of ambulatory audio and video. In *ICASSP*, 1999. *cited on pp. 5, 17, 18*
- [Consolvo *et al.* 2008] Sunny Consolvo, D. McDonald, T. Toscos, M. Chen, Jon Froehlich, B. Harrison, P. Klasnja, A. LaMarca, L. LeGrand, R. Libby, I. Smith, and James Landay. Activity Sensing in the Wild: A Field Trial of UbiFit Garden. In *CHI*, 2008. *cited on pp. 11*

- [Dempster *et al.* 1977] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B*, 39, 1977. *cited on pp.* 20
- [Deng and Tsui 2000] J.W. Deng and H.T. Tsui. An hmm-based approach for gesture segmentation and recognition. *International Conference on Pattern Recognition*, 3, 2000. *cited on pp.* 13, 53
- [Dey and Abowd 2000] Anind K. Dey and Gregory D. Abowd. Cybreminder: A context-aware system for supporting reminders. In *HUC*, 2000. *cited on pp.* 10
- [Dey and Mankoff 2005] Anind K. Dey and Jennifer Mankoff. Designing mediation for context-aware applications. *ACM Trans. Comput.-Hum. Interact.*, 12, 2005. *cited on pp.* 10
- [Dey 2001] Anind K. Dey. Understanding and using context. *Personal Ubiquitous Computing*, 5, 2001. *cited on pp.* 2, 9
- [Dourish 2004] Paul Dourish. What we talk about when we talk about context. *Personal Ubiquitous Comput.*, 8, 2004. *cited on pp.* 10
- [Eagle and Pentland 2006] N. Eagle and A. Pentland. Reality mining: sensing complex social systems. *Personal and Ubiquitous Computing*, 10, 2006. *cited on pp.* 10
- [Farella *et al.* 2007] Elisabetta Farella, Luca Benini, Bruno Riccò, and Andrea Acquaviva. Moca: A low-power, low-cost motion capture system based on integrated accelerometers. *Advances in Multimedia*, 2007. *cited on pp.* 73
- [Friedman *et al.* 2000] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Additive logistic regression: a statistical view of boosting. *Annals of Statistics*, 2000. *cited on pp.* 19, 20
- [Ghosh and Mitchell 2006] Payel Ghosh and Melanie Mitchell. Segmentation of medical images using a genetic algorithm. In *GECCO*, 2006. *cited on pp.* 12
- [Green and Guan 2004] R. D. Green and Ling Guan. Continuous human activity recognition. In *ICARCV*, 2004. *cited on pp.* 16, 17
- [Guerra-Filho and Aloimonos 2007] Gutemberg Guerra-Filho and Yiannis Aloimonos. A language for human action. *Computer*, 40, 2007. *cited on pp.* 16
- [Gyration 2009] Gyration. Gyration, May 2009. Available from: <http://www.gyration.com/> [cited April 29, 2009]. *cited on pp.* 23
- [Harter and Hopper 1994] Andy Harter and Andy Hopper. A distributed location system for the active office. *IEEE Network*, 8, 1994. *cited on pp.* 9
- [Harter *et al.* 1999] Andy Harter, Pete Steggles, Andy Ward, and Paul Webster. The anatomy of a context-aware application. In *Mobile Computing and Networking*, 1999. *cited on pp.* 9
- [Heinz *et al.* 2006] E.A. Heinz, K. Kunze, M. Gruber, D. Bannach, and P. Lukowicz. Using wearable sensors for real-time recognition tasks in games of martial arts - an initial experiment. In *CIG*, 2006. *cited on pp.* 12

- [Hofmann *et al.* 1998] Frank G. Hofmann, Peter Heyer, and Günter Hommel. Velocity profile based recognition of dynamic gestures with discrete hidden markov models. In *International Gesture Workshop on Gesture and Sign Language in Human-Computer Interaction*, 1998. *cited on pp.* 13
- [Horvitz *et al.* 2002] Eric Horvitz, Paul Koch, Carl M. Kadie, and Andy Jacobs. Coordinate: Probabilistic Forecasting of Presence and Availability. In *Proc. UAI*, 2002. *cited on pp.* 10
- [Huynh and Schiele 2006a] T. Huynh and B. Schiele. Towards less supervision in activity recognition from wearable sensors. In *ISWC*, 2006. *cited on pp.* 5
- [Huynh and Schiele 2006b] T. Huynh and B. Schiele. Unsupervised discovery of structure in activity data using multiple eigenspaces. In *LoCA*, 2006. *cited on pp.* 5, 18
- [Huynh *et al.* 2008] Tâm Huynh, Mario Fritz, and Bernt Schiele. Discovery of activity patterns using topic models. In *UbiComp*, 2008. *cited on pp.* 18
- [Iacucci *et al.* 2004] Giulio Iacucci, Juha Kela, and Pekka Pehkonen. Computational support to record and re-experience visits. *Personal Ubiquitous Comput.*, 8, 2004. *cited on pp.* 13
- [Ida and Sambonsugi 1995] T. Ida and Y. Sambonsugi. Image segmentation using fractal coding. 5:567–570, 1995. *cited on pp.* 12
- [Jaakkola and Haussler 1998] T. Jaakkola and D. Haussler. Exploiting generative models in discriminative classifiers. *Advances in Neural Information Processing Systems*, 11:487–493, 1998. *cited on pp.* 19
- [Jafari *et al.* 2007] R. Jafari, W. Li, R. Bajcsy, S. Glaser, and S. Sastry. Physical Activity Monitoring for Assisted Living at Home. *BSN*, 2007. *cited on pp.* 11
- [Junker *et al.* 2008] Holger Junker, Oliver Amft, Paul Lukowicz, and Gerhard Tröster. Gesture spotting with body-worn inertial sensors to detect user activities. *Pattern Recognition*, 41:2010–2024, 2008. *cited on pp.* 14, 16
- [Kallio *et al.* 2006] Sanna Kallio, Juha Kela, Panu Koripää, and Jani Mäntyjärvi. User independent gesture interaction for small handheld devices. *IJPRAI*, 20:505–524, 2006. *cited on pp.* 15, 53
- [Kela *et al.* 2006] Juha Kela, Panu Koripää, Jani Mäntyjärvi, Sanna Kallio, Giuseppe Savino, Luca Jozzo, and D. Marca. Accelerometer-based gesture control for a design environment. *Personal Ubiquitous Comput.*, 2006. *cited on pp.* 13, 15, 74
- [Keogh *et al.* 2001] Eamonn Keogh, Selina Chu, David Hart, and Michael Pazzani. An online algorithm for segmenting time series. *Data Mining*, page 289, 2001. *cited on pp.* 14
- [Kern *et al.* 2003] Nicky Kern, Bernt Schiele, and Albrecht Schmidt. Multi-sensor activity context detection for wearable computing. In *In Proc. EUSAI, LNCS*, 2003. *cited on pp.* 16

- [Ko *et al.* 2008] Ming Hsiao Ko, Geoff West, Svetha Venkatesh, and Mohan Kumar. Using dynamic time warping for online temporal fusion in multisensor systems. *Inf. Fusion*, 9:370–388, 2008. *cited on pp. 14*
- [Kojima *et al.* 2002] Atsuhiro Kojima, Takeshi Tamura, and Kunio Fukunaga. Natural language description of human activities from video images based on concept hierarchy of actions. *Int. J. Comput. Vision*, 50:171–184, 2002. *cited on pp. 16, 17*
- [Krumm and Horvitz 2006] J. Krumm and E. Horvitz. Predestination: Inferring Destinations from Partial Trajectories. In *UbiComp*, 2006. *cited on pp. 17*
- [Langheinrich 2005] Marc Langheinrich. *Personal Privacy in Ubiquitous Computing – Tools and System Support*. PhD thesis, ETH Zurich, 2005. *cited on pp. 5*
- [Lee and Kim 1998] Hyeon-Kyu Lee and Jin-Hyung Kim. Gesture spotting from continuous hand motion. *Pattern Recogn. Lett.*, 19:513–520, 1998. *cited on pp. 14*
- [Lee and Kim 1999] Hyeon-Kyu Lee and Jin H. Kim. An hmm-based threshold model approach for gesture recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21:961–973, 1999. *cited on pp. 14*
- [Lee and Xu 1996] C. Lee and Yangsheng Xu. Online, interactive learning of gestures for human/robot interfaces. In *Robotics and Automation*, 1996. *cited on pp. 13*
- [Lester *et al.* 2005] Jonathan Lester, Tanzeem Choudhury, Nicky Kern, Gaetano Borriello, and Blake Hannaford. A hybrid discriminative/generative approach for modeling human activities. In *IJCAI*, 2005. *cited on pp. 53, 72, 99*
- [Lester *et al.* 2006] Jonathan Lester, Tanzeem Choudhury, and Gaetano Borriello. A practical approach to recognizing physical activities. In *Lecture Notes in Computer Science : Pervasive Computing*, 2006. *cited on pp. 75*
- [Liao *et al.* 2007a] Lin Liao, Dieter Fox, and Henry Kautz. Extracting places and activities from gps traces using hierarchical conditional random fields. *Int. J. Rob. Res.*, 26:119–134, 2007. *cited on pp. 5, 18*
- [Liao *et al.* 2007b] Lin Liao, Donald J. Patterson, Dieter Fox, and Henry Kautz. Learning and inferring transportation routines. *Artif. Intell.*, 171:311–331, 2007. *cited on pp. 17*
- [Liszka *et al.* 2004] K. Liszka, Michael A. Mackin, Michael J. Lichter, David W. York, Dilip Pillai, and David S. Rosenbaum. Keeping a Beat on the Heart. *IEEE Pervasive Computing*, 3:42–49, 2004. *cited on pp. 11*
- [Lühr *et al.* 2003] Sebastian Lühr, Hung H. Bui, Svetha Venkatesh, and Geoff A. W. West. Recognition of human activity through hierarchical stochastic learning. In *PERCOM*, 2003. *cited on pp. 17*
- [Lukowicz *et al.* 2004] Paul Lukowicz, Jamie A. Ward, Holger Junker, Mathias Stäger, Gerhard Tröster, Amin Atrash, and Thad Starner. Recognizing workshop activity using body worn microphones and accelerometers. In *Pervasive*, 2004. *cited on pp. 13, 15*
- [Lukowicz *et al.* 2007] P. Lukowicz, A. Timm-Giel, M. Lawo, and O. Herzog. Wearit@work: Toward real-world industrial wearable computing. *Pervasive Computing*, 2007. *cited on pp. 11*

- [Lyons and Starner 2001] Kent Lyons and Thad Starner. Mobile capture for wearable computer usability testing. In *ISWC*, 2001. *cited on pp. 4*
- [Maitland *et al.* 2006] Julie Maitland, Scott Sherwood, Louise Barkhuus, Ian Anderson, Malcolm Hall, Barry Brown, Matthew Chalmers, and Henk Muller. Increasing the awareness of daily activity levels with pervasive computing. In *Pervasive Computing Technologies for Healthcare*, 2006. *cited on pp. 11*
- [Mäntyjarvi *et al.* 2004] J Mäntyjarvi, J Kela, P Korppiä, and S Kallio. Enabling fast and effortless customisation in accelerometer based gesture interaction. *MUM*, 2004. *cited on pp. 15*
- [Mayrhofer and Gellersen 2007] R. Mayrhofer and Hans Gellersen. Shake well before use: Authentication based on accelerometer data. In *Pervasive*, 2007. *cited on pp. 5*
- [Milone *et al.* 2002] D.H. Milone, J.J. Merelo, and H.L. Rufiner. Evolutionary algorithm for speech segmentation. *E-Commerce Technology*, 2:1115–1120, 2002. *cited on pp. 12*
- [Min and Kasturi 2004] Junghye Min and Rangachar Kasturi. Activity recognition based on multiple motion trajectories. In *ICPR*, 2004. *cited on pp. 13*
- [Minnen *et al.* 2006] David Minnen, Thad Starner, Irfan Essa, and Charles Isbell. Discovering characteristic actions from on-body sensor data. In *ISWC*, 2006. *cited on pp. 13*
- [Minnen *et al.* 2007a] D. Minnen, C. Isbell, M. Essa, and T. Starner. Detecting subdimensional motifs: An efficient algorithm for generalized multivariate pattern discovery. In *ICDM*, 2007. *cited on pp. 18*
- [Minnen *et al.* 2007b] D. Minnen, T. Westeyn, D. Ashbrook, P. Presti, and T. Starner. Recognizing Soldier Activities in the Field. *BSN*, 2007. *cited on pp. 12*
- [Mizell and Cray 2003] D. Mizell and I. Cray. Using gravity to estimate accelerometer orientation. In *ISWC*, 2003. *cited on pp. 73*
- [M.J.Swain and D.H.Ballard 1991] M.J.Swain and D.H.Ballard. Color indexing. *IJCV*, 1991. *cited on pp. 48*
- [Mäntylä *et al.* 2000] V. M. Mäntylä, J. Mäntyjärvi, T. Seppänen, and E. Tuulari. Hand gesture recognition of a mobile device user. In *ICME*, 2000. *cited on pp. 13, 15, 53*
- [Morguet and Lang 1998a] Peter Morguet and Manfred Lang. An integral stochastic approach to image sequence segmentation and classification. *ICASSP*, 5:2705–2708, 1998. *cited on pp. 14*
- [Morguet and Lang 1998b] Peter Morguet and Manfred Lang. Spotting dynamic hand gestures in video image sequences using hidden markov models. In *ICIP*, 1998. *cited on pp. 14*
- [Moven 2009] Moven. XSens Moven, May 2009. Available from: http://www.moven.com/en/home_moven.php [cited April 29, 2009]. *cited on pp. 16, 17, 55, 56*

- [Murphy 2009] Kevin Murphy. Hidden Markov Toolbox for Matlab, May 2009. Available from: <http://www.cs.ubc.ca/~murphyk/Software/HMM/hmm.html> [cited April 29, 2009]. *cited on pp. 28*
- [Nagel *et al.* 2001] Kris Nagel, Cory D. Kidd, Thomas O'Connell, Anind Dey, and Gregory D. Abowd. The family intercom: Developing a context-aware audio communication system. *Lecture Notes in Computer Science*, 2001. *cited on pp. 10*
- [Ng and Jordan 2002] A. Ng and M. Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In *Advances in Neural Information Processing Systems*, 2002. *cited on pp. 19*
- [Nintendo 2009] Nintendo. Wii Entertainment System, May 2009. Available from: <http://www.nintendo.com/wii> [cited April 29, 2009]. *cited on pp. 12*
- [Niu and Abdel-Mottaleb 2005] Feng Niu and M. Abdel-Mottaleb. Hmm-based segmentation and recognition of human activities from video sequences. *Multimedia and Expo*, 0:804–807, 2005. *cited on pp. 12*
- [OECD 2009] OECD. Organization for economic co-operation and development, May 2009. Available from: <http://www.oecd.org/> [cited April 29, 2009]. *cited on pp. 100*
- [Ogris *et al.* 2007] G. Ogris, M. Kreil, and P. Lukowicz. Using fsr based muscle activity monitoring to recognize manipulative arm gestures. In *ISWC*, 2007. *cited on pp. 53*
- [Ogris *et al.* 2008] G. Ogris, T. Stiefmeier, P. Lukowicz, and G. Tröster. Using a complex multi-modal on-body sensor system for activity spotting. In *ISWC*, 2008. *cited on pp. 15, 18, 53, 54, 65, 66, 67, 103*
- [Oliver *et al.* 2002] Nuria Oliver, Eric Horvitz, and Ashutosh Garg. Layered representations for human activity recognition. In *ICMI*, 2002. *cited on pp. 17*
- [OpenCVLibrary 2009] OpenCVLibrary. OpenCVLibrary, May 2009. Available from: <http://opencv.willowgarage.com/wiki/> [cited April 29, 2009]. *cited on pp. 23*
- [Paradiso *et al.* 2005] R. Paradiso, G. Loriga, and N. Taccini. A wearable health care system based on knitted integrated sensors. *Information Technology in Biomedicine*, 9:337–344, 2005. *cited on pp. 11*
- [Patterson *et al.* 2004] Donald J. Patterson, Lin Liao, Krzysztof Gajos, Michael Collier, Nik Livic, Katherine Olson, Shiaokai Wang, Dieter Fox, and Henry Kautz. Opportunity Knock: a System to Provide Cognitive Assistance with Transportation Services. In *UBICOMP*, 2004. *cited on pp. 18*
- [Patterson *et al.* 2005] Donald J. Patterson, Dieter Fox, Henry Kautz, and Matthai Philipose. Fine-grained activity recognition by aggregating abstract object usage. In *ISWC*, 2005. *cited on pp. 15*
- [Pentland *et al.* 2005] A. Pentland, T. Choudhury, N. Eagle, and P. Singh. Human dynamics: computation for organizations. *Pattern Recognition Letters*, 26:503–511, 2005. *cited on pp. 10*

- [Pentland 2007] A. Pentland. Automatic mapping and modeling of human networks. *Physica A: Statistical Mechanics and its Applications*, 378:59–67, 2007. *cited on pp. 10*
- [Perkowitz *et al.* 2004] Mike Perkowitz, Matthai Philipose, Kenneth Fishkin, and Donald J. Patterson. Mining models of human activities from the web. In *WWW*, 2004. *cited on pp. 17*
- [Perng *et al.* 1999] K Perng, B Fisher, S Hollar, and K Pister. Acceleration sensing glove (asg). *ISWC*, 1999. *cited on pp. 15*
- [Pinhanez 1999] Claudio Santos Pinhanez. *Representation and recognition of action in interactive spaces*. PhD thesis, 1999. *cited on pp. 18*
- [Pylvänäinen 2005] Timo Pylvänäinen. Accelerometer based gesture recognition using continuous hmms. In *IbPRIA*, 2005. *cited on pp. 15, 53*
- [Quint 2000] Julien Quint. A formalism for universal segmentation of text. In *Proceedings of the 18th conference on Computational linguistics*, 2000. *cited on pp. 12*
- [Rabiner 1989] Lawrence Rabiner. A tutorial on hmm and selected applications in speech recognition. *Proceedings of the IEEE*, 1989. *cited on pp. 19, 20, 28*
- [Rekimoto 2001] Jun Rekimoto. Gesturewrist and gesturepad: Unobtrusive wearable interaction devices. In *ISWC*, 2001. *cited on pp. 15*
- [Ryoo and Aggarwal 2006] M. S. Ryoo and J. K. Aggarwal. Recognition of composite human activities through context-free grammar based representation. In *CVPR*, 2006. *cited on pp. 16, 17, 56*
- [Schmidt 2000] Albrecht Schmidt. Implicit human computer interaction through context. Technical report, Personal Technologies, 2000. *cited on pp. 1*
- [Shi *et al.* 2004] Y. Shi, Y. Huang, D. Minnen, A. Bobick, and I. Essa. Propagation networks for recognition of partially ordered sequential action. In *CVPR*, 2004. *cited on pp. 17*
- [Si *et al.* 2007] Hua Si, Seung Jin Kim, Nao Kawanishi, and Hiroyuki Morikawa. A context-aware reminding system for daily activities of dementia patients. In *ISWAWC*, 2007. *cited on pp. 11*
- [Sklansky 1978] J Sklansky. Image segmentation and feature extraction. *IEEESMC*, 8:337–347, 1978. *cited on pp. 12*
- [Slyper and Hodgins 2008] Ronit Slyper and Jessica Hodgins. Action capture with accelerometers. In *Eurographics Symposium on Computer Animation*, 2008. *cited on pp. 73*
- [Starner *et al.* 1998] T Starner, J Weaver, and A Pentland. Real-time american sign language recognition using desk and wearable computer based video. In *IEEE TPAMI*, 20, 1998. *cited on pp. 15*
- [Starner 2001] Thad Starner. The challenges of wearable computing: Part 2. *IEEE Micro*, 21:54–67, 2001. *cited on pp. 4*

- [Stiefmeier *et al.* 2006] Thomas Stiefmeier, Georg Ogris, Holger Junker, Paul Lukowicz, and Gerhard Tröster. Combining motion sensors and ultrasonic hands tracking for continuous activity recognition in a maintenance scenario. *ISWC*, 0:97–104, 2006. *cited on pp.* 15, 53
- [Stiefmeier *et al.* 2007] Thomas Stiefmeier, Daniel Roggen, and Gerhard Tröster. Gestures are strings: Efficient online gesture spotting and classification using string matching. In *BodyNets*, 2007. *cited on pp.* 53
- [Stiefmeier *et al.* 2008] T. Stiefmeier, D. Roggen, G. Troster, G. Ogris, and P. Lukowicz. Wearable activity tracking in car manufacturing. *Pervasive Computing*, 7:42–50, 2008. *cited on pp.* 15, 65
- [Tapia and Intille 2007] Emmanuel Munguia Tapia and Stephen Intille. Real-time recognition of physical activities and their intensities using wireless accelerometers and a heart rate monitor. In *ISWC*, 2007. *cited on pp.* 18
- [Tentori and Favela 2008] M. Tentori and J. Favela. Activity-Aware Computing for Healthcare. *IEEE Pervasice Computing*, pages 51–57, 2008. *cited on pp.* 11
- [Tiesel and Loviscach 2006] J.P. Tiesel and J. Loviscach. A Mobile Low-Cost Motion Capture System Based on Accelerometers. *Advances in Visual Computing*, 2006. *cited on pp.* 73
- [Torralba *et al.* 2007] A. Torralba, K. P. Murphy, and W. T. Freeman. Sharing visual features for multiclass and multiview object detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29:854–869, 2007. *cited on pp.* 19, 20, 54
- [Ubisense 2009] Ubisense. Ubisense, May 2009. Available from: <http://www.ubisense.com> [cited April 29, 2009]. *cited on pp.* 59, 65, 72
- [UML 2009] UML. Unified Modeling Language, May 2009. Available from: <http://www.uml.org/> [cited April 29, 2009]. *cited on pp.* 85
- [Van Laerhoven *et al.* 2003] K. Van Laerhoven, N. Kern, H.W. Gellersen, and B. Schiele. Towards a wearable inertial sensor network. In *Proc. of the IEE Eurowearable*, pages 125–130, 2003. *cited on pp.* 18
- [Van Laerhoven *et al.* 2006] K. Van Laerhoven, H.W. Gellersen, and Y.G. Malliaris. Long-Term Activity Monitoring with a Wearable Sensor Node. In *BSN Workshop*, 2006. *cited on pp.* 4
- [Van Laerhoven *et al.* 2008] K. Van Laerhoven, D. Kilian, and B. Schiele. Using rhythm awareness in long-term activity recognition. In *ISWC*, 2008. *cited on pp.* 4
- [van Sinderen *et al.* 2006] M. J. van Sinderen, A. T. van Halteren, M. Wegdam, H. B. Meeuwissen, and E. H. Eertink. Supporting context-aware mobile applications: an infrastructure approach. *IEEE Communications Magazine*, 44:96–104, 2006. *cited on pp.* 10
- [Vardy *et al.* 1999] Andrew Vardy, John Robinson, and Li te Cheng. The wristcam as input device. In *ISWC*, 1999. *cited on pp.* 15

- [Villalba *et al.* 2006] E. Villalba, M. Ottaviano, M.T. Arredondo, A. Martinez, and S. Guillen. Wearable monitoring system for heart failure assessment in a mobile environment. In *Computers in Cardiology*, 2006. *cited on pp.* 11
- [Viola and Jones 2004] Paul Viola and Michael J. Jones. Robust real-time face detection. *IJCV*, 57:137–154, 2004. *cited on pp.* 20, 23
- [Wang *et al.* 2007] Shiaokai Wang, William Pentney, Ana-Maria Popescu, Tanzeem Choudhury, and Matthai Philipose. Common sense based joint training of human activity recognizers. In *IJCAI*, 2007. *cited on pp.* 17
- [Want *et al.* 1992] Roy Want, Andy Hopper, Veronica Falcao, and Jonathan Gibbons. The active badge location system. *ACM Trans. Inf. Syst.*, 10:91–102, 1992. *cited on pp.* 9
- [Ward *et al.* 2006] Jamie A. Ward, Paul Lukowicz, Gerhard Tröster, and Thad Starner. Activity recognition of assembly tasks using body-worn microphones and accelerometers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28, 2006. *cited on pp.* 11, 18, 53, 74, 75
- [Xiang 2003] S. Gong & T. Xiang. Recognition of group activities using a dynamic probabilistic network. In *ICCV*, 2003. *cited on pp.* 18
- [XSens 2009] XSens. XSens Technologies, May 2009. Available from: <http://www.xsens.com> [cited April 29, 2009]. *cited on pp.* 4, 16, 27, 89
- [Zappi *et al.* 2008] Piero Zappi, Clemens Lombriser, Thomas Stiefmeier, Elisabetta Farella, Daniel Roggen, Luca Benini, and Gerhard Tröster. Activity recognition from on-body sensors: Accuracy-power trade-off by dynamic sensor selection. In *EWSN*, 2008. *cited on pp.* 75
- [Zatsiorsky 1997] Vladimir M. Zatsiorsky. *Kinematics of Human Motion*. Human Kinetics Publishers, September 1997. Available from: <http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/0880116765>. *cited on pp.* 55
- [Zhang and Hartmann 2007] H. Zhang and B. Hartmann. Building upon everyday play. In *CHI*, 2007. *cited on pp.* 12
- [Zinnen and Schiele 2008] Andreas Zinnen and Bernt Schiele. A new approach to enable gesture recognition in continuous data streams. In *ISWC*, 2008. *cited on pp.* 7
- [Zinnen *et al.* 2007a] Andreas Zinnen, Kristof Van Laerhoven, and Bernt Schiele. Toward recognition of short and non-repetitive activities from wearable sensors. In *AmI*, 2007. *cited on pp.* 7
- [Zinnen *et al.* 2007b] Andreas Zinnen, Thomas Ziegert, and Bernt Schiele. Browsing patient records during ward rounds with a body worn gyroscope. 2007. *cited on pp.* 7, 12
- [Zinnen *et al.* 2008] Andreas Zinnen, Andreas Faatz, Eicke Godehardt, Manuel Goertz, and Robert Lokaiczky. Privacy issues when rolling out an e-learning solution. In *ED-MEDIA*, 2008. *cited on pp.* 5, 7, 100

[Zinnen *et al.* 2009a] Andreas Zinnen, Ulf Blanke, and Bernt Schiele. An analysis of sensor-oriented vs. model-based activity recognition. In *ISWC*, 2009. *cited on pp. 7*

[Zinnen *et al.* 2009b] Andreas Zinnen, Christian Wojek, and Bernt Schiele. Multi activity recognition based on bodymodel-derived primitives. In *LoCA*, 2009. *cited on pp. 7*



Andreas Zinnen

Ausbildung

- 07/1984–06/1997 **Abitur**, *Gymnasium, Hermeskeil*.
- 07/1997–04/1998 **Grundwehrdienst**, *Heeresmusikkorps 300, Koblenz*.
- 10/1998–07/2004 **Dipl. Inf.**, *TU Kaiserslautern*.
Diplomarbeit: Sicherheit in verteilten Informationssystemen
- 10/2005–07/2009 **Doktorand**, *TU Darmstadt & SAP Research*.
Dissertation: Spotting Human Activities and Gestures in Continuous Data Streams

Beschäftigungsverhältnisse

- 10/2000–10/2002 **Wissenschaftliche Hilfskraft**, *TU Kaiserslautern*.
- 11/2002–11/2003 **Praktikant**, *IBM Silicon Valley Lab, San Jose, CA*.
- 10/2004–09/2005 **Software-Engineer**, *Capgemini sd&m , München*.
- 10/2005–09/2009 **Doktorand**, *TU Darmstadt & SAP Research, Darmstadt*.

Wissenschaftliche Projekte

- 10/2005–12/2007 **wearIT@work**, *Tragbare Computer am Arbeitsplatz der Zukunft*.
- 05/2005–08/2008 **CESORA**, *Analyse und Design von Kontext-bezogenen Diensten*.
- 01/2008–07/2009 **APOSDE**, *Integriertes Lernen am Arbeitsplatz*.
- 09/2007–07/2009 **SiWear**, *Sichere Wearable-Systeme zur Kommissionierung industrieller Güter*.

Lehrtätigkeit

- WS2006/2007 **Einführung in Human Computer Systems**, *300+ Studenten*, Prof. Schiele.
Konzeption/Organisation der Übungen und Klausuren, Vorlesung Audio
- WS2007/2008 **Einführung in Human Computer Systems**, *300+ Studenten*, Prof. Schiele.
Konzeption/Organisation der Übungen und Klausuren, Vorlesungen Audio und Principal Component Analysis
- SS2008 **Einführung in Human Computer Systems**, *300+ Studenten*, Prof. Schiele.
Konzeption/Organisation der Übungen und Klausuren, Vorlesung Audio und Spracherkennung

Languages

Englisch	Fortgeschritten	<i>Fließend in Wort und Schrift</i>
Französisch	Fortgeschritten	<i>Sehr gute Kenntnisse in Wort und Schrift</i>
Spanisch	Intermediär	<i>Fortgeschrittene Grundkenntnisse in Wort und Schrift</i>

Förderungen

10/2000–07/2004 **TOPAZ**, *Mitglied im Siemens Student Program.*

Seminare im Rahmen von TOPAZ

- 03/2001 **Einstiegsseminar.**
- 06/2001 **Teambildung und Persönlichkeitsentwicklung.**
- 01/2002 **Grundlagen des Projektmanagements.**
- 03/2002 **Betriebswirtschaft für Ingenieure und Naturwissenschaftler.**

Literatur

Andreas Zinnen, Ulf Blanke, and Bernt Schiele. An analysis of sensor-oriented vs. model-based activity recognition. In *ISWC*, 2009.

Andreas Zinnen, Andreas Faatz, Eicke Godehardt, Manuel Goertz, and Robert Lokaiczyk. Privacy issues when rolling out an e-learning solution. In *ED-MEDIA*, 2008.

Andreas Zinnen, Kristof Van Laerhoven, and Bernt Schiele. Toward recognition of short and non-repetitive activities from wearable sensors. In *Aml*, 2007.

Andreas Zinnen and Bernt Schiele. A new approach to enable gesture recognition in continuous data streams. In *ISWC*, 2008.

Andreas Zinnen, Christian Wojek, and Bernt Schiele. Multi activity recognition based on bodymodel-derived primitives. In *LoCA*, 2009.

Andreas Zinnen, Thomas Ziegert, and Bernt Schiele. Browsing patient records during ward rounds with a body worn gyroscope. In *ISWC*, 2007.