Georg Ogris

Multi-modal on-body sensing of human activities

Dissertation

Fakultät für Informatik und Mathematik
Universität Passau

November 2009

Multi-modal on-body sensing of human activities

A dissertation submitted to the

UNIVERSITÄT PASSAU
FAKULTÄT FÜR INFORMATIK UND MATHEMATIK

in partial fulfillment of the requirements
for the degree of
Dr. rer. nat.

presented by

GEORG OGRIS

M.Sc. UMIT Hall i.T. (2006)
Dipl. Ing. (FH) Technikum Wien (2003)
born August 25, 1975

accepted on the recommendation of

Prof. Dr. Paul Lukowicz
Prof. Dr. Bernt Schiele

November, 2009

Georg Ogris

Multi-modal on-body sensing
of human activities

Revised submission
November 2009
First submission: March 2009
Published by Universität Passau

Contents

Abstract

Increased usage and integration of state-of-the-art information technology in our everyday work life aims at increasing the working efficiency. Due to unhandy human-computer-interaction methods this progress does not always result in increased efficiency, for mobile workers in particular. Activity recognition based contextual computing attempts to balance this interaction deficiency.

This work investigates wearable, on-body sensing techniques on their applicability in the field of human activity recognition. More precisely we are interested in the spotting and recognition of so-called manipulative hand gestures. In particular the thesis focuses on the question whether the widely used motion sensing based approach can be enhanced through additional information sources.

The set of gestures a person usually performs on a specific place is limited – in the contemplated production and maintenance scenarios in particular. As a consequence this thesis investigates whether the knowledge about the user's hand location provides essential hints for the activity recognition process. In addition, manipulative hand gestures – due to their *object manipulating character* – typically start in the moment the user's hand reaches a specific place, e.g. a specific part of a machinery. And the gestures most likely stop in the moment the hand leaves the position again. Hence this thesis investigates whether hand location can help solving the spotting problem.

Moreover, as user-independence is still a major challenge in activity recognition, this thesis investigates location context as a possible key component in a user-independent recognition system.

We test a Kalman filter based method to blend absolute position readings with orientation readings based on inertial measurements. A filter structure is suggested which allows up-sampling of slow absolute position readings, and thus introduces higher dynamics to the position estimations. In such a way the position measurement series is made aware of wrist *motions* in addition to the wrist *position*. We suggest location based gesture spotting and recognition approaches. Various methods to model the location classes used in the spotting and recognition stages as well as different location distance measures are suggested and evaluated.

In addition a rather novel sensing approach in the field of human activity recognition is studied. This aims at compensating drawbacks of the mere motion sensing based approach. To this end we develop a wearable hardware architecture for lower arm muscular activity measurements. The sensing hardware based on force sensing resistors is designed to have a high dynamic range. In contrast to preliminary attempts the proposed new design makes hardware calibration unnecessary.

Finally we suggest a modular and multi-modal recognition system; modular with respect to sensors, algorithms, and gesture classes. This means that adding or removing a sensor modality or an additional algorithm has little impact on the rest of the recognition system. Sensors and algorithms used for spotting and recognition can be selected and fine-tuned separately for each single activity. New activities can be added without impact on the recognition rates of the other activities.

Keywords: *Wearable computing – Contextual computing – Manipulative hand gestures – Gesture spotting – Gesture recognition – Force sensing resistors – Muscular activity recognition – Location sensing – Kalman filter based sensor fusion – Location modeling*

Zusammenfassung

Informationstechnologie wird zunehmend in unseren Arbeitsalltag integriert – eine Entwicklung, die vor allem auch eine Steigerung der Arbeitsproduktivität erwarten läßt. Allerdings wird aufgrund ungeeigneter und ineffizienter Eingabe- und Interaktionsmethoden die Effizienz nur bedingt erhöht. Effizienzprobleme ergeben sich im besonderen bei Tätigkeiten, die die Benutzung mobiler Geräte erfordern. Kontext-gesteuerter Mensch-Computer-Interaktion, aufbauend auf einer detaillierten Erkennung der momentanen Aktivität des Benutzers, wird bescheinigt, dieses Defizit ausgleichen zu können.

Die vorliegende Arbeit untersucht nun unauffällig am Körper bzw. in der Kleidung getragene Sensorsysteme auf ihre Anwendbarkeit zur Erkennung von menschlichen Aktivitäten. Im Speziellen befasst sich die Arbeit mit so genannten interaktiven Handgesten. Die Frage, ob der weit verbreitete, auf Bewegungssensoren (hauptsächlich Beschleunigungs- und Gyroskopsensoren) basierende Ansatz mittels zusätzlicher Sensorik verbessert werden kann, wird untersucht.

Die Menge der Gesten, die eine Person an einem bestimmten Ort für gewöhnlich ausführt, ist endlich und klein – vor allem in den angedachten Szenarien industrielle Serienproduktion und industrielle Wartung. Ausgehend von dieser Überlegung untersucht die Arbeit, ob die Kenntnis des räumlichen Zustandes einer Person essentielle Information für den Erkennungsprozess bereit stellt.

Interaktive Gesten starten und stoppen typischerweise in dem Moment, in dem die Hand eine bestimmte Position erreicht bzw. verlässt; vor allem auch weil diese interaktiven Gesten per se auf ein bestimmtes Objekt bezogen sein müssen, mit dem Ziel dieses eben – in welcher Form auch immer – zu verändern bzw. mit diesem zu interagieren. Diese Arbeit untersucht im Speziellen, ob die Erfassung der Handposition dazu beitragen kann, das so genannte *Spotting*-Problem zu lösen, also das Auffinden von Bereichen in den Sensorsignalen, die möglicherweise eine relevante Aktivität beinhalten.

Weiters gilt personenunabhängige Erkennung als eines der schwerwiegenden Probleme in der Gestenerkennung. Da räumliche Information an sich personenunabhängig sein muss, untersucht diese Arbeit auch, ob aufgrund der ortsbasierten Erkennung ein höheres Maß an Personenunabhängigkeit erzielt werden kann.

Um absolute Positionsmessungen mit relativen Bewegungs- und Orientierungsmessungen zu fusionieren, evaluiert diese Arbeit einen Kalman-Filter-basierten Ansatz. Die entworfene Filterstruktur erlaubt es, die langsame Abtastfrequenz der Positionsabschätzungen um den Faktor 70 zu steigern mit dem Ergebnis einer dynamischen Positionstrajektorie. Die Positionstrajektorie dient damit nicht mehr ausschließlich der Positionsabschätzung sondern auch der Bewegungsabschätzung. Methoden zur positionsbasierten Auffindung und Erkennung von Handgesten werden vorgeschlagen und untersucht. Unterschiedliche Ansätze zur Modellierung von Ortsklassen und den dazugehörigen Distanzmaßen werden vorgestellt und evaluiert.

In weiterer Folge werden Kraft-sensitive Folienwiderstände (FSR) als neuer Ansatz zur Erkennung von Aktivitäten untersucht. Mithilfe dieser Sensorik können bestimmte Nachteile klassischer Bewegungssensoren ausgeglichen werden. Zu diesem Zweck wird eine Sensorplattform zur Messung von Muskelaktivitäten des Unterarmes entwickelt. Gegenüber früherer Systeme bietet diese Plattform einen

großen dynamischen Messbereich, ohne benutzerabhängige Kalibrierung notwendig zu machen.

Das vorgestellte Erkennungssystem zielt auf Modularität durch Multimodalität ab – modular einerseits im Bezug auf die verwendeten Sensormodalitäten bzw. deren algorithmische Verarbeitung und andererseits im Bezug auf die Anzahl der Aktivitätsklassen. Das Ziel ist es, die Erkennungsraten des Systems unabhängig von der Anzahl der Klassen und unabhängig von der Sensorik und den Algorithmen zu halten. Modelle für die Erkennung einzelner Klassen können darüber hinaus unabhängig voneinander trainiert und optimiert werden.

Acknowledgments

First of all, I would like to thank my academic advisor Prof. Dr. Paul Lukowicz for giving me the opportunity to work at his institutes at UMIT and University of Passau. I thank him for his excellent guidance and support over the past years. Special thanks go to Prof. Dr. Bernt Schiele for co-examining my dissertation.

I owe many thanks to David Bannach for sharing his knowledge, endless and fruitful discussions and for being a great room-mate over the years. I would like to thank Matthias Kreil for help with various experiments and demonstrators. Thanks to Karl Stockinger for his great support in hardware questions. Thanks to all my colleagues at the ESL at University of Passau and at CSN:::UMIT for great help and for providing an inspiring and motivating work ambience.

Special thanks to Thomas Stiefmeier of ETH Zürich for great companionship throughout my whole PhD years. Thanks for months of joint coding sessions via skype. Thanks for enabling and facilitating various experiments. Great thanks go to all the people and alumni of the Wearable Computing Lab. at ETH Zürich for providing me a second academic home during the past five years.

I appreciate the help and teamwork I experienced within the wearIT@work project. Special thanks go to the partners of the car production scenario. I owe many thanks to all the people who volunteered for experiments and data recordings: Andreas, Bernd, Clemens, Corinne, Cornelia, Enno, Fatin, Helga, Holger H., Holger J., Jan, Johannes, Kilian, Marc, Matthias, Michael, Philip, and Thiemo.

I am indebted to Herbert Birnbaumer for proofreading a preliminary version of this work.

I wish to thank my friends and family for encouragement and welcome distraction. I am deeply thankful to Katrin for her loving and motivating support and for being there.

Passau, Tuesday 24th November, 2009 GEORG OGRIS

Chapter 1

Introduction

This chapter gives a brief statement on the need for contextual and wearable computing and moreover on activity and gesture recognition as one potential enabling technique. Some of the major challenges within the field of gesture spotting and recognition from continuous data streams of on-body sensors are described. Subsequently this thesis' approach to tackle these issues – i.e. a multi-modal mixed motion-location-muscular activity sensing approach – is fostered.

Moreover, this introductory chapter summarizes ongoing work from related research areas, such are (continuous) activity and gesture recognition with various sensing approaches, location awareness and indoor positioning approaches, and contemplated context aware and wearable computing application scenarios.

Finally this chapter summarizes the thesis by means of contrasting the objectives with the contributions and by means of outlining the subsequent chapters.

1.1 Status quo

The miniaturization of integrated electronic circuits has been proceeding for approximately five decades and is still advancing. Among other things this miniaturization resulted in a continuing downsizing of electronic computing devices and subsequently in mobile and ubiquitous computing power. People's habits concerning both their private and their professional lives are constantly changing towards a growing employment of but also a growing dependence on this permanently available computing power. This development does not necessarily result in increased efficiency; in some cases it may even result in the contrary, or at least it may feel like it. Even more so due to the great limitations concerning the usability of current human-computer-interaction (HCI) methods.

Among other strategies *context-aware* and *wearable* computing techniques are trying to balance this interaction deficiency. A context-aware device intends to act according to pre-estimated, most probable, prospective user demands. In some contemplated application areas wearable computing can be seen to a certain extent as one possible manifestation of context-aware computing. Admittedly, wearable computing systems solve complex context-aware computing tasks at present only under laboratory conditions; hence wearable computing devices will not be *wearable* for the short term whereas consumer electronic *wearables* will not comply wearable computing paradigms – at least for the short term.

For hardware purposes the major challenges of wearable computing are unobtrusiveness and more advanced sensing technologies. The ongoing progress in the field of miniaturization of electronic devices, in particular the high integration of electromechanical systems – see e.g. the advancing MEMS technology – will provide new opportunities in the near future.

Energy consumption is of course a major issue. Currently mobile energy storage solutions are the limiting factor when miniaturizing wearable devices both in size and weight. Efforts made in the field of miniaturized power generators e.g. converting body heat or kinetic energy into electrical energy – often referred to as energy harvesters – attempt to solve this issue. Among other techniques advances in nanotechnology will provide new chances in this area.

On the computing side proper recognition of what is currently going on in the environment, i.e. high-level though on-line interpretation of raw sensor readings, is one of the major challenges.

Social aspects and security and privacy challenges are besides these technical questions major research topics in the field of wearable and contextual computing.

1.2 Wearable and contextual computing

As indicated in the previous section, *wearable computers* and *wearable computing* are widely fostered by an increasing demand of or rather request for permanently available personalized computing power.

Mobile computing

Mobile and portable devices can satisfy this need only to a certain extent, by making a desktop-like computer available to the mobile user. These devices are small scale versions of ordinary desktop computers, using quite similar interaction concepts, e.g. a keyboard, a pointing device, menu-driven navigation. Thus, any

interaction with such mobile devices demands full user attention.

Wearable computing

The wearable computing [Man97] concept heads into other directions. It investigates and fosters unobtrusive interaction [Rek01] between the user and non-intrusive [Che08] and pro-actively [Ten00] behaving computing devices. The ultimate goal is to let the user benefit from a permanently available, ubiquitous [Wei99] computing resource in a hands-free [HS95] and non-attentive manner.

Contextual Computing

To fulfill this requirement the computing device needs to be aware of the user's state and the state of the environment. In general this is referred to as *context sensing* and *context awareness* [ST94, SAW94, Sch02]. In the literature various definitions and interpretations of context, context sensing and context awareness can be found. Schilit *et al.* e.g. state in [SAW94]

> *Context encompasses more than just the user's location, because other things of interest are also mobile and changing. Context includes lighting, noise level, network connectivity, communication costs, communication bandwidth, and even the social situation [...].*

In general *context* can be defined as pieces of information that describe the state of the user and the user's environment. At best this description contains enough information to form the basis for a correct and well-behaving decision making process of the computing device.

As a consequence, *context awareness* is the ability of a computing device to solve its given tasks considering these pieces of contextual information, or as defined by Schilit *et al.* in [ST94]

> *Context-aware computing is the ability of a mobile user's applications to discover and react to changes in the environment they are situated in.*

The character of contextual information can vary. It can be of a physical nature, e.g. user location, user activity, gestures, health state, physical state; or of a more abstract nature, e.g. informational context, device availability, or even emotional state and user intentions.

Applications

The wearable and contextual approach can benefit several application areas. Frequently given examples (see also Section 1.5.4) are: *support of daily routines*, *maintenance and production assistance for mobile workers*, *assistance for mobile worker trainees*, *sports activity monitoring*, *pervasive health-care*, and *information services for rescue operators*. Their commonness is the idea to provide just the right information at just the right time [SSS08] or even to provide

> *the right information to the right person at the right place at the right time in the right language at the right level of abstraction. [Sie01]*

wearIT@work

The context of this work and the context of the presented case studies in particular is a large research project, the *wearIT@work* project[1] [LTGLH07, LHW07, Mau07]. The major goal of the project is to investigate and foster the applicability

[1] Sponsored by the European Union under contract EC IP 004216
http://www.wearitatwork.com/

of currently existing wearable computing techniques and concepts in industrial environments.

WearIT@work is focusing among other scenarios on activity recognition in industrial production and maintenance. The project envisions the development and test of activity recognition systems for the tracking of complex car assembly and aircraft maintenance tasks and designs and conducts test runs together with the end user partners, e.g. the car manufacturer Škoda and the aircraft manufacturer EADS.

To this end the project explores wearable, on-body, and context sensing technologies and aims to encourage wearable solutions within an industrial environment arguing that a wearable solution provides *pro-active information delivery* and *unobtrusive* access to information technology infrastructure, i.e. pro-actively deliver relevant information, e.g. display manual pages, display warnings in case of a missing step or an unsafe situation, record the progress of the procedure for later verification or supporting and assessing trainees' progress. By providing this information in pro-active, unobtrusive, and non-intrusive manner one can assure to leave the workers' hands free and allow freedom of motion, not distracting them from their primary task [Wit07, SRO+08].

Objective

Consequently, this work focuses on the recognition of so-called *manipulative hand gestures* [JKS98] in conjunction with the tracking of car assembly and aircraft manufacturing tasks. As stated above, the ultimate aim is to recognize what part of a given maintenance or assembly procedure is executed by the worker at any given point in time.

1.3 Problem definition

Manipulative hand gestures

By the very definition of an assembly or maintenance task relevant activities are given by the interaction of the users hands or arms with predefined parts of some machinery or other objects; we call such motion sequences *manipulative hand gestures*. As a simple everyday example consider changing a wheel in a car. It consists of taking off the wheel cover, loosening the screws on the wheel, taking the car-jack from the boot, attaching it to a proper location of the car, lifting the car by turning a lever on the car-jack, unscrewing the wheel, taking it off and so on.

In general any maintenance or assembly task can be decomposed into such a sequence of simple activities. Moreover, each single activity is characterized by a specific motion pattern of the hands or arms and a part of the machinery or objects – in our case a specific part of a large stationary machinery, e.g. the car body or a part of the aircraft – with which the user is interacting. Thus manipulative hand gestures are characterized by two factors:

- the motion of the hands or arms and
- the object that is being manipulated.

Note that even if a tool is used, the manipulation still targets a specific part of the machinery. Thus this work argues that in the envisioned scenarios with the focus on manipulative gestures knowledge of the user's hand location is an essential information because the user is interacting with objects at specific locations.

Moreover, the number of gestures that can be performed at a specific location is limited. Thus the search space of the gesture recognition system decreases in case of a well-recognized location context.

Furthermore, this work aims at a specific recognition problem: *the spotting of sporadic hand gesture events within a continuous data stream*. Spotting aims at *retrieving the starts and stops of meaningful segments* in a time series. To retrieve an accurate recognition result, it is often required that only the data stream containing *exactly* the gesture of interest is presented to the algorithm. Therefore the spotting step is an essential or even crucial part in many activity and gesture recognition problems.

Gesture spotting

A typical gesture spotting task within an industrial scenario is defined by a moderate set (in the range of ten to 100) of user actions which the system needs to identify in a continuous stream of data. Between the relevant actions the workers could perform arbitrary other activities; which means that a large body of arbitrary non-relevant actions are mixed randomly with relevant gestures. We will refer to these non-relevant events as *NULL class* events.

The NULL class

Spotting actions based on on-body sensor data is known to be a difficult problem that has so far not been solved satisfactorily. In general the major challenges in the field of spotting and recognition of manipulative hand gestures by means of on-body sensing are:

Problem description

- ✗ *Ambiguity of individual sensor signals:* In general most activities cannot be unambiguously characterized with a single on-body sensing modality. For example, since many activities are associated with characteristic arm motions, motion sensors on the wrist have been widely used. However, many activities contain similar motions. In fact, people tend to move their arms a lot so that any motion is likely to occur randomly with a connection to a specific motion.
- ✗ *Measurement of pseudo-motions:* Body-mounted motion sensors are prone to deceptive measurement; consider e.g. slipping sensors or a sensor housing accidentally touching an object. Due to this and the previous point, distinguishing gestures based on very similar motion patterns by means of motion sensors alone is not feasible.
- ✗ *Mixed motion and orientation information:* In addition, acceleration sensors measure a mixture of rotation and translation information, i.e. the sensor signal contains a mixture of motion and orientation information that is difficult to separate with simple sensor setups. Moreover these sensors provide no information about the palm and finger activities.
- ✗ *High inter-user variability:* People tend to perform activities and gestures in a very individual way, which results in highly user-specific motion patterns. What is more, these patterns may also vary over time, i.e. users perform activities differently depending on various factors like grade of tiredness, mood, stress factor and others.
- ✗ *High variability in event length:* Human actions can significantly vary in length. This is true both within a certain activity class and between different classes. As a consequence obvious techniques such as fixed-size-sliding-windows and correlations are often not applicable.
- ✗ *Lack of NULL class models:* In activity spotting the NULL class can be

vaguely described as *all possible human actions other than the ones belonging to the set we need to spot*. Evidently, given the complexity of human actions, useful models are difficult to be derived. Thus the spotting system needs to work with absolute thresholds for the similarity to the individual relevant classes rather than with relative similarity between relevant classes and the NULL class.

- ✗ *NULL class size:* In many applications the relevant actions comprise only a small amount of the overall measurement time. In industrial production scenarios this fraction may be quite high, but in other scenarios, such as monitoring household activities and daily life routines, this fraction may go down to a couple of percent. This means that the system needs to be highly selective – i.e. it needs to operate with high precision – to avoid a prohibitive number of insertion errors.

1.4 Approach

Location based spotting

Tracking of the user's hand location seems to be a promising approach to solve the spotting problem at least for the investigated scenario. In our case a manipulative gesture typically starts the moment the user's hand reaches a specific part, i.e. a specific position at the machinery; it should be finished, the moment the hand is moving away. Thus the location can be a strong indicator for the starts and stops of a manipulative gesture.

FSRs

We also add a novel activity sensing technique: force sensitive resistors (FSRs) for the purpose of arm muscle monitoring. Motion sensors provide a significant amount of activity information but accelerometers tend to contain a mixture of motion and orientation information that is difficult to separate with simple sensor setups. In addition they provide no information about the palm and finger activity. Due to the fact that palm and finger motions are driven by muscles in the forearm, resulting in significant contraction and relaxation phases, muscles significantly change their shape during a certain gesture, which in turn results in mechanical pressure being applied to the sensors, when mounted on the forearm.

FSRs have already been introduced to the wearable computing community. Still there are some open questions concerning the sensing hardware itself, e.g. when applying them for muscle activity monitoring. Hence this work will also deal with FSR hardware questions and suggest a new wearable hardware platform for distributed, wireless, multi-channel FSR sensing with linear characteristics.

User-independence

In addition, this work focuses on user-independence. Current concepts and systems for gesture recognition based on motion sensing are often highly user-dependent because of the individual ways humans perform gestures and activities. Persuasive recognition results can only be achieved in highly experimental setups, i.e. the subject is exactly instructed how to behave and how to perform a gesture or an activity. Thus real-life variability of human gestures often cannot be reproduced in such experiments. Variability in the way gestures are performed remains a major challenge; and the state-of-the-art approaches to get a grip on it, are:

- ✗ recording a training set with a representative number of gesture instances,
- ✗ recording a training set with a representative number of subjects.

- ✗ providing additional information, i.e. contextual information to the recognition system.

Additional information about the user's context should obviously increase the performance of a recognition system; but with the obvious drawback of increasing the complexity of the sensing system. However, the addition of *location context* information should outperform any other contextual information source, due to its user-independent nature: location information is evidently non-sensitive concerning the different ways a gesture can be performed.

Modularity

In addition to the spotting approach this work aims at suggesting an approach to solve the continuous gesture recognition problem in a modular manner, i.e. the suggested approach aims at flexibility concerning the number of gestures and concerning the sensing modalities and algorithms. Achieving modularity of the recognition system with respect to sensors and algorithms allows to add, remove or exchange a sensor modality or an additional processing step without impacting on the rest of the recognition system. Modularity with respect to gesture classes is achieved by means of a recognition architecture with independent spotting processes for each gesture. Thus new activities can be added without impact on the recognition systems for the old ones. Also, sensors and algorithms used for spotting can be selected and fine-tuned separately for each gesture.

Information fusion

As the modular recognition architectures proposed in this thesis result in highly parallel recognition result streams, this work also investigates how information from the individual gesture spotting processes can be consistently and efficiently combined into a single recognition result. Thus different strategies for information fusion will be designed and evaluated.

1.5 Related work

1.5.1 Activity and gesture recognition

The main approaches to activity recognition and recognition of manipulative gestures are video analysis, augmentation of the environment and wearable and body-worn sensors. These sensing approaches are neither mutually exclusive nor can one be said to be generally superior. Instead, the choice of a method or method combination depends on a specific application.

1.5.1.1 Vision based activity and gesture recognition

Vision based activity recognition

In the field of video analysis Starner *et al.* [SSP98] work e.g. with probabilistic object recognition methods to recognize non-manipulative hand gestures recorded with body-worn cameras. A Hidden Markov Model (HMM) is used to assign sequences of thus derived probabilities to either one out of two trained gesture classes. Non-relevant gestures are included into training to model the NULL class. Yamato *et al.* [YOI92] work with HMMs trained on discretized image sequence feature vectors to recognize sport activities.

Vision based ASL recognition

Vogler and Metaxas [VM98] present a work aiming at vision based recognition of American sign language. They are using model based tracking of human body parts. The thus derived arm orientations are used as input to HMMs modeling individual signs. The arm motions are pre-segmented into time segments containing elementary

motion subsegments by searching for minima in the velocity vector. Additional work on vision based recognition of American sign language was also presented in [LX96, SWP98, JS04, BHP+06, NSSKAA06].

Vision based interaction

Störring *et al.* [SMLG04] use skin color segmentation for the recognition of pointing hand gestures. The images are captured using a head mounted camera. An on-body stereo camera is presented by de la Hamette and Tröster [dlHT08]. By means of recognizing finger postures the authors create a stereo vision based pointing and input device. Hasanuzzaman *et al.* [HAZ+04] present vision-based gesture recognition for human robot interaction.

1.5.1.2 Environmental sensing approaches

Everyday activities

Augmentation of the environment is extensively used in the area of everyday activities and industrial maintenance tasks. Stikic *et al.* [SHvLS08] are comprising acceleration sensors and radio-frequency identification (RFID) tags for the use of monitoring everyday household activities. Patterson *et al.* [PFKP05] solely use a RFID reader integrated into a glove and RFID tagged household objects for the recognition of everyday activities.

Assembly and maintenance

Switches incorporated in manipulated objects are used in [AMS02] to guide users when assembling furniture. A combination of body-worn and force sensing resistors used as environmental sensors for the use of progress recognition in a car assembly task is studied by Stiefmeier *et al.* [SLR+06]. RFIDs are also used by Nicolai *et al.* [NSKW05] to tag an aircraft cabin area aiming at non-intrusive assistance in aircraft maintenance. A body-worn RFID scanner is used for user location identification and thus allows conclusions on the worker's activity.

1.5.1.3 On-body approaches

Basic activities

In the field of body-worn sensing of human activities primarily *acceleration* sensors and *gyroscopes* attached to the hands and arms have been shown to be a promising sensing modality. Randell and Muller [RM00] are investigating the use of a single body-worn bi-axial acceleration sensor for the recognition of basic activities like *sitting, walking,* etc. They use a set of four features – RMS and integration in fixed window size – as input to a preliminary trained neural network. Seon-Woo and Mase [LM02] also use on-body acceleration and gyroscope sensors. In such a way the authors set up a dead-reckoning system which is used to determine basic activities like *sitting, walking,* and *standing.* The same activities are investigated by Mäntyjärvi *et al.* [MHS01]. They are using two three-dimensional acceleration sensors attached to the hips of the user. Features are extracted using principle component analysis (PCA) and independent component analysis (ICA) methods and a subsequent wavelet transform. The classification stage is using neural network techniques. Thigh-mounted acceleration sensors are used by Laerhoven and Cakmakci [LC00] to recognize similar activities. The recognition of basic activities such as *sitting and standing* plus three basic hand gestures namely *shaking hands, writing on a white-board, and writing on a keyboard* are investigated by Kern *et al.* [KSS03]. They investigate the recognition performance of different on-body positions of a three-dimensional acceleration sensor. Bao and Intille [BI04] work with multiple bi-axial acceleration sensors to recognize everyday activities.

Gestures

More complex gestures are being investigated by Stiefmeier *et al.* [Sti08, SRO+08, SRT07b, SRT07a]. The authors investigate the use of upper body motion tracking

by means of a set of inertial measurement units for recognition of manipulative gestures. As shown by Ward *et al.* [WLT06, War06] and [LWJ+04, SLP+03] body-worn microphones and subsequent sound analysis can be useful for recognition of manipulative gestures. In particular it provides information about those tasks that actually cause a characteristic sound pattern but it has got limitations in noisy environments, i.e. in a typical industrial assembly environment.

1.5.1.4 On-body approaches using pressure sensors

Gait analysis

Junker *et al.* [Jun05, JLT04b] use force sensing resistors (FSRs) to recognize basic walking modes. Bamber *et al.* [BBS+08] present a wireless wearable sensing platform that can be integrated in an ordinary shoe for the use of gait analysis under real-life conditions. The platform comprises among other sensing hardware four FSRs.

Muscular activity monitoring

Kreil *et al.* [KOL08] present FSRs integrated into cycling tights for wearable sport activity monitoring. Lukowicz *et al.* [LHSS06] demonstrate the general feasibility of using FSRs to monitor leg muscle activity. [AJL+06] shows that different arm actions such as holding a heavy object or making a fist produce distinct FSR signals.

Textile pressure sensing

Meyer *et al.* [MLT06] present a capacitive pressure sensor that can be integrated into textiles.

1.5.1.5 Continuous activity and gesture recognition

Independent of the sensing modalities, the recognition from a continuous, unsegmented data stream is known to be a difficult problem. It is particularly difficult in the so-called spotting scenario (see also Section 1.3) where the relevant activities are mixed with a large number of arbitrary other actions. Any gesture or activity recognition approach has to deal with that issue somehow. And in fact much work has been done on the spotting problem in the activity and gesture recognition area.

Vision based

Lee and Kim [LK98] describe continuous hand gesture spotting based on video input. They use a threshold model using HMMs to segment motion data. Yoon *et al.* [YSBY01] present a work on continuous gesture recognition from video data. The recognition and tracking of the hand is based on skin color information. This hand trajectory is furthermore used to continuously recognize hand gestures based on both hand motion and hand location information. Activity segmentation in video was also presented by Brand and Kettnaker [BK00].

Environmental sensors

Continuous recognition results are presented by Patterson [PFKP05] where RFIDs are used to track which household objects the user interacts with.

Multi-modal on-body approaches

Sound based spotting methods are used e.g. by Ward *et al.* [WLT06, War06]. Amft *et al.* [AJT05] describe a detection system for eating gestures using a multi-modal on-body sensor approach. Activity spotting is done by means of the SWAB algorithm. Lester *et al.* [LCB06] describe an activity recognition system based on a multi-modal sensor augmented cellphone platform.

On-body motion sensors

A novel activity spotting method using motion sensors based on so-called closed motions is investigated by Junker *et al.* [JLT04a, Jun05]. Zinnen *et al.* [ZLS07] introduce a method for spotting hand gestures based on characteristic start and stop patterns in the accelerometer signals.

Stiefmeier *et al.* [Sti08, SRO+08, SRT07b, SRT07a] present an approach to continuously recognize manipulative hand gestures in industrial environments. It is based on a discretization of the motion trajectories of the lower arms or the hands, resulting in a motion trajectory alphabet. Spotting and recognition are then

based on a string matching algorithm trained for this motion trajectory alphabet. Minnen *et al.* [MIES07] also work with discretization of time series signals by means of using the SAX algorithm proposed by Lin *et al.* [LKLC03, LWL07]. They furthermore propose a method for unsupervised learning of multivariate discretized time series patterns. The authors test the proposed method on data recorded with wrist-worn motion sensors during different dumbbell exercises.

1.5.1.6 Conclusion

Activity and gesture recognition is widely investigated both by computer vision researchers and by the wearable computing community. Researchers focus primarily on the continuous case as pre-segmented activities are effectively non-existing in real-life. Nevertheless a general, user-independent solution for the spotting of sporadic hand gestures is not available.

FSRs are well-known in the wearable computing community. Nonetheless the monitoring of muscular activity for the use of recognizing complex gestures has so far not been attempted.

1.5.2 Location awareness

Together with activity and gesture recognition location awareness is probably the most explored issue in the *context aware* computing domain.

The use of an ultrasonic location systems for context aware computing has been investigated in [HHS+02, WJH97, War98]. Performance of ultrasonic indoor tracking, using the Cricket system [PCB00, PMBT01, BP03] has been investigated by [SBGP04].

Helal *et al.* [HWL+03] e.g. investigate surveillance of and support for physically challenged persons and elderly people, through *remote monitoring* and *attention capture* or an automated guide for blind people. Liao *et al.* [LFK05] describe an activity recognition approach based on GPS data. The activity models are defined in Markov networks manner. The activities of interest are of the type *dining out, being at home, at work, shopping, visiting.* The approach is evaluated in an experimental manner using data, annotated by the users themselves.

Ashbrook *et al.* [AS03, AS02] describe an approach to model and predict user location for various applications, which can be divided into two categories: *single-user* and *collaborative* applications. Such applications are: location prediction dependent reminder, location prediction dependent resource planning, meeting suggestions and location dependent social networking. Nord *et al.* [NSP02] aim at a friend finder application.

1.5.2.1 Conclusion

Location awareness is a widely considered aspect in context aware computing. Nonetheless the detailed tracking of hand locations by means of a solely wearable sensing approach for the use of activity recognition has so far not been attempted. In what manner hand location tracking can enhance activity and gesture recognition and which enhancements can be expected are thus open questions.

1.5.3 Indoor positioning systems

The following sections give a short summary on location systems and list available implementations and techniques both commercial and non-commercial with the focus on indoor applicability. More general overviews of automatic location sensing techniques in the field of wearable computing can be found in [HB01, Hig03, FHK+03, Tau02].

1.5.3.1 Ultra-wide-band

Ubisense is a commercial indoor location system using ultra-wide-band (UWB) radio. The UWB technology has the following advantages:

- ✗ it is independent of line of sight,
- ✗ the range of a device can cover tens of meters,

The system is based on *time-difference-of-arrival* and *angle-of-arrival* measurements between a network of base stations with known and static positions and moving tags attached to the object under tracking control. An additional radio-frequency (RF) channel provides wireless communication between base stations and tags. The Ubisense framework scales for a large number of base stations and moving tags – which is typically a major advantage of commercial location systems – and thus can cover a huge indoor area such as a hospital or a factory ground.

1.5.3.2 RFID location systems

SpotON [HVBW01, HWB00] and LANDMARC [JLP06, NLLP03] are doing indoor localization based on RFID tags. Basically LANDMARC is working with stationary readers and stationary reference tags. The readers cycle through eight different ranges and thus result in an eightfold *received signal strength indication* (RSSI) vector for each tag in range. Comparing the results of the moving devices with those of the reference devices using a special distance measure results in the final location estimation.

1.5.3.3 RSSI based location techniques

There are several implementations using this technique: Ekahau/NSC is using a WLAN network to track tags equipped with WLAN access cards. It works on the IEEE 802.11 wireless network standard. The localization is based on signal strength measurements. These measurements are mapped to locations, i.e. each location is assigned a so-called measurement fingerprint. Commercial systems similar to Ekahau are PanGo and AeroScout. Non-commercial implementations working similar to Ekahau are RADAR, see e.g. [BP00], WILMA [BBC+03] and NeBULa [FMPS06].

Nerguizian *et al.* [NDA04, NDA06] propose an approach that does not just learn the RSSI fingerprints of certain locations and combines them via optimization techniques to estimate a location, but enhances the RSSI approach by means of applying the measurements to an artificial neural network.

Chipcon CC2431: Texas Instruments and Chipcon have developed an *on chip* location engine, see e.g. Chipcon CC2431. This location engine implements a statistical location estimation algorithm that uses received signal strength values from known reference nodes, such as stationary infrastructure nodes or other mobile

neighbor nodes that use the same location engine. Moreover, nodes using this engine parallelize the computational effort of the localization algorithm.

The LANDMARC system described in Section 1.5.3.2 also belongs to this category.

1.5.3.4 Ultrasonic localization

Ultrasonic indoor positioning systems (UPS) are widely used for indoor location [MRC05, SBGP04] and relative positioning [HKG+05].

Sonitor provides an ultrasonic based indoor location system that tracks people on a room-level basis. It scales for a large number of tags and base stations. Hexamite, the Bat system [ACH+01] and Cricket [Pri05] are the traditional ultrasonic positioning systems. The implementations vary but in general these systems rely on time-of-flight measurements between base stations with known positions and moving tags. Hazas *et al.* [HKG+05] use ultrasonic relative positioning within a wireless sensor network. Hazas *et al.* [HW02, HH06] also investigate a broadband ultrasonic system that outperforms the widely used narrow-band UPS significantly, though it also increases the complexity of the sensor system. The presented broadband approach enhances the above mentioned narrow-band ultrasonic systems by means of:

- ✗ eliminated interference problems,
- ✗ increased update rates and thus low latency positioning,
- ✗ information encoding via ultrasound.

1.5.3.5 Powerline positioning

Patel *et al.* suggest in [PTA06] a fingerprint based location technique using Powerline infrastructure. The location approach works on sub-room level.

1.5.3.6 Infrared cameras

The infrared (IR) cameras of the Lukotronic system[2] use active IR markers and three IR row cameras to track these active markers in three-dimensional space. The calculation of the resulting point is based on triangulation. The advantages are great resolution (the accuracy depends on the elaborateness of the setup) and high update rate. The disadvantages are a small angle of view (one triple camera system is usable for a single standing person). This could be enhanced by using couples of triple camera systems. Due to the small operation radius and despite the high accuracy and the high update rate, it is not ideal for user tracking but it can be used for tracking of body parts, e.g. in sport medicine or virtual reality.

1.5.3.7 Solar cells

Randall *et al.* [RAT04] present an indoor navigation approach based on ceiling lighting and on-body solar cells.

1.5.3.8 Vision based user localization

Bauer *et al.* [BL08] describe an approach on how vision based object detection can be combined with on-body motion sensors for a sub-room level user localization. A ceiling-mounted fish-eye camera is used to identify and track different moving objects and to identify the user's status: walking, sitting, standing, sitting down, standing

[2] http://www.lukotronic.com

up. A mobile phone with an integrated motion sensor assigned to a specific user is doing the same motion classification in parallel. Matching these motion streams assigns each recognized moving object a unique ID; thus each user is assigned a specific location.

Starner *et al.* [SSP98] present a room-level indoor location system based on two head-mounted cameras. HMMs are trained for each location of interest. The observations presented to the HMMs are color and lightning features extracted from designated areas in each sampled frame.

de la Hamette *et al.* [dlHT08] present a wearable stereo camera system for relative hand tracking as a novel human-computer-interaction (HCI) concept.

1.5.3.9 Microphone arrays

Microphone arrays are proven to be useful for speaker localization. Adcock and Silverman [BAS97, BAS96] present a location technique based on microphone arrays – each microphone has assigned a static and known position – and time difference of arrival estimations.

1.5.3.10 Inertial measurement units

Inertial measurements units (IMU) comprising three-dimensional acceleration and three-dimensional gyroscopes (see also Section 4.2) are usually used for ship and aeroplane navigation, see e.g. [BH97]. Due to drift errors being integrated over time, such a navigation system always needs an absolute aiding source. For outdoor navigation GNSS, i.e. GPS, is usually used as an absolute aiding source whereas in indoor navigation either a UPS or a magnetic sensor can be used as a reference system, see e.g. [RDM03, Fox05, WH08].

1.5.3.11 Interaction

This refers to the huge field of tracking people by logging their interaction with objects that have a known position. Locating a person due to his/her credit card and bank customer card use belongs to this category.

Another example is the monitoring of power consumption. The German electricity provider Yello Strom[3] is going to launch a service in 2009 that will allow the provider to monitor the electricity consumption per power outlet per second. A service that can also be used to track people and their activities – particularly in case each power outlet is assigned a specific location and a specific device, either manually or using a *dSID-Chip*, as done by Yello Strom. dSID-Chips provide unique identification for electronic devices using the communication standard *digitalSTROM*[4]. [5]

[3] http://www.yellostrom.de/

[4] http://www.digitalstrom.org/

[5] The reason for Yello Strom being awarded the *BigBrotherAward 2008 in the technical category* by the data privacy protection watch organization *FoeBuD e.V.* is that – as stated on http://www.bigbrotherawards.de/2008/.tec – this technology might lead to a detailed surveillance of activities in the home with resolution of a single second. Though the digitalStrom consortium states that the physical principals of their technology do not allow surveillance from outside the in-house electricity network and thus leaves the decision to the customer, who is allowed access to the data, this will probably be circumvented by the Yello Strom electricity meter that will collect the data and transmit it to the provider every quarter of an hour.

1.5.3.12 Tracking of body parts

With regard to tracking body parts several different approaches can be taken. In biomechanics applications such as high performance sports or rehabilitation magnetic systems (e.g. Flock of Birds by ascension[6]) are widely used. Such systems use a stationary source of a predefined magnetic field to track body-mounted magnetic sensors. The main problem with magnetic tracking is that it is easily disturbed by metal objects, which are common in industrial environments.

One alternative is the use of optical markers (often IR markers, see e.g. the Lukotronic system[7]) together with appropriate cameras. Here problems like background lighting, for IR systems in particular, and occlusion need to be dealt with. The main disadvantage of both magnetic and optical tracking systems is that they are optimized for ultra-high spatial resolution and thus expensive and bulky.

Roetenberg [Roe06] is combining inertial measurement units with on-body magnetic sensing and vision based marker tracking for high resolution tracking of body parts.

1.5.3.13 Conclusion

Various solutions to the indoor tracking problem are proposed. Little are applicable for the detailed tracking of body parts. Many indoor tracking systems provide a spatial resolution of room-level or sub-room level. Others provide a temporal resolution that is too small to reflect the dynamics of human hand motion. Some vision and IR based systems provide both decent spatial and temporal resolutions and are thus widely used for detailed recordings of human motions, e.g. in the movie industry. In our contemplated scenario large objects in the line of sight – in most cases the object that is being manipulated – prohibit the employment of such systems.

1.5.4 Context-aware and wearable computing applications

Various applications and application areas are being investigated by researchers. This section lists frequently contemplated application scenarios.

1.5.4.1 Maintenance and production assistance

Well-suited sensors both in the environment as well as fixed on the clothing and equipment of the (mobile) worker are monitoring the progress of a specific activity with the goal to give the worker advice, display warnings in case of critical situations, pro-actively display manual pages, and optimize industrial workflows in general, see e.g. [LABT04, LWJ+04, LTGLH07, SRO+08].

The benefit of wearable and contextual computing to assembly and maintenance tasks in particular is approved by e.g. Randell [Ran05]. In fact, there are various case studies investigated that – though using different sensing approaches – deal with such an application scenario.

Moreover, a context-aware system can also give hints and advice to a trainee doing test-runs for a specific production workflow [SRO+08].

In the field of wearable pro-active maintenance support [WNK06, WLKK06] present a data glove for maintenance tasks; the glove evolves from earlier findings

[6] http://www.ascension-tech.com/realtime/RTflockofBIRDS.php
[7] http://www.lukotronic.com/

in the Winspect project [BNSS01], a wearable system for crane maintenance comprising among other sensors a RFID scanner to follow the progress of a maintenance task.

In [AMS02] it is shown how pressure and tilt sensors integrated in tools and components can be used to track a furniture assembly task. The system recognizes single steps within the assembly process and thus can pro-actively guide the process.

Ward *et al.* [WLTS06] investigate the combination of acceleration sensors and body-worn microphones to spot and recognize manipulative hand gestures in a wood workshop task.

1.5.4.2 Support for daily routines

Daily routines can be supported via context-awareness, e.g. learning activities [YJ07]. Stikic *et al.* [SHvLS08] are comprising acceleration sensors and RFID tags for the monitoring of everyday household activities to facilitate care of the elderly. In such a way wearable computing may also enable applications such as assisted living [LKT04]. A personal assistant may also *just in time* and *dependent on the user's location* manage reminders for upcoming appointments and *to-do list* items [AS03].

1.5.4.3 Sport activities

Wijnalda *et al.* [WPVS05] propose a wearable architecture capable of selecting and playing music to motivate the user while doing recreational exercises. The music is selected according to congruity or incongruity of the current exercise effort and the previously defined exercise goals.

Kreil *et al.* [KOL08] present FSRs integrated into cycling tights for wearable sport activity monitoring giving on-line feedback for runners and bicyclists.

1.5.4.4 Pervasive health-care

Considerable research is done in the field of patient monitoring by means of wearable sensing and wearable computing; e.g. Amft [Amf08] presents an on-body sensor system for detecting eating and drinking behavior aiming at context-aware diet monitoring and assistance. Paradiso [Par03] presents a wearable setup to monitor vital signs aiming at remote and automatic assistance for cardiopathic patients during their rehabilitation.

Another research field is dealing with novel information access and communication methods in hospitals. Adamer *et al.* [ABK+08] present a wearable system for unobtrusive information access for the medical staff in hospitals. Horvitz and Shwe [HS95] describe a wearable, hands-free system providing medical staff access to a decision-theoretic diagnostic system in emergency situations.

1.5.4.5 Information services for rescue operations

Information at the right time at the right place with respect to the contextual situation of the user is essential, in life critical services such as rescue operations in particular. In this application domain wearable computing research also aims at enhanced inter-operator communication methods [JCH+04] and new navigation concepts [KRG+07, Kla07].

1.5.4.6 Conclusion

Various application areas for contextual and wearable computing have been fostered by researchers. It remains to be seen which application areas can benefit in real-life

from these concepts. That depends also on the relation of the actual benefit for the user to the grade of obtrusiveness and intrusiveness that can be achieved by the final implementation of such a system.

1.6 Objectives of the thesis

To support mobile workers following the wearable demands the knowledge of the workers activity is essential. In general the aim of this thesis is to investigate the aptitude of on-body sensors for activity recognition and the spotting of manipulative hand gestures.

As a state-of-the-art approach we use orientation sensors to measure the worker's upper body motions and further recognize the gestures the worker is performing.

The general questions are: Can the recognition of manipulative gestures based on motion sensors be enhanced through additional sensor sources? What kind of enhancements can be expected? In more detail we aim at investigating the following questions:

- ✗ The set of gestures a person usually performs on a specific place is limited, in the production and maintenance scenario we have in mind in particular. Does thus the knowledge of the worker's location context or even the hand location give essential hints to the activity recognition process?
- ✗ Does the addition of location context introduce more user-independence to the recognition system?
- ✗ Can knowledge of the hand location help solving the spotting problem?
- ✗ An additional information source for motion can be the muscular activity. Are force sensitive resistors (FSRs) a feasible way to recognize gestures based on the information about muscular activity of the lower arms?
- ✗ Can FSR based muscular activity monitoring further enhance the motion based recognition approach?
- ✗ How can these different kinds of contextual information – *motion, location,* and *muscular activity* – be combined?
- ✗ Even within a single application different activities are likely to have different sensing modalities that characterize them best. The same is true for the feature sets. Does – as a consequence – a multi-modal sensing approach increase the performance of an activity recognition system?
- ✗ Modularity is a key feature of flexible recognition systems. It enables flexibility concerning activity classes, sensors, and processing and recognition algorithms. Moreover the overall performance of a truly modular setup is immune to addition and removal of activities, sensors and algorithms. How can a recognition system be set up to achieve such modularity? What additional enhancements can be achieved thereby?

Another major challenge within context recognition is the sensing process itself. In this context we will investigate the following questions:

- ✗ Can a slow sampling location system be enhanced by means of fast sampling inertial motion sensors in terms of enhanced dynamics in order to comprise a decent hand tracking indoor location system?

- How can a decent FSR system be set up to be both wearable and applicable for measuring muscular activities?

1.7 Contributions

The main contributions of this thesis with respect to the questions raised in Section 1.6 are:

- We test a Kalman filter based method to blend absolute position readings with orientation readings based on inertial measurements. A filter structure is suggested that allows up-sampling of slow absolute position readings and thus introduces higher dynamics to the position estimations. In such a way the position measurement series is made aware of wrist *motions* in addition to the wrist *position*.
- We suggest a location based gesture spotting and recognition approach. To this end various methods to model location classes used in the spotting and recognition stages as well as different location distance measures are suggested and evaluated.
- We develop a wearable hardware architecture for lower arm muscular activity measurements. The sensing hardware is designed to have a high dynamic range. In contrast to preliminary attempts the new design makes hardware calibration unnecessary.
- We suggest a modular and multi-modal recognition system; modular with respect to both sensors and algorithms and to gesture classes. This means that adding or removing a sensor modality or an additional algorithm has little impact on the rest of the recognition system.

  Our recognition architecture has an independent spotting process for each activity. New activities can be added without impact on the recognition systems for the old ones. Sensors and algorithms used for spotting can also be selected and fine-tuned separately for each single activity.
- The methods are evaluated on different case studies. All case studies are recorded in a laboratory-like environment but are designed to be as realistic as possible and necessary.

1.8 Outline

The thesis is organized as follows (for a graphical overview refer to Figure 2.5):

- *Chapter 2* gives a brief overview of our basic approach how to achieve modularity within an activity spotting and recognition system. The chapter summarizes the basic concepts of the class-wise approach that will be used throughout the thesis for the presented spotting and continuous recognition tasks.
- *Chapter 3* gives a summary of the three presented case studies. The case studies are inspired by the mobile worker scenario, more precisely the maintenance and production assembly scenario. As a consequence, the case

studies investigate a bicycle maintenance task and a car assembly conveyor belt quality assurance task.

An additional case study – giving a multimedia talk – is designed to test the limits concerning gesture recognition based on monitoring of lower arm muscular activities by means of a FSR system.

- *Chapter 4* outlines some background information about indoor location techniques. The requirements to the location systems concerning our specific application scenarios in mind, the chapter as a consequence gives reasons for specific indoor location system choices and describes the realization and the setup of the finally used sensor systems.

  Furthermore, the chapter outlines the position estimation methods used. The methods are intended to be able to handle the great presence of erroneous distance readings due to the limitations of the ultrasonic positioning technique.

  Moreover, the chapter describes our approach to fuse motion and orientation readings with ultrasonic based position estimations, resulting in a hand trajectory with a far better dynamic response than the mere ultrasonic positioning approach is able to provide.

- *Chapter 5* defines various probabilistic methods in order to model hand locations. The training of these location models can be done either in a supervised or in a semi-supervised manner. Appropriate distance measures are defined for each location modeling approach.

  The methods are tested and evaluated on the bicycle maintenance test scenario. The chapter gives detailed evaluation results for pre-segmented activities in order to foster the combination of hand location and hand orientation for the use of gesture recognition.

- *Chapter 6* presents in detail our approach on combining location and motion information for activity spotting and recognition.

  The proposed approach comprises location based spotting enhanced by means of a location trajectory based spotting step. Finally a strategy for fusing intermediate, class-wise results is described. The methods are tested and evaluated on the bicycle maintenance test scenario.

  Finally, the chapter gives also a comparison with the approach applied in [SOJ+06] and contrasts results achieved by both methods.

- *Chapter 7* investigates the usefulness of muscular information of the lower arms. To this end an adequate sensing hardware comprising several FSRs is developed and implemented.

  We then systematically investigate the usability of the FSR system to recognize different manipulative gestures. The aim is to test the limits of the system, compare them to established sensing modalities, i.e. three-dimensional acceleration and gyroscopes, and establish the advantages of combining FSRs with other sensing modalities.

- *Chapter 8* describes an approach to real-life task tracking using a multi-modal, on-body sensor system. The specific example that we study will be quality inspection in car production. This task is composed of up to 20 activity classes such as checking gaps between parts of the chassis, opening and closing the hood and trunk, moving the driver's seat, and turning the steering wheel.

Most of these involve subtle and short movements and have a high degree of variability in the way they are performed.

To spot those actions nonetheless in a continuous data stream we use a wearable system composed of seven motion sensors, 16 FSRs for lower arm muscle monitoring and four UWB tags for tracking user position. We propose a recognition approach that deals separately with each activity class and then merges the results in a final reasoning step. This allows us to fine-tune the system parameters separately for each activity. It also means that the system can easily be extended to accommodate further activities.

In order to demonstrate the feasibility of our approach we present the results of a study with eight participants and a total of 2394 activities.

✗ *Chapter 9* concludes the thesis, gives a short summary of the achievements and a short outlook on future research questions.

Chapter

2

Modular multi-modal activity recognition*

This chapter gives a brief overview of our basic approach how to achieve modularity within an activity spotting and recognition system. The chapter summarizes the basic concepts of the class-wise approach that will be used throughout the thesis for the presented spotting and continuous recognition tasks.

*This chapter is partly based on reference [OSLT08].

2.1 Introduction

As described in Section 1.4 on page 6 this work aims at suggesting an approach to solve the continuous gesture recognition problem in a modular manner, i.e. the suggested approach aims at flexibility concerning the number of gestures and concerning the sensing modalities and algorithms. This chapter will give an overview of this approach and summarize its objectives and advantages.

In order to achieve a recognition system that addresses the following issues:

- ✗ decent recall and precision rates,
- ✗ recall and precision rates that are independent of the overall number of activity classes,
- ✗ independence of sensors or combination of sensors,
- ✗ user-independence, and
- ✗ independent parameterization for each individual activity class and thus a class-wise optimized recognition,

we propose to use a modular, multi-modal approach. Our approach has two major advantages. First, it allows to use for each class different sensors, features and algorithms depending on what best identifies the unique characteristics of the corresponding activity. This addresses the challenge of large heterogeneity inherent in activity spotting. For example, a specific gesture might be characterized best by a specific sound pattern, e.g. tightening a screw by means of an electric drill, or by a specific location, e.g. checking the fixation of a specific screw, or by location *and* sound, e.g. tightening a *specific* screw by means of an electric drill.

Second, it facilitates easy addition and removal of activities from the spotting system. The only part that needs to be modified when classes are added/removed or sensors/algorithms are exchanged is the final reasoning step that produces the union of the output of the individual spotting and recognition processes.

This approach also aims at a self configuring activity system. Individual processing stages are tuned to achieve the best possible performance for each individual class given a specific sensing modality. Assuming a sensing modality is exchanged or malfunctions the system might be trained to automatically select the best available alternative sensing modality for each individual activity class.

The obvious drawback of such a *late* fusion approach is of course the fact that it neglects the advantages of joint inference such are high performance on the one hand and a lower demand on the amount of training data on the other hand.

2.2 Modularity with respect to activities

We treat each activity as a separate event stream; thus for n activities we have n independent spotting processes (see also Figure 2.1). Each process is responsible for a single activity and determines where events corresponding to this activity occur in the data stream. The final output of the spotting system is the union of the outputs of the individual processes. By contrast most previously published work (including our own work) combines spotting with multi-class recognition from the start.

Independent class recognition enables easy addition and removal of classes. Moreover, the class-separation prevents that activity event hypotheses can delete

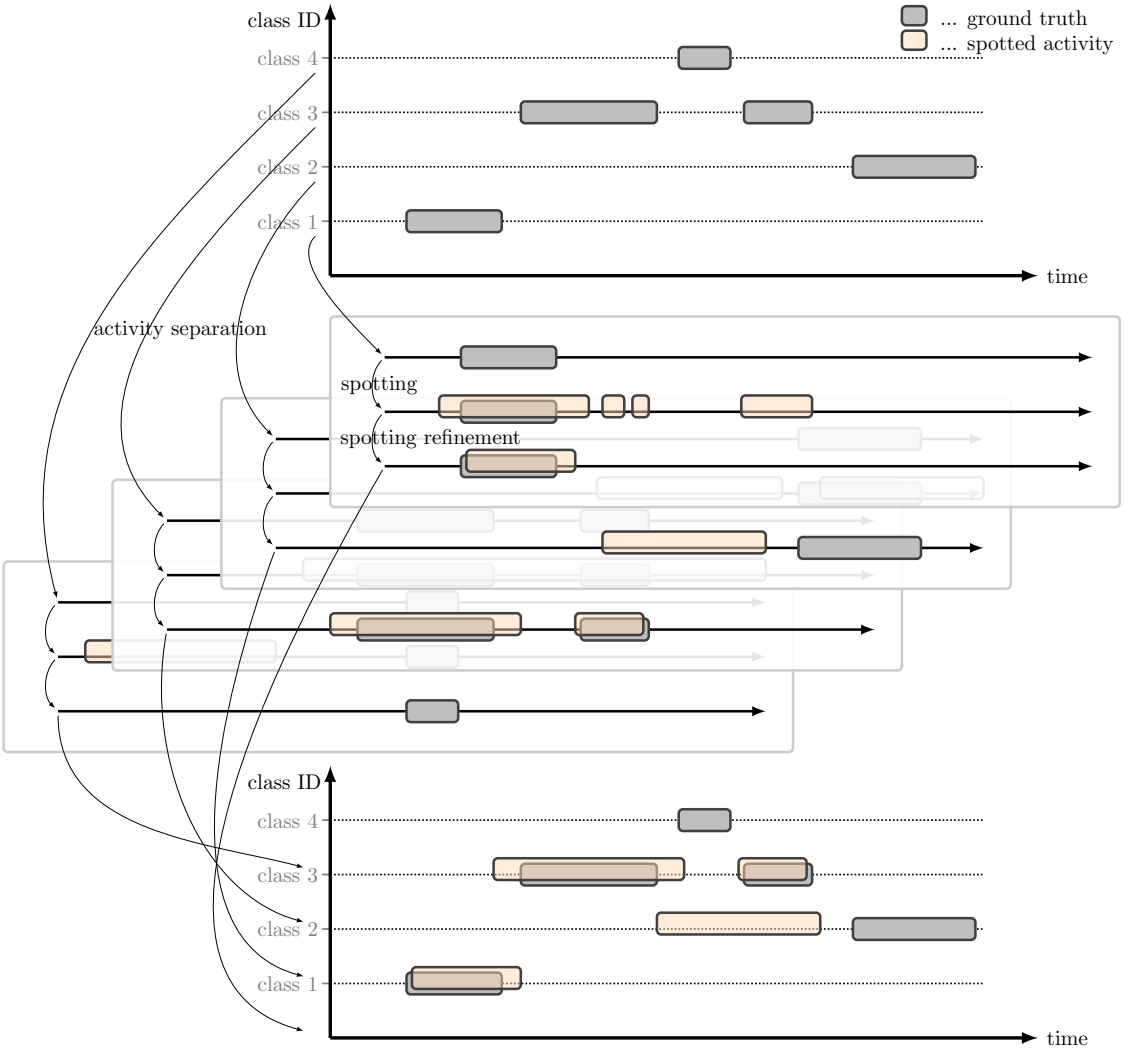

Figure 2.1
The figure schematically depicts the class-wise spotting approach.

each other, and thus the maximum information is passed on to the next processing stages. Additionally the class separation assures that individual performances are independent of the overall amount of activity classes. Thus individual classes can be added or removed without influencing the overall recognition system. And, what is more, each individual activity class can use the sensing modality that suits best and each activity class can be tuned individually and as good as possible. Hence this class-wise spotting and recognition approach aims at a class-dependent selection of sensors, features and algorithms with algorithms trained and parameterized for each class individually.

Evidently the class-wise processing approach results in concurrent prediction event streams with activity events that may have temporal overlaps. A final processing stage is needed to solve event prediction conflicts. This final processing step is trivial in case the individual processing steps achieve a decent precision, which is equivalent to a small amount of concurrent activity event predictions. This final reasoning step makes a classifier necessary that is based on an overall class-

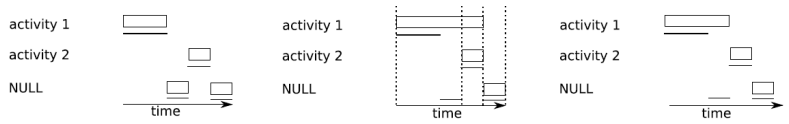

Figure 2.2
Merging individual spotting streams.

independent feature or rather confidence value. Thus this stage will be dependent on both the overall number of gesture classes and the available sensing modalities. In case the individual processing steps are not optimized to achieve high precision the overall performance will greatly depend on this final reasoning stage.

2.3 Merging activity event streams

The class-wise approach results in n separate outputs with n being the number of classes. If all the spotting processes were 100% correct, at any given time at most one stream could contain an activity while all the others would have to indicate the NULL class since the user performs only one activity at a time. In order to resolve this we apply different merge strategies, as explained in the following sections.

2.3.1 Approach 1

We propose a simple, efficient heuristic for the resolution of conflicts. It is based on dividing the output of all processes into what we call *invariant segments*. An invariant segment is defined as a time period during which the output of *none* of the spotting streams changes. This is illustrated in Figure 2.2 in the middle. New segments start when the activity 2 event starts and when activities 2 and 1 both stop. Note that the *invariant segments* partitioning is identical for all spotting processes and thus provides a natural, coarse grain partitioning of the output into partial events.

In the next step conflicts are identified concerning invariant segments. An invariant segment is said to contain a conflict, if more than one spotting process reports an activity in this segment. In Figure 2.2 segment 2 contains a conflict as both activity recognition processes report an event. Next we decide which activity to retain. To this end we first look at each activity reported in this segment and check if the corresponding event contains invariant segments without conflicts. If this is the case, removing this segment from the event would not cause it to be deleted. In Figure 2.2 this is the case for the activity 1 event which is conflict-free in segment 1, but not for activity 2. As a consequence, in case of a conflicting segment we retain the activity where the corresponding event contains no segments without conflicts. In Figure 2.2 this is activity 2. If there are no such activities in the segment, the choice is made randomly. The same is true, if there is more than one such activity in the invariant segment.

Obviously, the above heuristic does not always guarantee perfect conflict resolution. However, if the conflicts do involve many activity streams they are unlikely to produce deletions. The idea behind it is that the merge process matters more when conflicts are rare and do not involve many activity streams. This is the case when all the streams have good recognition performance. In this case it is important that the merge step does not introduce additional errors. If there are

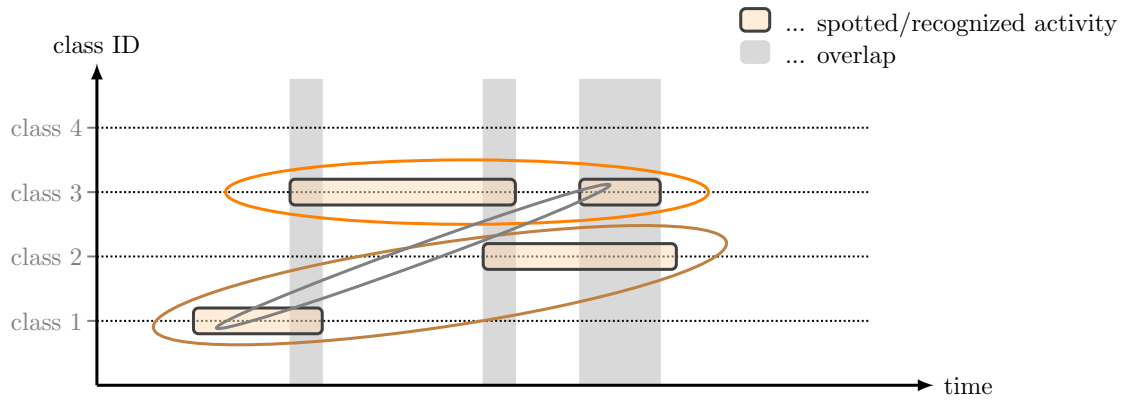

Figure 2.3
The figure depicts a possible approach to handle concurrent activity hypothesis. The three ellipses mark those event prediction sub-lists, that contain no prediction concurrency. The most probable sub-list according to a specific criteria is selected as a final prediction stream.

various and complex conflicts, the system performance is poor anyway. Thus it makes little sense to optimize the merge system for this case.

2.3.2 Approach 2

For this approach confidence values derived from preceding spotting and recognition steps are calculated. In case of a (local) concurrency we search for all possible sub-lists containing no concurrency (see also Figure 2.3) and decide in favor of the sub-list that contains the gesture segment with the highest confidence value. Evidently, this strategy will fail and cause a considerable amount of deletions in case of heavily nested concurrencies.

Before this strategy is applied, some *minor* concurrencies are pseudo-resolved, i.e. in case of a concurrency of two segments with the overlapping fraction of both segments being smaller than a certain threshold we shorten these segments by half of the respective overlapping fraction. In Figure 2.3 the first and the second overlap may be resolved in such a way – depending on the threshold. It is obvious that the third concurrency in the depicted example cannot and must not be resolved in such a simple way. For any given segment repeated pseudo-resolving is allowed while the deleted fraction on each side of the segment is equal to or smaller than half of the threshold.

2.4 Modularity with respect to sensors

Our recognition approach comprises a multi-modal on-body sensing hardware that is set up to facilitate both addition and removal of individual sensing modalities. As stated in Section 2.2, the recognition component employs a specific sensing modality in order to spot potential activity events.

The recognition approach starts with individual, class-dependent processes with a fast spotting stage that provides an initial guess about possible locations of the relevant class while removing obviously non-relevant signal segments. It is optimized for high recall and low precision to ensure low deletion rates.

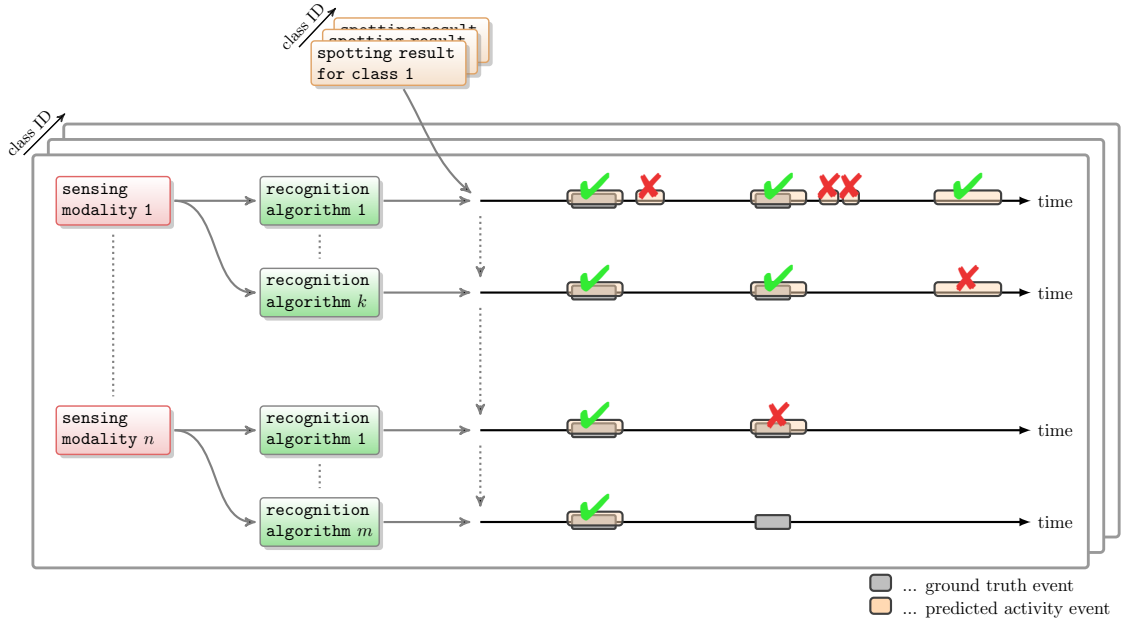

Figure 2.4
The figure illustrates the class-wise masking passes (plausibility analysis).

Due to the reason that each spotting process is only responsible for a single activity class this approach assures a maximum *recall* spotting stage – at least the maximum that can be achieved considering the particular sensing modality and considering the particular parameterization of the spotting process.

In the next stage any available sensing modality can be consulted for each individual event hypothesis to decide on potential incorrectly inserted activities. Once more this stage is parameterized in a class-wise manner to assure an optimal performance for each individual class. This stage might even apply different classifiers to decide whether a specific class hypothesis achieved from a particular sensing modality is plausible with respect to a specific other sensing modality; or speaking in terms of context: Do various contextual information sources agree on a specific activity event hypothesis?

The idea behind this is to incrementally improve the precision of the system through the application of a sequence of what we call *masking passes* or *plausibility analyses*. A masking pass works on the signal segments that have been identified as possible occurrences of a given class by the spotting stage. It makes a binary decision of either retaining or rejecting the segment. In the later case the segment is not presented to the following stage. For a schematic overview of this see Figure 2.4.

Evidently, masking passes are activity specific, so that each of the parallel activity spotting processes can have its own sequence of different or differently tuned/parameterized masking passes. The sequence of passes can be determined class-dependently. This approach allows easy removal/addition of sensors or algorithms without changing the rest of the system. Adding a new masking pass requires no changes in the rest of the recognition system regardless whether the pass uses different sensor signals or just different algorithms. Note that since all that a masking pass does is discard event candidates, the system is not capable of

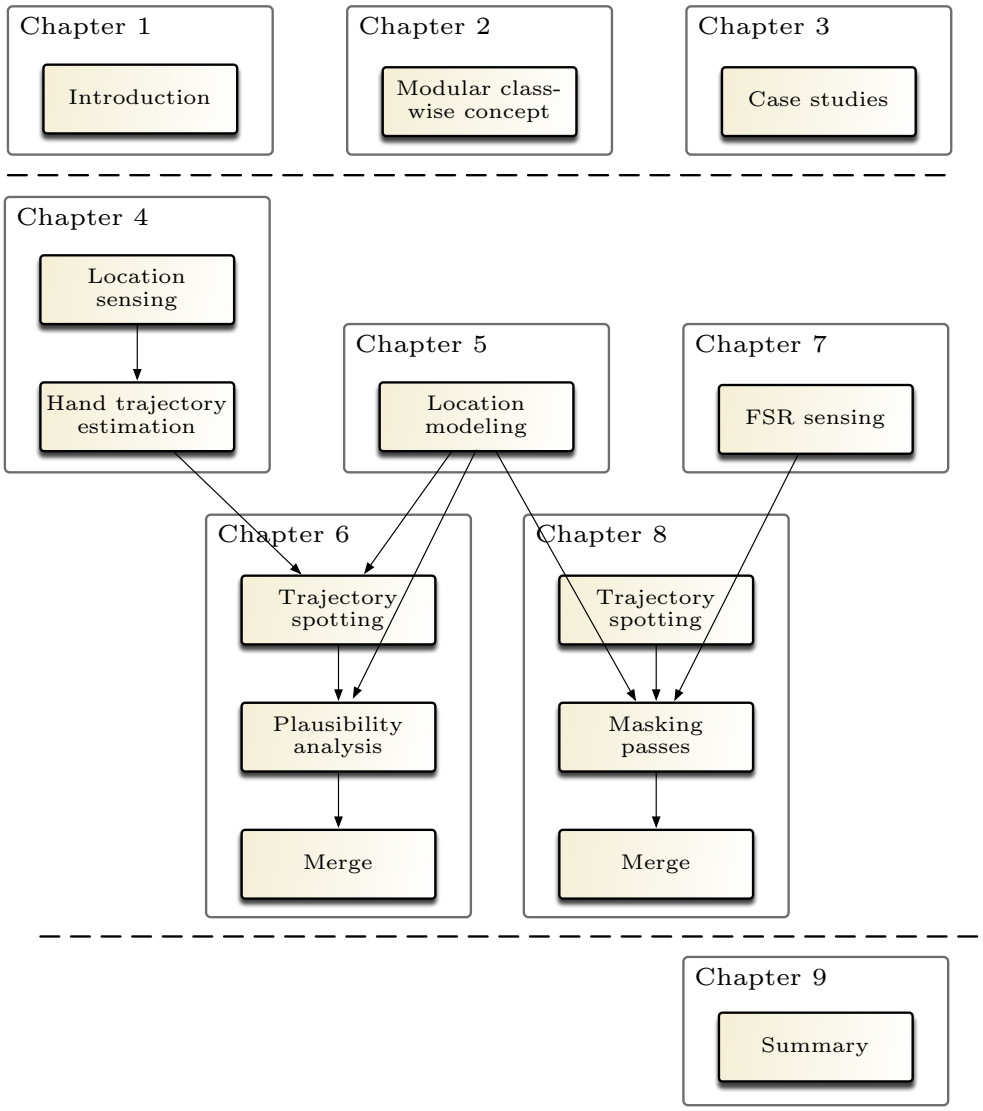

Figure 2.5
The figure depicts the contributions of the different chapters to the proposed modular recognition system (Chapters 4 to 8).

detecting events that are deleted during the spotting step.

This masking stage may also include high level information about the activities. This may comprise knowledge about typical event length and most likely event sequences or even impossible event sequences.

Another advantage of this approach is that it is tolerant of synchronization mismatches between sensors. Events will in general be long in relation to the sensor sampling rate. Since additional sensors are applied to event candidates produced by previous masking stages, synchronization needs to be accurate on event time scale not on sampling rate time scale.

Given a set of individually and class-dependently trained sensing modalities and algorithms the recognition system can also take over the responsibility of deciding which sensing modality is applied for which activity depending on the currently available hardware setup. In such a way not only the effect of addition and removal

of individual activity classes but also of individual sensing modalities on the overall performance of the recognition system is minimized.

2.5 Outline

The proposed gesture spotting and recognition approach uses multi-modal sensing and multi-modal, class-wise processing in order to achieve modularity and thus flexibility. Hence Chapters 4 and 7 investigate two on-body sensing modalities, i.e. location sensing and force sensitive resistor (FSR) based muscular activity sensing. Chapter 5 describes the location models applied for the location spotting and recognition stages. Finally Chapters 6 and 8 apply the presented approach on two different experimental continuous activity recognition tasks. A chapter-wise overview of this contributions and the attendant chapters is also depicted by Figure 2.5.

Chapter

3

Case studies*

This chapter gives a summary of the three presented case studies. The case studies are inspired by the mobile worker scenario, more precisely the maintenance and production assembly scenario. As a consequence, the case studies investigate a bicycle maintenance task and a car assembly conveyor belt quality assurance task.

An additional case study – giving a multimedia talk – is designed to test the limits concerning gesture recognition based on monitoring of lower arm muscular activities by means of a FSR system.

*The case studies presented in this chapter have already been presented in preliminary publications, see references [SOJ+06, OKL07, OSLT08].

3.1 Introduction

A series of case studies was performed in conjunction with this thesis to test and evaluate the gesture recognition abilities of the envisioned wearable sensor setups. The activities in our mind are *manipulative hand gestures*, i.e. *arm movements or rather a successive sequence of arm movements with the objective of manipulating a specific object either with or without using a tool*, e.g. opening a notebook, tightening a screw, etc. More precisely we focus on manipulative gestures in the area of industrial assembly or maintenance tasks (see also Section 1.2 on page 2).

Typical wearable approaches to activity recognition are based on acceleration and gyroscope sensors as the basic source of information. As stated in Section 1.3 on page 4 this work aims to explore the use of additional on-body information sources, namely on-body location sensing and the monitoring of muscular activities of the lower arm by means of force sensing resistors (FSR).

In order to explore the use of these additional information sources, a series of case studies was conducted. This chapter gives a summary of these three case studies. In addition it discusses the design choices and design parameters, their goals and challenges, and their ability and constraints when it comes to general statements beyond the focus of the experiments.

3.2 The bicycle maintenance scenario

The experiment focuses on the tracking of the user's hand position in order to assist the activity and gesture recognition process. The envisioned application scenario within the wearIT@work project is the car assembly scenario (see Section 1.2 on page 2).

Due to shortage of space in our lab, we decided to start with a bicycle maintenance scenario. Previous results of this experiment and a preceding case study have already been presented in [SOJ+06, OSJ+05]. The experiment in [OSJ+05] has been further extended to reflect as realistically as possible a real-life continuous task tracking scenario. This includes[1]:

- ✗ The recording of a large data set, consisting of 404 minutes of overall data with 1240 relevant gestures.
- ✗ Frequent, random insertion of complex NULL events – 68.2% of the total data length and approximately 50% of the overall number of gestures performed.
- ✗ A separate, *non-in-sequence* training set, i.e. 3035 additional gestures with an overall length of 291 minutes.
- ✗ Consideration of inter-subject training and recognition, i.e. six subjects volunteered for the data recordings.

3.2.1 Sensor setup

The main focus of the experiment is the question whether knowledge of a user's hand position can significantly increase the recognition rate of the user's activity.

[1] The numbers are slightly bigger than those given in reference [SOJ+06] because additional data was used for the presented evaluations.

Thus the sensor setup encompasses a motion sensor system as well as a positioning system (see also Section 4.3.2.2 on page 52). As the general focus of our work is a wearable approach, no instrumentation or sensors of any kind have been attached to the bicycle. It has been mounted on a special repair stand

- ✗ for ease of reaching different parts and tools,
- ✗ to ensure repeatable and thus recognizable motions,
- ✗ to ensure the user is facing the positioning system,
- ✗ to ensure an exact and static position of the bicycle, and thus to ensure static locations of interest.

In order to use the ultrasonic system, the room has been equipped with four ultrasonic base stations (receivers). They have been placed at exactly predefined locations in the room and serve as the absolute reference for the UPS, for details see section 4.3.2.1 on page 50. The locations of the base stations have been chosen in a way to minimize the possibility of a subject occluding the line of sight between base stations and wrist-mounted devices.

Two types of sensors have been placed on the user, as depicted in Figure 3.1 on the next page. Ultrasonic devices (Hexamite HX900SIO) are mounted on both wrists. These two devices measure their distances to the four base stations in an alternating manner in order to be able to track the hands' position with respect to the bicycle. Thus each wrist-worn device simultaneously measures four distances; we will refer to these as *four distance channels*. Second, a set of nine IMUs (MT9B from Xsens) have been attached to the user's hands, lower and upper arms, the chest, and the thighs. The MT9B are sampled at a rate of $100Hz$ each, the ultrasonic transmitters at approximately $1.4Hz$ each. The presented approach uses only the wrist mounted devices.

3.2.2 The task

We adapted and extended the gesture set used in [OSJ+05]. The result is a set of 23 manipulative gestures that are part of a regular bicycle maintenance task. They have been chosen to provide as much information as possible about the suitability of our approach to the recognition of different types of activities. There are gestures that contain very characteristic motions as well as such that are highly unstructured. Similarly, there are activities that take place at different, well-defined locations as well as such that are performed at (nearly) the same locations or are associated with vague locations only. Table 3.1 on page 33 gives a full overview of the used gestures. The key properties in terms of recognition challenges can be summarized as follows.

- ✗ *pumping (gestures 1 and 2)* Pumping begins with unscrewing the valve. Thus it consists of more than just the characteristic periodic motion. Pumping the front and the back wheel differs in terms of location, however, depending on where the valve is during pumping the location is rather vaguely defined. People tend to use different valve positions for the front and the back wheel, which means that statistically there is a difference in the acceleration signal as well.

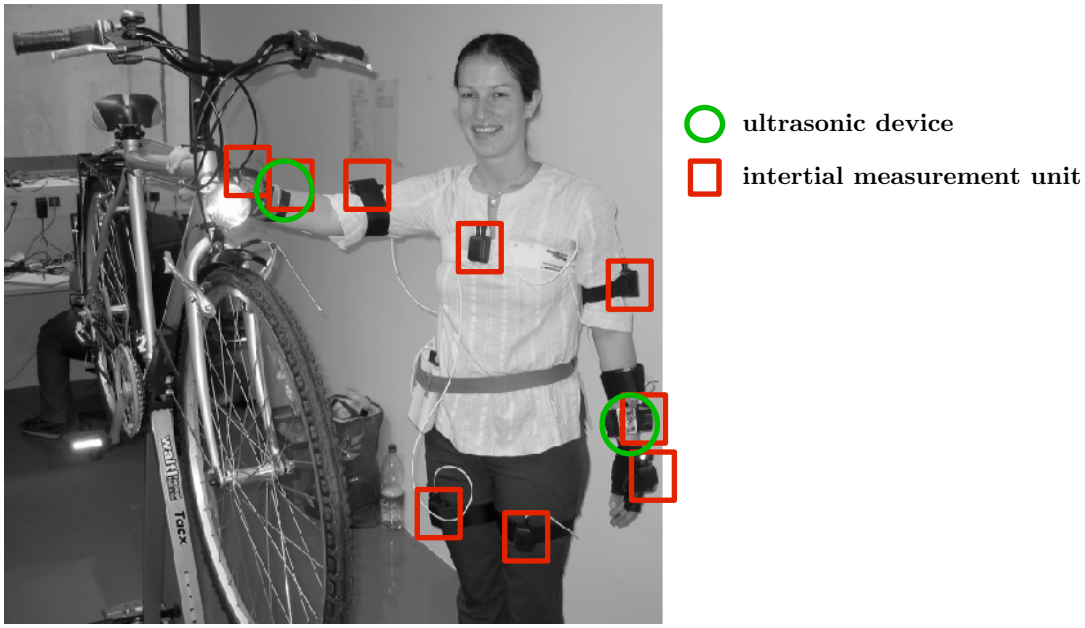

Figure 3.1
The figure depicts one subject of the bicycle maintenance case study as well as the bicycle mounted on a repair stand. The red squares indicate where the IMUs (Xsens MT9B) are placed on the body whereas the green circles indicate the positions of the ultrasonic devices (Hexamite HX900SIO). Note, that not all motion sensors are used for evaluation though used for recording. For the results presented in this work only the four wrist mounted devices (two IMUs and two ultrasonic devices) have been used.

- *screws (gestures 3 to 8)* The sequence contains the screwing and unscrewing of three screws at different, clearly separable locations. Screw A requires a hexagon wrench key, screws B and C require an ordinary screwdriver. Combined with different arm positions required to handle each screw, this provides some acceleration information to distinguish between the screws (in addition to location information).
- *pedals (gestures 9 to 12)* The set contains four pedal related gestures: just turning the pedals, turning the pedals and oiling the chain, testing the gear switch while turning the pedals and turning the pedals while marking buckles of the back-wheel with chalk. Pedal turning is a reasonably well-defined gesture.
- *(dis)assembly (gestures 13, 14, and 20 to 23)* Among the gestures most difficult to recognize are assembly and disassembly of the pedals, the front wheel, and the back light. All of them can be performed in many different ways, while the hand seldom remains at the same location for a significant time.
- *wheel spinning (gestures 15 and 16)* The wheel spinning gestures involve hand-turning of the front or the back wheel. The gestures contain a reasonably well defined motion (the actual spinning). However, there is also a considerable amount of freedom in terms of overall gesture. Front and back can be easily distinguished by location. In most cases different hand positions were used for turning the front and the back wheel.

Table 3.1
Set of manipulative gestures for the bicycle maintenance *scenario.*

full gesture class resolution		reduced gesture class resolution		
class ID	**description**	**group ID**	**description**	**periodic**
1	pumping (front wheel)	1	pumping (front wheel)	- /√
2	pumping (back wheel)	2	pumping (back wheel)	- /√
3	screw A (loose)	3	screw A	√
4	screw A (tighten)			√
5	screw B (loose)	4	screw B	√
6	screw B (tighten)			√
7	screw C (loose)	5	screw C	√
8	screw C (tighten)			√
9	turning pedal	6	turning pedal	√
10	turning pedal (chain oiling)	7	turning pedal (chain oiling)	- /√
11	turning pedal (testing gear switch)	8	turning pedal (testing gear switch)	- /√
12	turning pedal (wheel buckles)	9	turning pedal (wheel buckles)	- /√
13	disassembling front wheel	10	changing front wheel	-
14	assembling front wheel			-
15	turning front wheel	11	turning front wheel	√
16	turning back wheel	12	turning back wheel	√
17	testing bell	13	testing bell	-
18	seat (up)	14	seat	- /√
19	seat (down)			- /√
20	disassembling pedal	15	changing pedal	- /√
21	assembling pedal			- /√
22	changing bulb (remove)	16	changing bulb	-
23	changing bulb (insert)			-

- ✗ *bell (gesture 17)* Another challenging gesture is the testing of the bell. The time for ringing the bell up to five times is so short that only few location samples are available.
- ✗ *seat (gestures 18 and 19)* These gestures alter the position of the seat. The first increases the seating position by twisting the seat within its mounting using both hands. In addition to the twisting, the degrading gesture requires the pounding with a fist to drive the seat into its mounting.

3.2.3 The class sets

The above gesture set contains many pairs that differ only with respect to one small detail. This includes fastening and unfastening a given screw, assembling and disassembling the pedal/light and lowering and raising the seat. In every pair, both gestures are performed at the same location. The motions differ only slightly. Thus both fastening and unfastening a screw involves rotational motion in both directions. The difference is that while turning in one direction the screwdriver is engaged with the screw, while in the other it is not. We have labeled such pairs as two distinct gestures, since the objective of this work is to test the limits of recognition performance. However, it is also interesting to see the overall system performance without these distinctions. To this end, we have defined a second, so-called reduced gesture set, in which such nearly identical pairs are treated as a single activity, see Table 3.1 columns 3 and 4. We will refer to this as reduced gesture class set or *reduced gesture class resolution*.

In addition the classes can be grouped together due to their hand location, see Table 5.1 on page 75 columns 2 and 3 for right hand locations and columns 6 and 7

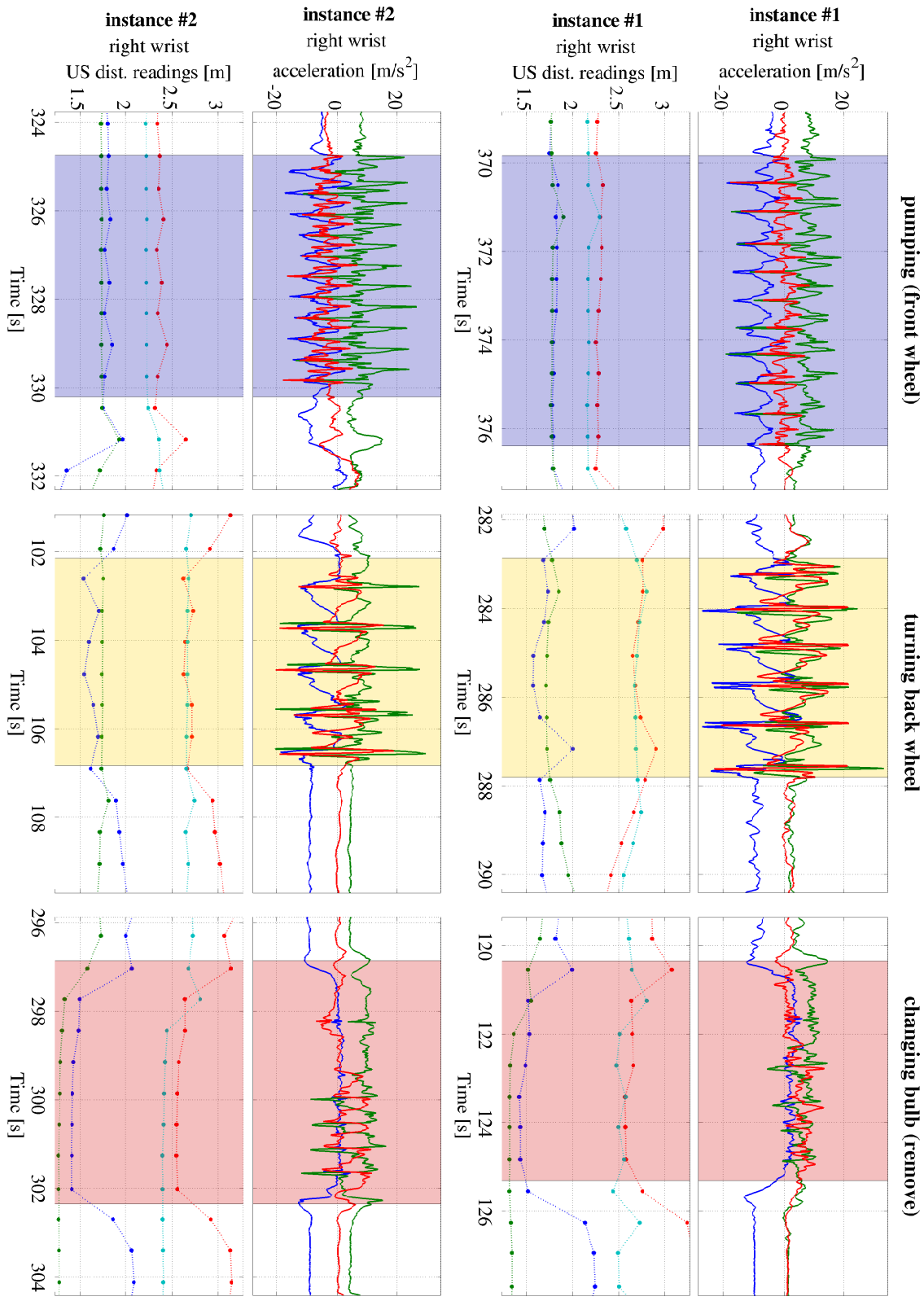

Figure 3.2

Signal examples for three different gesture classes of the bicycle maintenance case study. The plots depict a subset of the signals recorded for the right wrist. Each gesture class is illustrated by one instance of two different subjects.

for left hand locations. Analogous to the above considerations we will refer to this class set as the location class set or the *location class resolution*.

3.2.4 The NULL class

The difficulty of continuous recognition depends on the complexity of the NULL events that separate the task related gestures. Recognition would be fairly easy if the user could be relied on to start and finish each gesture in a well defined position. It would also be easier if the relevant activities were performed immediately after each other with little in between but moving between the different locations. Neither of the above is likely in a real-life maintenance scenario. As a consequence, we have put great emphasis on having complex, random NULL events in our tests. To this end the following events were randomly included in the stream of manipulative gestures used to test our method.

- ✗ Walking over to a notebook placed about three meters away from the bike to type a few characters.
- ✗ Cleaning a user-selected part of the bike. Here the NULL class could be potentially close to some relevant gestures both in terms of location and motion.
- ✗ Holding on to a user-selected part of the frame for a user selected period of time (a few seconds). The user was free to choose a location and often chose a location relevant for a specific gesture.

In addition to the above random gestures the user had to pick up and put away tools. No instructions were given to the user what to do/not to do between gestures. Overall the NULL class amounted to 68.2% of the recording time whereas no NULL class event interrupted a manipulative gesture. A random number generator was set to generate approximately as many NULL class events as there were real events in the sequence.

3.2.5 Data recording

We recorded two types of data sets. The first type comprises approximately 20 repetitions of each of the 23 available manipulative gestures. Here, the repetitions are separated by a few seconds during which the subject returned into a defined *home* position. This data type is called *train data*.

The second type of recordings involves all 23 manipulative gestures in a randomly generated order. We refer to this data type as *sequence data*. This ensures that even gestures with little complexity are carried out with a certain variability, thus giving the data real-life conditions. The gestures in the sequence data are separated by randomly inserted NULL class events as described above.

The experiment was performed by one female and five male subjects. For each of the six subjects, we recorded at least 20 train repetitions of each gesture and nine sequences containing all of the 23 gestures. This resulted in more than 3035 different gesture instances of the type *train*, i.e. 291 minutes; 1240 gesture instances of the type *sequence*, i.e. 128 minutes of gestures. The whole length of sequences was 404 minutes.

We used the framework proposed by [BAL08, BKL06] for recording and annotating the sensor data. With this software tool, the different sensor streams could be merged into one data file and simultaneously annotated using a regular keyboard. Any errors in the annotation stream were corrected off-line using the MARKER tool [Amf].

3.2.5.1 Training set rationale

Splitting the data into a train and a test set in the above way is fostered by practical considerations related to the envisioned real-life implementation of such systems. On any given piece of machinery, the set of possible individual actions (manipulative gestures) that can be taken is likely to be by far smaller than the set of all possible maintenance sequences. In fact since the maintenance sequences are permutations of individual actions in theory, there can be exponentially more sequences than individual actions. As a consequence in any practical system we will have to train gestures individually and not as part of a sequence that is trained as a whole. This also makes labeling of the training sequence much easier since predefined start and stop positions can be used.

The downside of this method is that if the gestures are trained separately as described above, the onset and the end phase of the gestures are likely to be different than in a real maintenance sequence. Also as a person is likely to repeat the same gesture a large number of times, the repetitions are likely to get *sloppy*. Thus the training will be less effective. However, since the objective of the work was to get as close as possible to a real-life scenario, we have used this training strategy despite the drawbacks mentioned above.

3.3 Giving a talk

As

> *context is not only about location [Jim08, Men03]*

we also aim at additional sensor modalities, in this case *Force sensitive resistors* (FSR) for the use of muscular activity measurements, as will be fostered in Section 7.2 on page 103.

This experiment design was not driven by a specific realistic application. Instead we aimed to implement a scenario that contains a reasonable variety of realistic gestures that would test the limits of what can be achieved with FSR based muscle activity monitoring. In particular we wanted to have a mixture of bold and subtle gestures, gestures performed with different force. At the same time we wanted to have a situation where the gestures are performed in a natural way as part of an everyday activity rather than make the subjects perform a set of artificial gestures.

3.3.1 Activities

As a consequence of the above considerations we have opted for a scenario of giving a talk, e.g. at a seminar. Table 3.2 on the opposite page lists the gestures performed. They can be organized in four groups each consisting of four different subroutines.

Table 3.2
Set of gestures for the talk *scenario.*

class ID	description	class ID	description
1	open the pen	9	open notebook
2	write on white-board	10	type a few words
3	close the pen	11	use the mouse
4	erase	12	close the notebook
5	scroll screen down	13	open bottle
6	point on screen	14	pour water in a glass
7	scroll screen up	15	close bottle
8	press buttons on remote control	16	drink

- ✗ *white-board (gestures 1-4)* First the subject opens a pen, then he/she starts to write some random words on the blackboard. After that he/she closes the pen, which is similar to the opening movement. Finally the words are cleared with an eraser, which is usually done in a periodic circulate movement.
- ✗ *beamer screen (gestures 5-8)* This gesture set contains pushing down the screen for the beamer. Then the subject points on the screen with a finger to show some things. After that the screen is lifted up again, which is similar to the first movement. The last action is using the remote control to switch off the beamer by pressing keys.
- ✗ *computer (gestures 9-12)* It starts with opening a laptop. After that some random text is entered via the keyboard. Then some random mouse movements are done. Next the laptop is closed again, which is a similar movement to the opening of the laptop.
- ✗ *drinking (gestures 13-16)* A bottle of water is first opened by unscrewing the lid. Water is then poured into a glass and the bottle is closed again. Finally the subject takes a sip of water from the glass.

3.3.2 Sensor setup

We mounted the FSR sleeve (see also Section 7.2 on page 103) on the right forearm. Furthermore the subjects wore a glove with a MTx sensor from Xsens (see Section 4.2 on page 47) comprising acceleration, gyroscopes, and magnetic field sensors, each of them in three dimensions. The MTx sensor is mounted on the back of the hand rather than on the wrist to prevent its housing from influencing the FSRs. Note that this provides significantly more information on palm actions than the more typical and much less obtrusive wrist based setup and, to a degree, neutralizes the inherent advantage of FSR.

Each FSR is sampled at a rate of $20Hz$, the MTx is sampled at $50Hz$.

3.3.3 Data recording

In total we recorded two subjects, with ten data sets each and one instance of every activity listed in Table 3.2 per data set, i.e. a total of 320 gestures. The experiment environment was a meeting room at our lab with a white-board, a beamer screen and a meeting table.

Figure 3.3
A user wearing the FSR sleeve on the lower arm, the MTx glove, a FSR thigh bandage, and a wearable computer used as a recording system during the experiments.

The sensor data acquisition and the ground truth annotation was again done using the framework proposed by [BAL08, BKL06]. As before any errors in the annotation stream were corrected off-line using the MARKER tool [Amf].

3.4 The car assembly scenario

This case study investigates manipulative hand gestures within a quality assurance procedure at the production site of the car manufacturer Škoda. This experiment aims at tracking the progress of the final quality inspection procedure at the end of a car production line. The procedure involves actions such as opening the trunk, door and hood, sliding the hands over parts of the car to detect gaps of wrong size, and moving parts such as the steering wheel and the seats. Table 3.3 on page 40 lists the gestures performed within this case study. They were chosen from the approximately 40 gestures actually performed at the Škoda factory in Mladá Boleslav at a certain point in their quality assurance process. There is neither a specific sequence nor a time frame within which the activities are to be executed by the worker. Between the relevant actions the workers could perform arbitrary other activities, e.g. picking something up, just walking around the car, talking to a work-mate, and so on.

This case study brings the scenario closer to the envisioned final application scenario. What is more, the manipulative gestures partly become more subtle than those in Section 3.2 on page 30.

3.4.1 Sensor setup

Instead of the user's hand position (as described in Section 3.2 on page 30) as additional source of information we now want to use the position of the user itself – which decreases the position information content.

Thus the worker's relative position to the car body is measured with an UWB

Figure 3.4
Examples of activities performed at the checkpoint part of the assembly line. From top to bottom and left to right these activities are: check trunk gaps (Class ID: 14), open the trunk (3), open the spare wheel box (18), writing (20), testing the fuel lid (6), open the right door (9), testing the mirror fixation (13).

positioning system from Ubisense (see also Section 4.3.2.3 on page 53). Four tags on the worker's shoulders enable the system to calculate his/her position with respect to four reference base stations placed around the car. Four tags are used to deal with the high data loss rate caused by a large amount of metallic objects around, e.g. the car. The UPS used in the bicycle maintenance case study (see Section 3.2.1) was exchanged by an UWB positioning system, because the ultrasonic signals are reflected at large scale from the metallic car body, which makes the UPS unreliable within this environment. Reflections are crucial, when the body-mounted device is close to the car in particular, i.e. there is no direct line of sight between transmitter and receiver, but at the same time the metallic car body reflects the signal without appreciable loss; that, of course, results in meaningless distance measurements. Thus the ultrasonic transmitters are not trackable in the close area of the car. The UWB system proved to be less influence-able by the metallic car body. Still there is a considerable amount of false or even no position readings, see also Figure 3.5 on page 41.

Analogous to Section 3.3 on page 36, we aim to measure the muscular activities of the lower arm by means of using two custom-built sleeves, each integrating an eight-channel FSR unit, as will be described in more detail in Section 7.2 on page 103.

The major sensor source is a set of motion sensors to measure the upper body motions. [SRO+08] describes the *Motion Jacket* that was developed to integrate the required motion sensors in an unobtrusive and robust working jacket. The jacket captures the upper body motion of the worker from seven IMUs (Xsens MTx) (see

Table 3.3
Activity classes and appropriate location classes for the car assembly *scenario. The last column defines which hand is involved (r=right, l=left, b=bi-manual).*

class ID	description	location classes	gesture type
1	open hood	1	b
2	close hood	1	b
3	open trunk	5	b
4	check trunk	5	b
5	close trunk	5	b
6	fuel lid	6	r
7	open left door	4,8	l
8	close left door	4,8	l
9	open right door	6,7	r
10	close right door	6,7	r
11	open two doors	6,8	b
12	close two doors	7,8	b
13	mirror	3	r
14	check trunk gaps	5	b
15	lock check left	4,8	l
16	lock check right	6,7	r
17	check hood gaps	2,7	b
18	open spare-wheel-box	5	r
19	close spare-wheel-box	5	r
20	writing	4,7	b

also Section 4.2 on page 47) within the jacket. The IMUs are placed on the lower arms, the upper arms, the torso, and on the back of the hands. The subject had to wear two gloves in addition to the Motion Jacket to mount the latter two IMUs. Two data acquisition units collect the data from these IMUs. The IMUs capture arm and hand orientation and thus their motion, which provides information about the activity performed by the worker.

The FSRs are sampled at a rate of $60Hz$ each, the MTx at $50Hz$, and the Ubisense tags at $100Hz$.

3.4.2 The task

As stated in the beginning of this section, the task performed by the subjects is a *car assembly quality checking procedure* copied as realistically as possible from an actually performed checking procedure at the Škoda factory. Table 3.3 lists the gesture classes re-enacted within this case study. The classes can be grouped into four categories:

- *Opening and closing:* These classes are short movements with a single hand without a significant body movement for the doors (classes: 7-12) and bi-manual gestures with a significant body movement for the trunk, hood and the spare wheel box (classes: 1,2,3,5,18,19).
- *Lock tests:* Testing a door lock (classes: 15,16), the trunk fixation (class: 4), the fuel lid (class: 6) and the mirror fixation (class: 13) are repeated usually

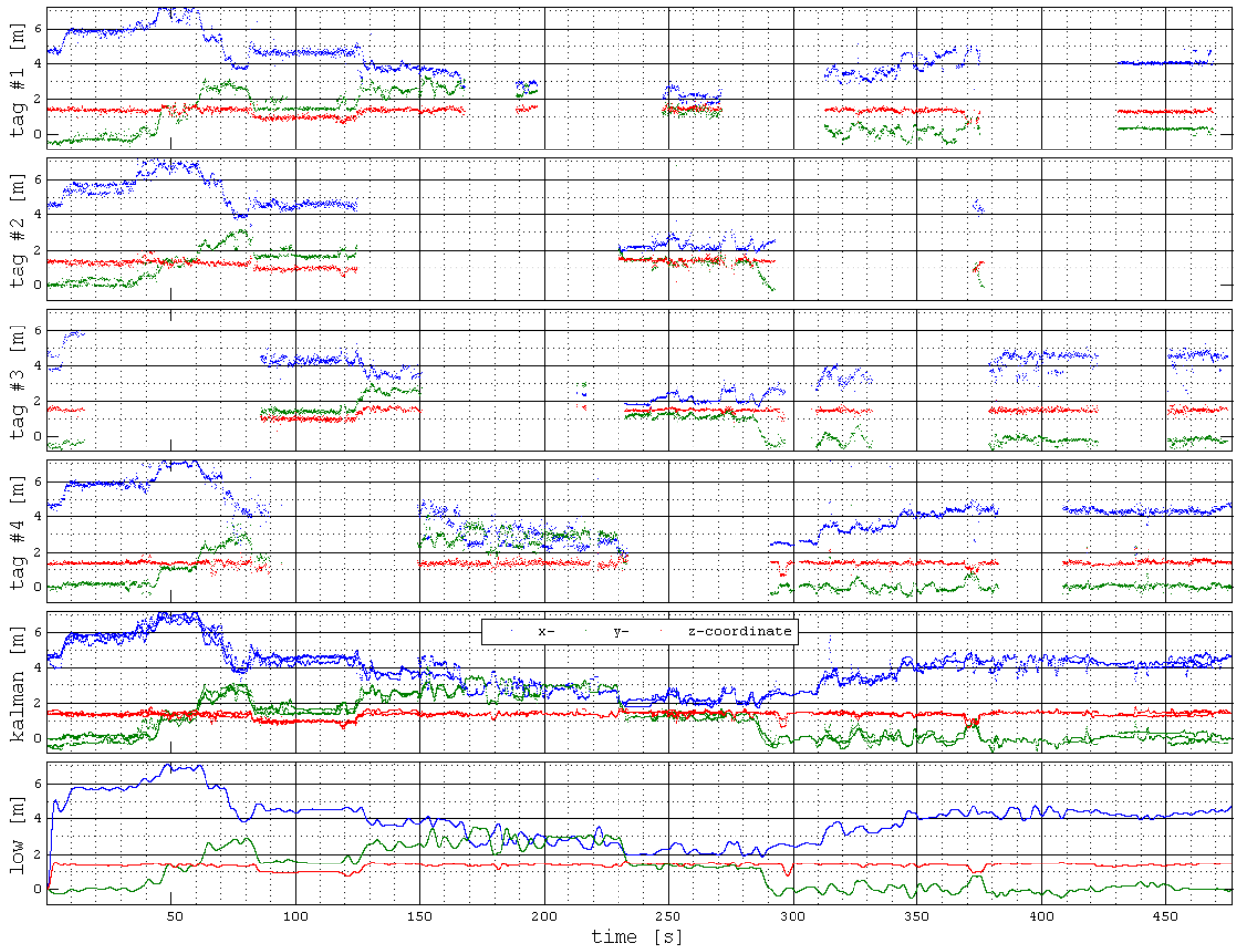

Figure 3.5
Data loss of the Ubisense system is avoided by using four tags. The picture depicts the data loss of the individual tags for a single run in plots 1 to 4. Plot 5 depicts the Kalman smoothed tag results merged together and plot 6 the final position result.

between three and five times, thus these gestures result in a repetitive pattern in both the motion and muscle signals.

- *Gap tests:* Testing chassis gaps (classes: 14,17) is a sliding movement of the fingers over the tested gap.
- *Writing:* Writing (class: 20) itself is a subtle gesture but should be distinguishable by its unique start and stop: taking the pen, putting away the pen. When unused, the pen is kept in a pocket of the Motion Jacket.

3.4.3 Data recording

We collected data in-situ at the Škoda production facility. A worker wearing the Motion Jacket performed the procedure on ten cars while the cars were moving on the conveyor belt of the assembly line. In addition we observed and filmed several other workers performing the procedure for later analysis.

Due to the high cost of interruptions, large scale data recording within the production process was not possible. The only way to collect a comprehensive data set was to recreate the production environment in our lab. For that Škoda provided us with a complete car. The video material from the factory was used to ensure a realistic setup.

On this recreated setup we recorded a data set with eight subjects (students, instructed from the video material) who each conducted ten repetitions of the procedure, see also Figure 3.4 on page 39. We collected about 3680 checking activities within 560 minutes of data. One experimenter annotated the start and end points of activities to provide an absolute reference, while a second experimenter simultaneously annotated the location ground truth of the user. As in the previous case studies both experimenters were using the framework proposed by [BAL08, BKL06] to synchronize the annotation streams with the data streams. As before any errors in the annotation stream were corrected off-line using the MARKER tool [Amf].

3.5 Summary and discussion

This chapter described the parameters of the experiments conducted in order to evaluate the methods in mind.

The talk scenario is designed to test the limits of FSR based gesture recognition. In particular we wanted to have a mixture of bold and subtle gestures, gestures performed with different force. This was intended as a step ahead of previous evaluations of basic motions such as bending an arm. The talk scenario was chosen to have a situation where the gestures are performed in a natural way as part of an everyday activity rather than make the subjects perform a set of artificial gestures.

The bicycle maintenance experiment aims at investigating the use of on-body hand tracking for the recognition of manipulative hand gestures. Gestures included in the experiment contain very characteristic motions as well as such that are highly unstructured. Moreover, there are activities that take place at different, well-defined locations as well as such that are performed at nearly the same locations or are associated with vague locations only. This way the experiment was intended to test the limitations concerning the gesture recognition ability of the proposed sensor approach.

Moreover, the bicycle scenario put great effort to model a complex and diverse NULL class to test the limits of location based gesture spotting.

Finally the car assembly scenario combined all three suggested sensor modalities – location, motion, and FSR based muscular activity measurement.

It aims at bringing the experiment from the laboratory setup to a *real-life* test scenario. That is why data recordings were performed at a real quality assurance conveyor belt at the Škoda factory in Mladá Boleslav. A Škoda worker volunteered as a subject for a two-hour data recording session executing his normal checking procedure. The experiment set up hereafter at ETH Zürich was as closely re-enacting the same quality assurance procedure as possible. It comprises 20 of the approximately 40 activities actually performed by Škoda workers at this specific part of the quality assurance process.

Moreover, the experiment aims at evaluating the sensor setup on its ability to provide a modular and thus flexible sensing setup.

The gestures included in the experiments are intended to reflect the diversity of the specific scenario, but by no means constitute an exhaustive set of possible activities in that specific application domain. Moreover, the gestures are performed in a uniform manner by all subjects, i.e. the subjects are instructed which hand to use for which tool and so forth. In such a way the outcome of the experimental evaluations has limited ability for generalization. It remains an open issue whether the presented approaches can be expanded in such a way that they can also handle a greater diversity in the way a certain gesture is performed.

Chapter

4

Hand Tracking*

This chapter outlines some background information about indoor location techniques. The requirements to the location systems concerning our specific application scenarios in mind, the chapter as a consequence gives reasons for specific indoor location system choices and describes the realization and the setup of the finally used sensor systems.

Furthermore, the chapter outlines the position estimation methods used. The methods are intended to be able to handle the great presence of erroneous distance readings due to the limitations of the ultrasonic positioning technique.

Moreover, the chapter describes our approach to fuse motion and orientation readings with ultrasonic based position estimations, resulting in a hand trajectory with a far better dynamic response than the mere ultrasonic positioning approach is able to provide.

*Parts of Chapters 4, 5 and 6 are under revision for publication in Springer's *Pattern Analysis and Applications*, see reference [OLST].

4.1 Introduction

As described in Section 1.3 purely body-worn sensors have only limited capability of detecting which part of an object the user is interacting with. Among other sensing modalities we aim to use hand tracking by means of ultrasonic indoor positioning to deal with this issue.

Ultrasonic indoor positioning systems (UPS) are widely used for indoor location [MRC05, SBGP04, WJH97] and relative positioning [HKG+05]. They are relatively cheap and require only little infrastructure. Placing four base stations in predefined locations in the environment is usually sufficient; at least when observing a space of up to approximately $100m^2$. Due to physical properties of ultrasound (see also Section 4.3.2.1) such an approach accounts for a number of problems when used for hands tracking. In particular, it is subject to reflections and occlusions and has got limited sampling rates in the range of 1 to $5Hz$.

As the proposed setup also comprises body-worn motion sensors – so-called inertial measurement units (IMUs) – actually an additional hand tracking system is available. This is because of the fact that the applied IMU platform measures its absolute orientation in three dimensions. By means of concatenating several IMUs and a well-defined concatenation model a trajectory of each IMU platform in the reference system of any other IMU platform within this concatenation can be derived. In such a way a trajectory of the hand can be calculated for the given sensor setup – at least in the reference system of the upper body. [Sti08, SRO+08, SRT07b, SRT07a] describe how a set of such IMUs can be combined to derive trajectories of different body parts used thereafter for activity and gesture spotting and recognition. A short overview on this trajectory estimation method is given in Section 6.4.1 on page 83.

Due to the fact that the given sensor setup provides two independent hand tracking possibilities, we will suggest a filter design capable of fusing an absolute aiding source – namely the UPS measurements – with the inertial measurements to achieve similar trajectories as in [Sti08, SRO+08, SRT07b, SRT07a]. [Roe06] has already described how the same IMU platform can enhance an absolute aiding source by means of a complementary Kalman fusion filter. Note that the vision based aiding source used in [Roe06] runs at $120Hz$, the IMUs sample at $100Hz$ as is the case with the IMUs used throughout this thesis, whereas the UPS samples at just $1.4Hz$.[1] Thus the fusion filter will be designed in way to deal with the differences in the sampling frequencies and to up-sample the slow hand position updates from $1.4Hz$ to $100Hz$.

Note that the trajectories that can be derived from a mere on-body IMU approach still lack absolute position information, i.e. the reference system is defined in relation to the person wearing the system whereas the herein presented approach results in trajectories whose coordinates are given in relation to the reference system of the aiding source and thus in a global reference system.

Moreover, the presented approach requires a slimmer sensor setup. A mere

[1] The UPS sampling rate is of course depending on the actual chosen setup. A sampling rate of $1.4Hz$ corresponds to the setup applied throughout the experiments presented in this thesis, i.e. two ultrasonic transmitters (one on each wrist) and four ultrasonic base stations (receivers) resulting in eight sequential time-of-flight measurements.

IMU based hand trajectory estimation approach makes use of at least seven devices, as shown in [Sti08, SRO+08, SRT07b, SRT07a], when tracking both hand positions or five devices when tracking both wrist positions. In contrast to this the mixed UPS/IMU approach needs four devices (one ultrasonic transmitter and one IMU on each wrist or lower arm) in the current setup. An optimized hardware implementation would integrate both hardware devices into one platform[2] resulting in just two on-body devices for the mixed hand tracking approach.

On the other hand the IMU based approach benefits from the fact that one can also derive upper body postures in addition to the relative hand or wrist trajectories. Such body postures can give essential hints to the activity spotting process as shown in [ZWS09] where the authors use the data set described in Section 3.4 on page 38 to derive these body postures from five upper body worn IMUs.

In addition to the position estimation approaches this chapter will describe in more detail the indoor location techniques in use. Sections 4.2 and 4.3 as a consequence give a preliminary introduction to the applied sensing systems.

4.2 Motion and orientation sensors

Acceleration and gyroscope sensors – measuring the rate of turn – are a state-of-the-art approach to on-body activity recognition. As shown e.g. in [Roe06] three-dimensional acceleration and gyroscope sensors can even be incorporated into an orientation sensor system. Such a sensor setup measures the orientation in an inertial manner effecting that even tiny drift errors of either sensor result due to the integration step – double integration for the acceleration sensor – in an orientation error rapidly growing over time.

A state-of-the-art approach to circumvent this issue is the incorporation of an absolute aiding sensor source, fusing the inertial measurements with absolute measurements. As [Roe06] shows, a three-dimensional magnetic sensor – despite its limitations – is feasible to serve as an absolute reference source in such an orientation sensor system.

Throughout this thesis we use sets of such orientation sensor systems or IMUs. More precisely, we apply the MTx and its predecessor the MT9B both commercially available from Xsens[3] for the use of tracking the motion and the orientation of different upper body parts. The same sensor system is used for activity recognition e.g. in [ZLS07, Jun05].

4.3 Location tracking

As this thesis investigates the use of location tracking in the field of activity and gesture recognition in indoor scenarios, we are limited to location techniques (see Section 1.5.3) functioning in indoor environments.

[2]For outdoor applications such devices are already commercially available as acceleration/rate of turn/air pressure enhanced GPS devices.

[3]http://www.xsens.com/ – A similar measurement system is also provided by the InterSens InertiaCube2+, see http://www.intersense.com/InertiaCube_Sensors.aspx?

When deciding on a location system a number of decisions have to be made or rather a number of requirements and their priorities have to be defined in advance. These requirements can be of the following kind, but are not limited to:

- ✗ form factor
- ✗ power requirements
- ✗ infrastructure vs. portable elements
- ✗ resolution in space
- ✗ resolution in time
- ✗ accuracy in space
- ✗ accuracy in time
- ✗ scalability with respect to the number of objects to be tracked
- ✗ scalability with respect to the area to be observed
- ✗ dynamic issues
- ✗ user-centralized vs. infrastructure-centralized vs. a mixed approach.
- ✗ privacy and security issues
- ✗ auto-calibration methods
- ✗ output: absolute vs. relative
- ✗ output: raw measurements (e.g. distance measurements or angle of arrival measurements) vs. coordinates (physical location) vs. location classes (symbolic location).[4]
- ✗ physical phenomena used for location determination and, what is more, does it fit into the environment? Such systems might affect the environment, e.g. clinical instruments, or, vice versa, the environment might disturb these location techniques, e.g. metallic environments.

4.3.1 Concerns

Indoor localization techniques have widely been investigated and developed throughout the last decades. Trying to give an exhaustive summary of techniques and systems that have been developed – and are still being investigated by both researchers and industry – in order to solve the indoor positioning problem is impossible because of the great number of approaches and suggestions. On the other hand, this great variety implies that no satisfactory or generally applicable solution has been found. Whereas for outdoor localization the Global Navigation Satellite System (GNSS) – with its implementations GPS (USA), GLONASS (Russia), GALILEO (Europe, not finished), Compass (China, not finished) – can be said to be useful within the majority of applications. Efforts have been made to enhance the system by other techniques such as GSM-based localization or vision [PWP+07] but this technique will most likely be state-of-the-art – despite, of course, further improvements – for at least the next couple of decades. For the indoor localization problem no general solution is in sight.

[4] For a definition on location models see e.g. [Leo98]. Symbolic locations themselves can be defined in a supervised (see e.g. [GF03]) or unsupervised manner (see e.g. [AM06]).

Thus researchers have started to endorse a *top down* approach instead of the so far widely applied *bottom up* approach when designing and implementing an indoor location system. In this context *top down* implies roughly the following design process:

- ✗ design of a possible solution to the positioning problem,
- ✗ design a widely applicable and thus often unhandy and bulky system, and
- ✗ testing the system's limitations concerning specific environments and application domains

whereas the *top down* approach demands that

- ✗ first the requirements regarding environment and contemplated application domain should be defined, and
- ✗ from these the list of requirements for the location system should be derived.

The location system implemented in the end should then be less generally applicable but ideal for the special case in mind. This is considered by commercial projects in particular; within research projects there are often various assumptions made concerning the location system. Assuming that an application under test depends on a high update rate and high accuracy (i.e. there are great requirements to the location system) an experimenter will have to make some efforts to set up a decent location system to prove the concept of the proposed application; thinking in terms of marketability these efforts will not pay off. Great efforts and expenses for long term investments such as *patient tracking* in hospitals will most likely amortize within the life span of the installed system, unlike short term applications like providing a location service for a three-day conference. Such a system should be more like a plug and play system. Thus auto-calibration techniques in particular (see e.g. in [DMC+05]) will be necessary for generally applicable indoor positioning systems.

Another important question is *user privacy*: the system can either function in a *tracking* manner or in a *(self-)positioning* manner. The Cricket team [PCB00] raises this concern; Hazas *et al.* also deal with this issue. In [HH06] the authors describe the requirements of a privacy oriented indoor positioning system, i.e.

> *[...] a user's presence is not advertised, even anonymously, and [...] entities outside of the user's control are not entrusted with gathering signal times-of-arrival or with calculating the user's location; otherwise, these entities may relay that data to other parties without permission.*

but they also remark:

> *It remains to be seen whether such systems provide benefits for the user in practice – denying location information to external devices severely limits the applications which can be made available, and complications arise when trying to reliably authenticate and distribute location information without being compromised by an attacker.*

4.3.2 System choices

Within this thesis the location systems are supposed to serve either as a hand location tracking system or as a user location tracking systems. In both cases the requirements to spatial and temporal granularity are high. The following sections describe the chosen location systems.

4.3.2.1 Ultrasonic hand localization

Hexamite, the Bat system [ACH+01] and Cricket [Pri05] are the traditional ultrasonic positioning systems (UPS). See also Section 1.5.3.4, which already gave an incomplete overview of available UPS implementations. The implementations vary significantly in their details but in general these systems rely on time-of-flight measurements between a mobile devices and at least four reference devices fixed at known positions in the environment. The categories in which most UPS implementations can be divided into are:

- ✗ *setting*
  - ✗ outdoors
  - ✗ indoors
- ✗ *purpose*
  - ✗ tracking
  - ✗ positioning
- ✗ *reference*
  - ✗ automated and absolute
  - ✗ absolute
  - ✗ relative
- ✗ *output*
  - ✗ distance readings
  - ✗ position estimations (coordinates)
  - ✗ location classes
- ✗ *synchronization*
  - ✗ wired
  - ✗ synchronized clocks
  - ✗ ultrasonic
  - ✗ RF-synchronization
  - ✗ asynchronous
- ✗ *bandwidth*
  - ✗ narrow-band
  - ✗ broadband

Due to physical properties a narrowband ultrasonic approach to hands tracking accounts for a number of problems. It is in particular subject to reflections and

occlusions and has got limited (1 to $5Hz$) sampling rates. Anyway we will decide in favor of such a system mainly due to its spatial resolution.

The envisioned setup with four stationary devices and synchronous distance readings allows to solve the positioning problem by adopting a Least Squares Optimizer (LSQ). However, independent of the UPS implementation details there are three issues that a LSQ cannot deal with properly:

- *Reflections:* Ultrasound is reflected by most materials present in the environment. Thus the location systems has to deal with false signals resulting from reflections.
- *Occlusions:* Ultrasonic distance measurements essentially require line of sight between the communicating devices. In case the transmitter turns away from the receiver or some person/object comes between the two, the signal is lost.
- *Temporal resolution:* The temporal resolution is limited by the speed of sound which is about is $340ms^{-1}$. In general several transmissions are needed to perform three-dimensional location (either one from every base station in the environment or one from every mobile device that needs to be localized). Unless some advanced coding schemes are used the transmission time slots need to be long enough apart for the reflections to subside. In a room a couple of meters in diameter this reduces the maximum number of transmissions to 10 to 20 a second. This means that the maximum realistic sampling frequency is a couple of Hz. Often – as is the case with the Hexamite sensors – it is about $1Hz$.

In the indoor location scenario – where ultrasonic devices are mainly used – the above factors can often be neglected and a standard extended Kalman filter approach would work fine. With base stations placed in the ceiling and the personal devices e.g. on the shoulder occlusions can be minimized. Except for applications dealing with fast motions, e.g. sports related scenarios, the temporal resolution of $< 10Hz$ is more than enough.

In the envisioned maintenance scenario things are much more difficult. As the transmitters need to be mounted on the arms, occlusions are a frequent problem. They may occur in case the subject

- is standing behind the maintenance object,
- occludes the moving devices or
- turns away from the fixed devices.

In all these cases two different problems can occur: either no signal reaches the measuring device in time (no measurement) or a reflected signal is measured (wrong measurement). In the majority of cases a reflected signal is easy to detect in case the reflection comes from a point far away, e.g. from a wall when the subject is standing in the middle of the room. So occlusions are likely to produce wrong, not detectable measurements in cases where the subject is close to an object, e.g. in cases where maintenance activities are performed.

The resulting coordinates of one moving device are dependent on distances to at least four fixed devices. The time frame for acquiring the distance to one fixed device is approximately 0.3 seconds. In the user-centered approach the distance

measurements are not simultaneous but consecutive, thus the calculation of the position of the moving devices is dependent on measurements with a time delay of at best 1.2 seconds. That means that the error for the resulting position is not so much dependent on the accuracy of the measurement system (approximately 2 to $3cm$ for the Hexamite system) than on the speed of the moving device.

4.3.2.2 Ultrasonic positioning system setup

The work at hand uses the *Hexamite* ultrasonic positioning system, more precisely the *Hexamite HX900* [5]. This system is specified as follows:

- ✗ *Bandwidth*: Narrow-band, frequency: $40kHz$.
- ✗ *Range*: The range is dependent on the voltage source, and can go up to ten meters.
- ✗ *Distance resolution*: The resolution is $1e^{-3}$ meter.
- ✗ *Accuracy*: The accuracy is approximately $3e^{-2}$ meter.
- ✗ *Synchronization*: Either wired synchronization or ultrasonic synchronization can be used. The ultrasonic synchronization causes a slower update rate; at least two times slower. The wired synchronization outperforms the ultrasonic synchronization mode due to faster update rate and more accurate distance readings. But the wired synchronization is uncomfortable for a user-centered setup due to the need of a wired connection between the moving devices. In principle the system would also be extendible to use radio frequency based wireless synchronization by means of off-the-shelf components.
- ✗ *Output*: Distance readings. The output is triggered by a master node asynchronous to distance measurements, i.e. the exact time-stamp of a distance reading has to be estimated by the position processing unit.

The reasons why we decided in favor of the Hexamite system are:

- ✗ good spatial resolution
- ✗ good spatial accuracy
- ✗ commercial availability.

For practical reasons the latter was the decisive argument. The drawbacks of the Hexamite implementation are

- ✗ *Output*: The system outputs distance measurements, i.e. one has to solve the positioning problem on its own, unlike with e.g. the Ubisense system. Kalman filter approaches to this problem are sketched in sections 4.4.4 on page 57 and 4.4.5 on page 58.
- ✗ *Missing time-stamps*: The system does not output the time-stamp of a certain measurement. That does not effect the room-centered approach but is a major drawback in case of the user-centered approach. In this mode the body-worn device polls the stationary devices consecutively, but outputs the results only after an entire measurement cycle; thus only the last measurement's time-stamp is known, any preceding time-stamp has to be interpolated.

[5] http://www.hexamite.com/hx900b.pdf

- *Synchronization*: A state-of-the-art UPS provides wireless synchronization. However, that does not effect the presented case study. Rather, the system setup for the final application would have to solve that issue.

The system is adopted in the room-centered mode with a single user wearing two transmitters, one on each wrist. The room is equipped with four ultrasonic base stations set up as receivers. The UPS is set to ultrasonic synchronization mode, thus the moving devices do not have to be wired with the stationary devices. Nonetheless, the two moving devices need to be connected for communication purpose, thus the devices are connected using a wired serial connection. Choosing this setup results in a positioning system as reliable and accurate as possible. A UPS can provide quite fine-grained distance readings compared to e.g. an ultra-wide-band system. Moreover, the chosen setup itself assures the best result that can be achieved by means of the applied Hexamite UPS.

4.3.2.3 UWB user localization

In addition to hand location tracking we also want to investigate how user location tracking can enhance on-body sensor based gesture recognition. The demands on an indoor user location tracking system regarding spatial and temporal resolution are low compared to a hand tracking location system.

Moreover, as this system had to be applied in a real-life test scenario at a Škoda factory site (see Section 3.4 on page 38) within a strict time frame of a couple of hours, stability, reliability, and ease of use have been one of the decisive categories.

The requirements to the location system can be summarized as follows:

- Stable, reliable off-the-shelf equipment, see above.
- Form factor: *wearable*, i.e. in our case the device carried by the user should be as small as possible, at least it must be small enough to be integrated in the work suit.
- Power requirements: battery driven, and batteries should last for at least one workday.
- Infrastructure vs. portable elements: within the factory environment there is room to install several base stations, i.e. the location system need not rely on moving devices alone.
- A couple of centimeters resolution in space is enough to define a user's location.
- A couple of Hz is enough to track a moving person. But, as will be seen later in this section, higher temporal resolution is needed anyway.
- Ability to track various sensor nodes at the same time. This requirement is at least valid for the final system setup, when tracking several workers at the same time within a factory site.

The major arguments for deciding in favor of the Ubisense [6] UWB system are:

- commercial availability,
- stable, reliable setup, and

[6] http://www.ubisense.net/

- ✗ it can be set up in a simple and straightforward process, which was essential because the system had to be set up on a factory site while the production line was running within a small time frame (see Section 3.4.3 on page 41).

The Ubisense system has a spatial resolution of approximately 20 cm and thus outperformed by the Hexamite UPS (see Section 4.3.2.1) by factor 10. Nevertheless, the Ubisense's spatial resolution seems adequate for the investigated user tracking scenario. The major reasons for deciding against the Hexamite system are:

- ✗ A bad scalability for large areas. The Hexamite system using four base stations can cover an area of up to approximately five by five meters at a supply voltage of $6V$.
- ✗ In addition, the system will not be able to deal with the huge amount of reflected and occluded ultrasonic signals due to the metallic environment in the car manufactory. What is more, the car being assembled will always be in the line of sight to at least some of the location base stations, while the worker is standing close to the car.

  The UWB system is not completely immune to that issue, but its high temporal resolution of up to $100Hz$ should be able to balance that problem.

4.4 Position estimation

4.4.1 Introduction

Each localization technique presupposes a specific algorithm to derive location information from the raw measurements. State estimation techniques that can handle a series of position readings are often very similar, though. Decent state estimation becomes necessary when relying on an inertial sensor system in particular or in case the dynamics of the system are essential. Locating a static object is easier than locating moving objects, particularly

- ✗ in case the update rate of the location system is low compared to the expected maximum speed and the acceleration of the moving device and
- ✗ in case of increasing spatial accuracy requirements.

Under these conditions localization becomes more difficult and demands a more sophisticated state estimation technique. For a survey on this issue refer to Fox [FHL+03] who identifies the following approaches:

- ✗ Bayes filters
- ✗ Kalman filters
- ✗ hybrid approaches
- ✗ topological approach
- ✗ particle filters

Additional approaches are:

- ✗ Markov processes, see e.g. [TMK04]

- ✗ Hidden Markov Models, see e.g. Krumm [Kru03], which gives a survey on this issue as well. Krumm summarizes Hidden Markov models, particle filters and Kalman filters as *recursive estimates*.

The trilateration problem, i.e. estimating a point in space from four or more absolute distance measurements, can be solved in various ways. A straightforward method is the *least squares optimization* (LSQ) approach.[7] The estimation of the best fitting point in space given a set of distance measurements is thus interpreted as an optimization problem with a given objective function. The major drawback of this approach is the fact that each set of distance measurements is treated separately, i.e. the fact that the hand positions are correlated in time is simply ignored.

A state-of-the-art approach to solve a state estimation problem given a *time-series* of measurements is the *Kalman filter*. In the past almost fifty years this algorithm became a state-of-the-art technique for positioning systems in particular. Since the measurement model is non-linear – in Euclidean space the coordinates of two points have a non-linear correlation with their distance – one has to make use of the extended Kalman filter. Depending on the efforts made when designing state transition model and measurement model the Kalman filter output is quite robust concerning erroneous input and thus concerning measurement errors.

The Kalman filter additionally has got the ability to fuse different sensor sources to an optimal solution. A lot of research has been carried out on how to blend inertial tracking data with absolute position measurements. Actually the field of sensor fusion has become a major application for the Kalman filter. This approach usually aims at combining high accuracy of an absolute measurement system with high dynamic response of an inertial measurement system. As this work is also aiming at combining inertial measurement readings with absolute location readings, we suggest and evaluate a Kalman filter design, see Section 4.4.5 on page 58.

There are different ways to implement such a fusion filter. Due to its good dynamic response a complementary filter design is a common method. Using a complementary Kalman filter we fuse the measurements of the orientation sensor system with the ultrasonic position measurements, achieving two different improvements:

- ✗ Reflections and occlusions of the UPS measurements are corrected automatically in most cases and thus the accuracy of the position estimation can be increased.
- ✗ The position estimation is up-sampled from the UPS sampling frequency ($\simeq 1.4Hz$) to the sampling frequency of the orientation system ($100Hz$). Thus the position estimation becomes more responsive to the dynamics of the hand motions. Hence the position estimation can be directly used for gesture recognition not only as a complementary contextual information source.

Refer to [BH97, RV05, Roe06] on how to set up such a complementary Kalman filter to fuse the measurements of an inertial navigation system (INS) with position measurements. Figure 4.2 on page 61 and Section 4.4.5 on page 58 describe our design of such a filter used for the evaluations presented in this work. In addition one could set up a constrained Kalman filter to exclude *impossible* states, reflecting

[7] For an exhaustive survey on how to solve positioning problems using LSQ see [CSMC06].

considerations similar to those listed in Section 4.4.2 – refer to [GH07] or [UDL07] for promising basic approaches.

4.4.2 Preprocessing

In previous approaches [OSJ+05, SOJ+06] we demonstrated how the inherent errors present in the ultrasonic signal can be handled through plausibility analysis based on physical constraints of the system; constraints concerning human anatomy and basic assumptions about plausible motions. These constraints reflect the following considerations:

- ✗ The user is equipped with a ultrasonic transmitters on each hand. Thus the maximum distance between these two devices is limited to a certain value. This maximum distance value is defined by the span of the user's arms minus two times the distance between fingertips and ultrasonic transmitter. That results in a typical maximum distance between $110cm$ and $140cm$.
- ✗ A UPS is based on distance measurements. Typical errors are caused by reflections or occlusions. A non-moving transmitter non-periodically produces wrong measurements due to these reflections and occlusions. In many cases these errors are present only on one of the distance channels. Due to the fact that it is impossible for a device to move on a path that is equidistant to three or more stationary devices, we can identify some of these *single-channel* errors: In case a segment with static distance readings on all channels is followed by a segment with a change bigger than the UPS distance measurement accuracy on a single channel, either the first or the second segment must be erroneous.
- ✗ The speed must be limited to a certain amount ($\lessapprox 10ms^{-1}$).
- ✗ As shown in [OSJ+05], an additional ultrasonic device on one of the user's shoulder enables additional constraint considerations.

Errors typically occur in sequences of less than three to four samples. Thus we apply a rather naive error correction: overwriting the current measurement by the last known *good* measurement sample.

4.4.3 Least squares optimization

A trilateration problem can be seen as an optimization task solving a specific objective function. The positions of the stationary devices are given by a set of points $\mathbf{B}_i$ with known coordinates (B_{ix}, B_{iy}, B_{iz}) with $i \in [1 \dots M]$ and $M > 3$. Furthermore there are M given distance measurements D_i between these points and a point $\mathbf{P}$ with unknown coordinates $[x, y, z]^T$. The coordinates of point $\mathbf{P}$ shall be optimized with respect to the distance measurements D_i. The distance r_i between the optimized coordinates of point $\mathbf{P}$ and the known point $\mathbf{B}_i$ can be calculated according to

$$r_i = \|\mathbf{B}_i - \mathbf{P}\| . \tag{4.1}$$

One strategy could be to minimize all the differences between the distances measured and the theoretically resulting distances in case the distance measurements were taken at the final assumption we make for the coordinates of point $\mathbf{P}$. Before summing up, these distances are squared following the *Gauss-Markov theorem*. Thus

the optimization task at hand is to minimize the squared differences between the measured distances D_i and the resulting distances r_i and thus the optimization problem can be written as

$$f(x, y, z) = \sum_{i}^{M} (r_i - D_i)^2 \rightarrow \min \tag{4.2}$$

which is the error or objective function of the optimization problem. This optimization problem can then be solved using a standard numerical optimization strategy (e.g. the Gauss-Newton algorithm). A major drawback of this solution is the missing ability to deal with measurement errors caused by occlusions or reflections, thus additional efforts must be made to filter these errors, as e.g. suggested in section 4.4.2 on the next page.

4.4.4 Kalman filtering

Kalman filtering is supposed to deal with error measurements far better than constraint considerations as described in 4.4.2 on the opposite page. This and the following section will focus on how to apply this algorithm in our contemplated scenario, for a quick reference on the Kalman filter see Appendix A on page 135.

The Kalman filter is an algorithm which makes optimal use of imprecise data in a linear system. It continuously updates the best estimate of the system's current state. The algorithm assures that all seen measurements influence the current state estimate. The resulting state estimation is ideal from a probabilistic point of view, i.e. it finds the best state estimate given a series of measurements. It should perform similar to a *least square optimization* algorithm (LSQ) (see Section 4.4.3) considering the accuracy of a single result, but far better given a series of measurements – even more so from the dynamics point of view.

The system must be defined by a state transition model and a measurement model. In case either one is non-linear, the Kalman filter in its original definition cannot be applied. The extended Kalman filter is one possible way to linearize such models.

In position estimation applications the state vector is usually set to

$$\mathbf{x} = [x, y, z, \dot{x}, \dot{y}, \dot{z}]^T \tag{4.3}$$

where x, y and z are the coordinates of the current position vector and the dot accent notates the first derivative and thus in this case the speed in the appropriate directions. Then the state transition matrix, see Eq. (A.1), can be defined according to

$$\mathbf{F}_k = \mathbf{F} = \begin{bmatrix} \mathbf{I}_3 \cdot \Delta t & \mathbf{I}_3 \\ \mathbf{0}_3 & \mathbf{I}_3 \end{bmatrix}, \ \forall\ k \in [-\infty, \infty] \tag{4.4}$$

with $\mathbf{I}_3$ and $\mathbf{0}_3$ being the 3×3 identity and 3×3 zero matrix, respectively. The distance d_i between an ultrasonic transmitter with coordinates $\mathbf{X}$ and the i^{th} fixed device with known coordinates $\mathbf{B}_i$ is given according to

$$d_i = \|\mathbf{B}_i - \mathbf{X}\| \tag{4.5}$$

which is a non-linear relation. Thus we have to apply the extended Kalman filter. Let the measurement vector be given by

$$\mathbf{z}_k = \left[d_{1,k}^2, d_{2,k}^2, d_{3,k}^2, d_{4,k}^2\right]^T \tag{4.6}$$

with $d_{i,k}$ being the measured distance to the fixed device i at time t_k. From Eq. 4.5 and Eq. 4.6 the measurement model can be derived according to

$$h(\mathbf{x}_k) = [x_k, y_k, z_k] \cdot [x_k, y_k, z_k]^T - 2 \cdot \mathbf{C} + \text{diagonal}(\mathbf{C} \cdot \mathbf{C}^T) \tag{4.7}$$

with $\mathbf{C}$ defining the coordinates of the fixed devices according to

$$\mathbf{C} = [\mathbf{B}_1, \mathbf{B}_2, \mathbf{B}_3, \mathbf{B}_4]^T \tag{4.8}$$

$$\mathbf{B}_i = [B_{i,x}, B_{i,y}, B_{i,z}]^T \tag{4.9}$$

According to Eq. (4.4) the linear state transition function f is given by

$$f(\mathbf{x}) = \begin{bmatrix} \mathbf{I}_3 \cdot \Delta t & \mathbf{I}_3 \\ \mathbf{0}_3 & \mathbf{I}_3 \end{bmatrix} \cdot \mathbf{x} \tag{4.10}$$

The force function $\mathbf{u}_k$ can be set to

$$\mathbf{u}_k \equiv \mathbf{u} \equiv \mathbf{0} \tag{4.11}$$

Several choices have to be made including initialization values, system noise modeling, and measurement error modeling. The choices for the initialization of the state estimation $\mathbf{x}_{0|0}$ and $\mathbf{P}_{0|0}$ influence the time the filter needs to converge to a feasible position estimation series and are thus rather unimportant because the filter will converge within a couple of seconds. Setting initialization values to $\mathbf{0}$ is a quite decent strategy. The choices for $\mathbf{Q}_k$ and $\mathbf{R}_k$ influence the performance of the filter during runtime according to its dynamic response and its immunity to measurement errors. In our case these covariance matrices are set to constant diagonal matrices.

Figure 4.3 on page 64 exemplifies the performance of the proposed Kalman filter design in comparison with the LSQ approach.

4.4.5 Kalman filter based fusion of absolute and inertial position measurements

A lot research has been carried out how to blend inertial tracking data with absolute position measurements. Actually the field of sensor fusion has become a major application for the Kalman filter. In most cases the absolute measurement system assures high accuracy whereas the inertial measurement system assures high dynamic response.

In the car assembly case study (see Section 3.2 on page 30) we use inertial measurement units (IMUs) to track the upper body motions. The IMUs are sampled at $100Hz$ and comprise three types of sensors:

- ✗ acceleration,
- ✗ gyroscope, i.e. rate of turn, and

- ✗ magnetic field sensors, for an absolute reference of orientation.

Each of these sensing modalities is implemented in three dimensions. This sensor system assures a setup that is able to measure the absolute orientation of the sensor platform.[8]

However, this sensor system also provides an inertial positioning system. The linear accelerometers measure acceleration $\mathbf{a}$ due to rate of velocity change plus acceleration $\mathbf{g}$ due to gravitation, both in sensor coordinates. Thus the sensor output is a vector $\mathbf{a} - \mathbf{g}$. The rate of turn sensors give the attitude of the sensor coordinate system, which can then be used to rotate both $\mathbf{a}$ and $\mathbf{g}$ from sensor coordinates into global coordinates. Due to the fact that the sensor system uses the magnetic field readings as aiding source for orientation estimations one can also use the orientation information directly to rotate the acceleration vectors.

In a next step the gravitational acceleration is added to the sensor output and the result has to be integrated twice to receive the position. Such a setup is called *inertial navigation system* (INS).

The obvious drawback of an INS is the fact that via double integration of the acceleration the measurement errors are also integrated twice. As stated before, typically an absolute positioning system is used to correct this error; in our case the UPS will be the guiding positioning system.

[Roe06] also follows such an approach, but with two major simplifications compared to the setup at hand. The absolute positioning system – an optical tracking system – samples at $120Hz$. After re-sampling to $100Hz$ both sensor systems run at the same sampling frequency. In our setup the INS runs at $100Hz$ and the UPS at approximately $1.4Hz$. The next difference are the error sources: In both setups the error source of the INS is assumed to be white noise though the error sources for the absolute positioning systems differ. [Roe06] assumes that the optical tracking system sometimes fails to get a measurement due to occlusions, that results in missing samples. In our case we also assume reflections which may result in totally wrong distance measurements, which cannot be modeled as Gaussian white noise. Nonetheless, the adaptation of a Kalman filter fusion algorithm seems to be promising for the following reasons:

- ✗ The position estimations will be up-sampled from to $100Hz$, which fits the speed of human arm motions better; and lets one anticipate better dynamics in the final position estimations.
- ✗ Reflection and occlusion errors are intended to be corrected more easily.

Following the recommendations given in [BH97, Roe06] we set up a complementary Kalman filter. The complementary filter approach is a state-of-the-art method in navigation when combining different sensor sources due to the following reasons:

[8] Evidently, this sensor setup requires again a decent method to blend the different sensor results. And, in fact, this platform is a good example of a Kalman filter application. A good overview of how it is applied in this scenario and beyond gives the thesis of Roetenberg [Roe06]. The author follows the recommendations given in [BH97], which is also the basic source for the Kalman filter questions in this thesis.

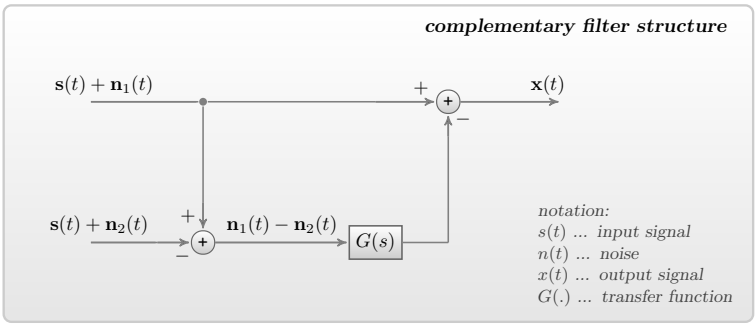

Figure 4.1
Complementary filter scheme – *In case $G(s)$ can be designed to separate the errors of both input signals, the resulting output $\mathbf{x}(t)$ will be approximately the actual input signal $\mathbf{s}(t)$.*

- ✗ The complementary approach usually reduces the grade of non-linearity, due to the fact that the state vector is built by the sum of inertial and aiding errors.
- ✗ It has got a better dynamic response but at the same time has got a small time delay.
- ✗ Generality – it scales for various mixes of aiding sources.

Figure 4.1 depicts the basic concept of the complementary filter approach. This section will outline the way we implemented this method; Figure 4.2 on the next page gives an overview of our implementation.

As depicted in Figure 4.1, the input of the complementary Kalman filter is equal to the difference of the errors of both positioning systems. The setup at hand has got a major drawback; both positioning devices use a Kalman filter itself. Thus our filter setup comprises a *feedback-less, decentralized filter* approach with a complementary centralized Kalman filter with a feedback loop as master filter. The decentralized filter approach is a rather simple approach with two problems:

- ✗ The full order state vector is typically not available to all filters, which causes a measurement information loss in the master filter.
- ✗ The estimation errors of the local filter outputs are usually correlated in time. This correlation is usually unknown and thus may cause a divergence in the master filter.

As Figure 4.2 illustrates, the inertial position estimation is done by integrating the (*rotated and gravitation free*) acceleration measurements twice, which are sampled at $100Hz$. The UPS is sampled at $\simeq 1.4Hz$. Thus UPS velocity and acceleration estimations must be compared with the average acceleration and average velocity of the INS results of the same time interval. The error model derived below tries to account for that.

We consider the time interval

$$T = [t_k, t_{k+1}] = [\tau_\kappa, \tau_{\kappa+s}] \tag{4.12}$$

with t_i being the points in time where the UPS is sampling, and τ_ι being the points in time where the INS is sampling.[9] In an analogous manner we define $\Delta\tau$ as the

[9] t_k is actually not equal to τ_κ and t_{k+1} is not equal to $\tau_{\kappa+s}$. But due to the way the data recording system (see [BAL08, BKL06]) was set up to fuse these raw sensor streams – i.e. triggering at the highest sampling data input, in our case at τ_κ – these time differences get lost anyway and Eq. (4.12) gets valid.

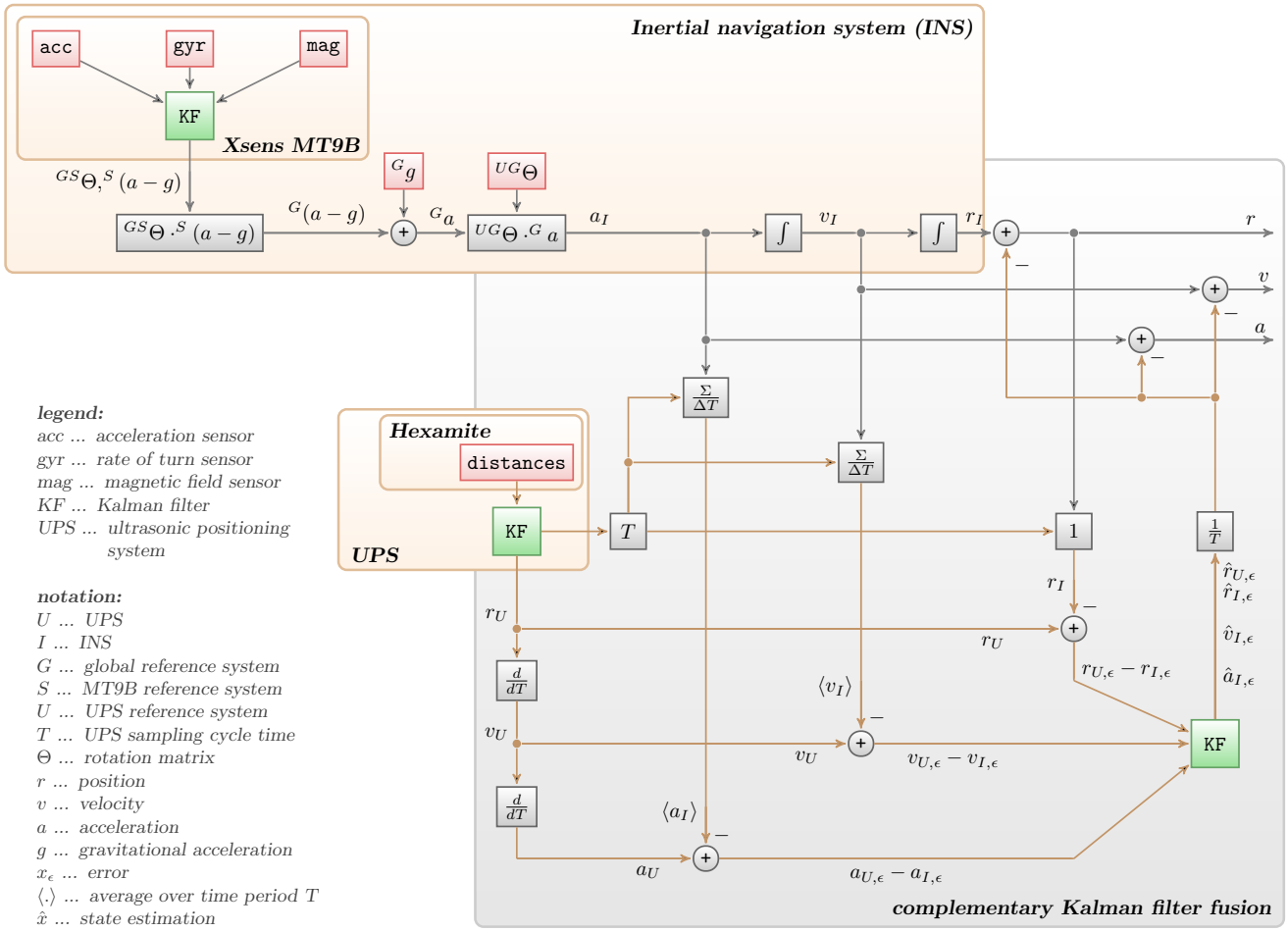

Figure 4.2

Complementary Kalman filter for blending the inertial positioning result with the aiding ultrasonic positioning sensor. The gray net depicts signals running at orientation sensor sampling rate, i.e. $100Hz$. The orange net depicts signals running at UPS sampling rate, i.e. $\simeq 1.4Hz$.
The Kalman filter for the mere ultrasonic based position estimation was set up according to the description in Section 4.4.4 on page 57.

time difference between two consecutive INS samples and Δt as the time difference between two consecutive UPS samples, i.e. $\Delta\tau = 0.01s$ and $\Delta t \simeq 0.7s$. The UPS position estimation at time t_k denoted as $\mathbf{r}_{U,t_k}$ can be written as the sum of the true position at time t_k and the error of the UPS result

$$\mathbf{r}_{U,t_k} = \mathbf{r}_{t_k} + \mathbf{r}_{U\epsilon,t_k} \tag{4.13}$$

The same can be written for acceleration and velocity

$$\mathbf{v}_{U,t_k} = \mathbf{v}_{t_k} + \mathbf{v}_{U\epsilon,t_k} \tag{4.14}$$

$$\mathbf{a}_{U,t_k} = \mathbf{a}_{t_k} + \mathbf{a}_{U\epsilon,t_k} \tag{4.15}$$

Analogous to that we can define for the INS

$$\mathbf{a}_{I,\tau_\kappa} = \mathbf{a}_{\tau_\kappa} + \mathbf{a}_{I\epsilon,\tau_\kappa} \tag{4.16}$$

and thus the INS velocity estimation can be calculated according to

$$\begin{aligned}\mathbf{v}_{I,\tau_\kappa} &= \mathbf{v}_{I,\tau_{\kappa-1}} + \mathbf{a}_{I,\tau_{\kappa-1}} \cdot \Delta\tau = \\ &= \mathbf{v}_{\tau_{\kappa-1}} + \mathbf{v}_{I\epsilon,\tau_{\kappa-1}} + \mathbf{a}_{\tau_{\kappa-1}} \cdot \Delta\tau + \mathbf{a}_{I\epsilon,\tau_{\kappa-1}} \cdot \Delta\tau = \\ &= \mathbf{v}_{\tau_\kappa} + \mathbf{v}_{I\epsilon,\tau_{\kappa-1}} + \mathbf{a}_{I\epsilon,\tau_{\kappa-1}} \cdot \Delta\tau\end{aligned} \tag{4.17}$$

With

$$\mathbf{v}_{I,\tau_\kappa} = \mathbf{v}_{\tau_\kappa} + \mathbf{v}_{I\epsilon,\tau_\kappa} \tag{4.18}$$

the velocity error of the INS at time τ_κ can then be written as

$$\mathbf{v}_{I\epsilon,\tau_\kappa} = \mathbf{v}_{I\epsilon,\tau_{\kappa-1}} + \mathbf{a}_{I\epsilon,\tau_{\kappa-1}} \cdot \Delta\tau \tag{4.19}$$

Analogous to that we can derive the INS position estimation

$$\begin{aligned}\mathbf{r}_{I,\tau_\kappa} &= \mathbf{r}_{I,\tau_{\kappa-1}} + \mathbf{v}_{I,\tau_{\kappa-1}} \cdot \Delta\tau + \mathbf{a}_{I,\tau_{\kappa-1}} \cdot \frac{\Delta\tau^2}{2} = \\ &= \mathbf{r}_{\tau_{\kappa-1}} + \mathbf{r}_{I\epsilon,\tau_{\kappa-1}} + (\mathbf{v}_{\tau_{\kappa-1}} + \mathbf{v}_{I\epsilon,\tau_{\kappa-1}}) \cdot \Delta\tau + (\mathbf{a}_{\tau_{\kappa-1}} + \mathbf{a}_{I\epsilon,\tau_{\kappa-1}}) \cdot \frac{\Delta\tau^2}{2} = \\ &= \mathbf{r}_{\tau_\kappa} + \mathbf{r}_{I\epsilon,\tau_{\kappa-1}} + \mathbf{v}_{I\epsilon,\tau_{\kappa-1}} \cdot \Delta\tau + \mathbf{a}_{I\epsilon,\tau_{\kappa-1}} \cdot \frac{\Delta\tau^2}{2}\end{aligned} \tag{4.20}$$

and its error at time τ_κ

$$\mathbf{r}_{I\epsilon,\tau_\kappa} = \mathbf{r}_{I\epsilon,\tau_{\kappa-1}} + \mathbf{v}_{I\epsilon,\tau_{\kappa-1}} \cdot \Delta\tau + \mathbf{a}_{I\epsilon,\tau_{\kappa-1}} \cdot \frac{\Delta\tau^2}{2} \tag{4.21}$$

To account for the different sampling rates we define the following average values for acceleration and velocity over time interval T as defined by Eq. (4.12)

$$\begin{aligned}\langle \mathbf{a}_I \rangle_T &= \sum_{j=0}^{s-1} \mathbf{a}_{I,\tau_{\kappa+j}} = \sum_{j=0}^{s-1} (\mathbf{a}_{\tau_{\kappa+j}} + \mathbf{a}_{I\epsilon,\tau_{\kappa+j}}) = \\ &= \langle \mathbf{a} \rangle_T + \langle \mathbf{a}_{I\epsilon} \rangle_T\end{aligned} \tag{4.22}$$

$$\begin{aligned}\langle \mathbf{v}_I \rangle_T &= \mathbf{v}_{I,\tau_\kappa} + (\langle \mathbf{a} \rangle_T + \langle \mathbf{a}_{I\epsilon} \rangle_T) \cdot \Delta\tau = \\ &= \mathbf{v}_{\tau_\kappa} + \mathbf{v}_{I\epsilon,\tau_\kappa} + \langle \mathbf{a} \rangle_T \cdot \Delta\tau + \langle \mathbf{a}_{I\epsilon} \rangle_T \cdot \Delta\tau = \\ &= \langle \mathbf{v} \rangle_T + \langle \mathbf{v}_{I\epsilon} \rangle_T\end{aligned} \tag{4.23}$$

Thus the position estimation at time t_{k+1} can be written according to

$$\begin{aligned}\mathbf{r}_{I,t_{k+1}} &= \mathbf{r}_{I,t_k} + \langle \mathbf{v}_I \rangle_T \cdot \Delta\tau + \langle \mathbf{a}_I \rangle_T \cdot \frac{\Delta\tau^2}{2} = \\ &= \mathbf{r}_{t_k} + \mathbf{r}_{I\epsilon,t_k} + (\langle \mathbf{v} \rangle_T + \langle \mathbf{v}_{I\epsilon} \rangle_T) \cdot \Delta\tau + (\langle \mathbf{a} \rangle_T + \langle \mathbf{a}_{I\epsilon} \rangle_T) \cdot \frac{\Delta\tau^2}{2} = \\ &= \mathbf{r}_{t_{k+1}} + \mathbf{r}_{I\epsilon,t_{k+1}}\end{aligned} \tag{4.24}$$

The state vector $\mathbf{x}$ of the complementary Kalman filter as depicted in Figure 4.2 on the previous page is set to

$$\mathbf{x} = \begin{bmatrix} \mathbf{r}_{I\epsilon} \\ \mathbf{v}_{I\epsilon} \\ \mathbf{a}_{I\epsilon} \\ \mathbf{r}_{U\epsilon} \\ \mathbf{v}_{U\epsilon} \\ \mathbf{a}_{U\epsilon} \end{bmatrix} \tag{4.25}$$

and finally we can define the state transition model according to

$$\mathbf{F} = \begin{bmatrix} \mathbf{I}_3 & \Delta t \cdot \mathbf{I}_3 & \frac{\Delta t^2}{2} \cdot \mathbf{I}_3 & & & & \\ & \mathbf{I}_3 & \Delta t \cdot \mathbf{I}_3 & & & & \\ & & e^{-\beta_{\langle a \rangle} \cdot \Delta t} \cdot \mathbf{I}_3 & & & & \\ & & & e^{-\beta_{\langle a \rangle} \cdot \Delta t} \cdot \mathbf{I}_3 & & & \\ & & & \frac{1-e^{-\beta_{\langle a \rangle} \cdot \Delta t}}{\Delta t} \cdot \mathbf{I}_3 & & & \\ & & & \frac{1-e^{-\beta_{\langle a \rangle} \cdot \Delta t}}{\Delta t^2} \cdot \mathbf{I}_3 & -\frac{1}{\Delta t} \cdot \mathbf{I}_3 & \mathbf{0}_3 \end{bmatrix} \tag{4.26}$$

As can be seen from Eq. (4.26) the mere measurement errors are modeled as Gauss-Markov processes. The parameters for these processes are tuned according to experimental test runs, as well as the values for $\mathbf{Q}$ and $\mathbf{R}$.

As already explained, the input to a complementary filter is not the measurements but the difference between the two measurement systems, i.e. the input can be written as

$$\begin{aligned} \mathbf{z} &= \begin{bmatrix} \mathbf{r}_{U,t_k} - \mathbf{r}_{I,t_k} \\ \mathbf{v}_{U,t_k} - \mathbf{v}_{I,t_k} \\ \mathbf{a}_{U,t_k} - \mathbf{a}_{I,t_k} \end{bmatrix} = \begin{bmatrix} (\mathbf{r}_{t_k} + \mathbf{r}_{U\epsilon,t_k}) - (\mathbf{r}_{t_k} + \mathbf{r}_{I\epsilon,t_k}) \\ (\mathbf{v}_{t_k} + \mathbf{v}_{U\epsilon,t_k}) - (\mathbf{v}_{t_k} + \mathbf{v}_{I\epsilon,t_k}) \\ (\mathbf{a}_{t_k} + \mathbf{a}_{U\epsilon,t_k}) - (\mathbf{a}_{t_k} + \mathbf{a}_{I\epsilon,t_k}) \end{bmatrix} = \\ &= \begin{bmatrix} \mathbf{r}_{U\epsilon,t_k} - \mathbf{r}_{I\epsilon,t_k} \\ \mathbf{v}_{U\epsilon,t_k} - \mathbf{v}_{I\epsilon,t_k} \\ \mathbf{a}_{U\epsilon,t_k} - \mathbf{a}_{I\epsilon,t_k} \end{bmatrix} \end{aligned} \tag{4.27}$$

and hence the observation model is given according to

$$\mathbf{H} = \begin{bmatrix} -\mathbf{I}_9 & \mathbf{I}_9 \end{bmatrix} \tag{4.28}$$

4.5 Result

Figure 4.3 on the following page depicts the results of both Kalman filters. Positions as estimated from mere ultrasonic distance readings by means of an extended Kalman filter (as defined in Section 4.4.4 on page 57) are illustrated. Additionally the results of the *complementary* Kalman filter fusing INS data (i.e. acceleration and gyroscope) with UPS distance readings are shown. The acceleration depicted here is not the same signal as that shown in Figure 3.2 on page 34. This is due to the fact that the acceleration readings from the INS have to be rotated from the local INS sensor reference system to the global UPS reference system preliminary to the complementary Kalman filter stage. Furthermore the earth acceleration has to be removed from the acceleration signal.

As the resulting trajectories are not the final results we are interested in but are further applied for probabilistic location modeling (see Chapter 6) no detailed quantitative evaluations of the resulting trajectories have been conducted so far. Thus closed loop tests or similar evaluations remain an open issue.

Another possibility to test the performance of a specific Kalman filter design is to evaluate the resulting time series of residuals $\tilde{\mathbf{y}}_k$ (innovations) derived according to Eq. (A.8). A decent filter design will result in a white innovation series, i.e. a normally distributed series with zero mean and equally distributed frequencies. Figure 4.4 on page 66 evaluates three out of nine innovation series ($r_{xU\epsilon} - r_{xI\epsilon}$,

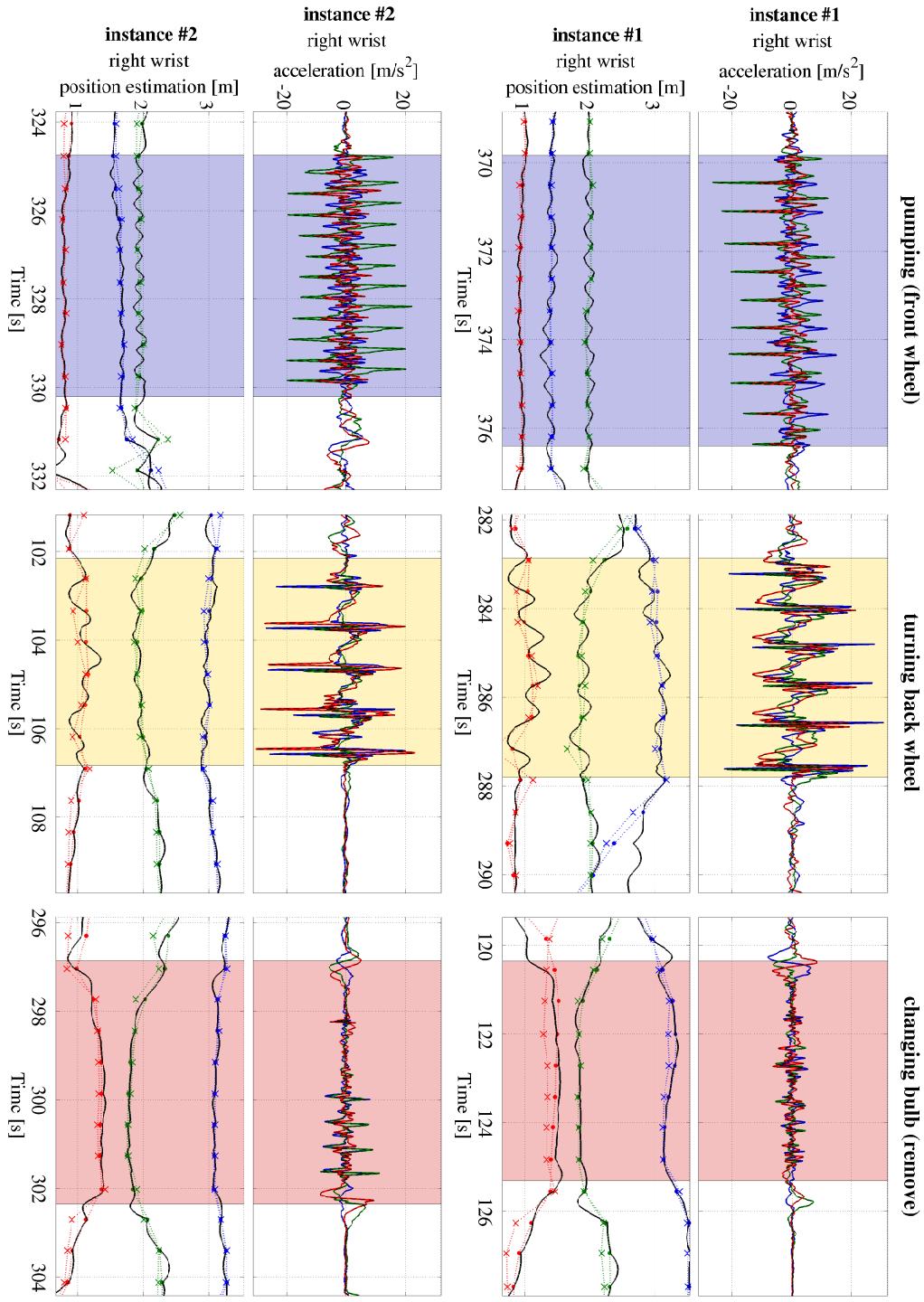

Figure 4.3

Results of three different state estimation approaches – *Position estimations using an LSQ approach are depicted by the cross-marked lines*, extended *Kalman filter based estimation of mere ultrasonic distance readings are depicted by dot-marked lines. The crosses and the big dots mark those points in time where the UPS provides its readings. The black graphs depict the results of the* complementary *Kalman filter. The acceleration plots also depict the acceleration state estimation of the complementary Kalman filter (black dashed lines).*

$r_{yU\epsilon} - r_{yI\epsilon}$, and $r_{zU\epsilon} - r_{zI\epsilon}$) for the complementary Kalman fusion filter applied on the entire *sequence* data set of the bicycle maintenance case study. The upper plots compare the histogram of innovation amplitudes with the normal distribution

$$N\left(0, \sqrt{\frac{1}{n}\sum_{k=0}^{n}\left(r_{iU\epsilon,k} - r_{iI\epsilon,k} - 0\right)^2}\right) \tag{4.29}$$

whereas the lower figures give the FFT based calculation of the spectrum of the innovations and a low-pass filtered version of the spectrum. The figures suggest that though the histograms are quite similar to normal distribution and the spectrum is similar to that of a white sequence there might still be some potential for enhancements in the Kalman fusion filter design.

4.6 Future work

Note that Kalman filtering assumes a Gaussian state transition estimation error and a Gaussian measurement error. The ultrasonic distance measurements can be assumed to be Gaussian. Actually, the measurement errors are only Gaussian given a static distance, i.e. the variance of the distance measurements increases with the distance. A proper implementation accounts for this by modeling the variance of the current measurement depending on the current distance estimation. Due to the fact that errors resulting from reflections cannot be modeled as Gaussian error the actual measurement error consists of two portions: a Gaussian measurement error and a non-Gaussian error resulting from sporadically occurring reflections. Thus the presented position estimation approach could benefit from either applying a constrained Kalman filter to exclude *impossible* states reflecting considerations similar to those listed in Section 4.4.2 on page 56; (refer to [GH07] or [UDL07] for such an approach) or by applying a particle filter (see e.g. [AMGC01]).

4.7 Summary

After giving an introduction to indoor positioning and location tracking techniques the chapter presented different position estimation techniques capable of deriving hand position trajectories from erroneous ultrasonic distance readings. Due to the fact that the sampling rate provided by a typical UPS is too slow to estimate trajectories reflecting the dynamics of hand motions, a complementary Kalman fusion filter was set up. By means of this Kalman filter motion and position readings can be fused to a hand trajectory providing both decent position estimations and dynamic trajectory estimations.

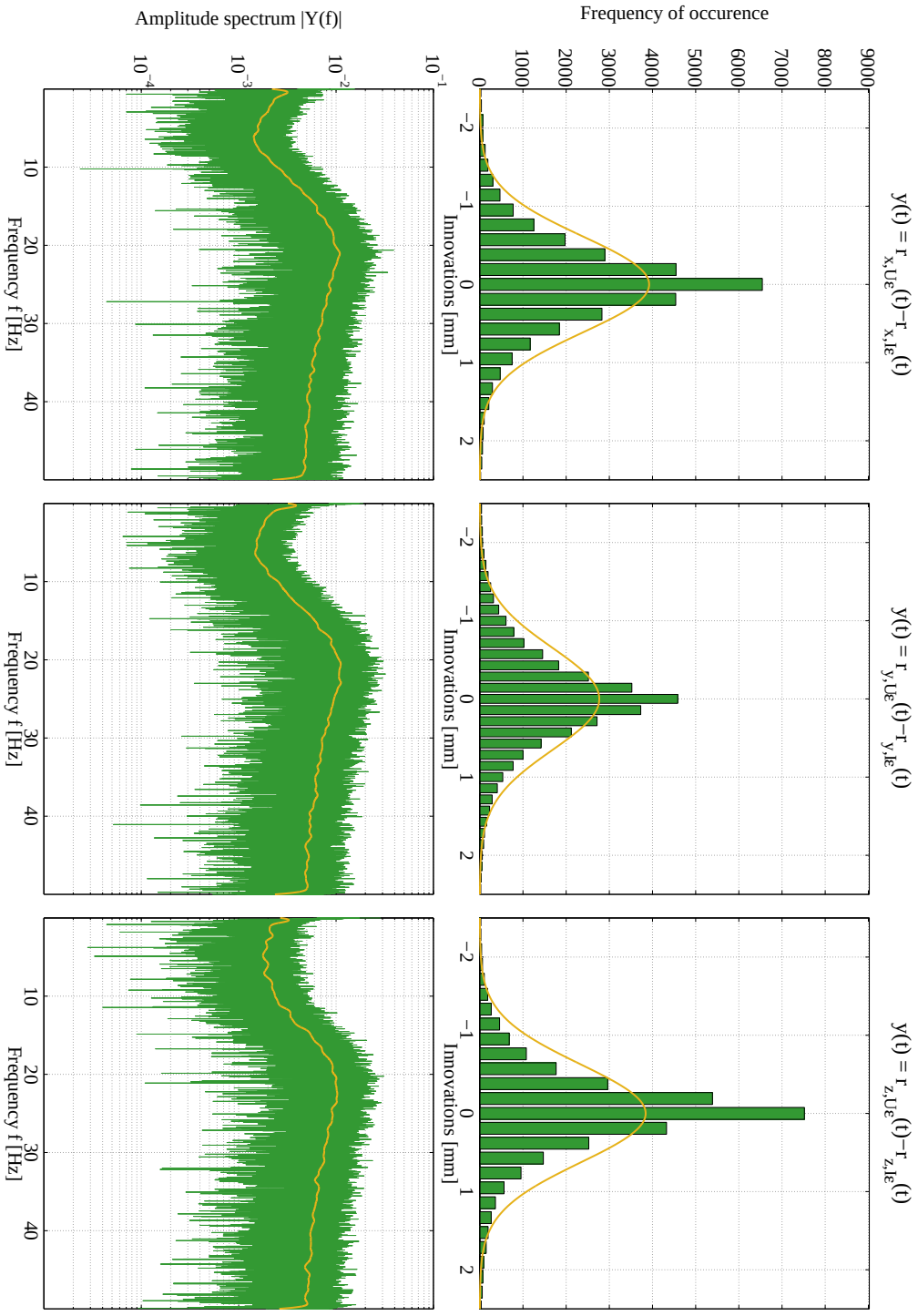

Figure 4.4

Evaluations of the Kalman filter innovations for the complementary *Kalman filter defined in Figure 4.2 on page 61.*

Chapter

5

Location modeling

This chapter defines various probabilistic methods in order to model hand locations. The training of these location models can be done either in a supervised or in a semi-supervised manner. Appropriate distance measures are defined for each location modeling approach.

The methods are tested and evaluated on the bicycle maintenance test scenario. The chapter gives detailed evaluation results for pre-segmented activities in order to foster the combination of hand location and hand orientation for the use of gesture recognition.

5.1 Introduction

This chapter demonstrates how, despite the preliminary described sensing problems, ultrasonic positioning can be used to improve the accuracy of manipulative gesture recognition. Specifically this chapter presents the following contributions

- ✗ We describe and contrast different ways of modeling locations of interest (including location distance metrics) in a supervised or semi-supervised manner.
- ✗ We describe and contrast different ways of combining the location information with motion information from orientation sensors at the user's arms given pre-segmented activities.
- ✗ We present the results of an experimental validation of our method. It is based on a bicycle maintenance task that has been repeatedly performed by six volunteers, see also Section 3.2 on page 30. The task consists of manipulative hand gestures that were chosen according to two criteria:
  - ✗ being typical for the maintenance task and
  - ✗ being ambitious in terms of recognition.
- ✗ One of the most significant results is the fact that our method can handle user-dependent training as well as user-independent without a significant difference in the performance.

5.2 From position to location

In order to incorporate position information within the activity and gesture recognition process, the position measurements are mapped to some abstract location representation. Within this work we will use the words *position* and *location* to distinguish between these two representations; thus

- ✗ *position* will refer to a place given by its absolute coordinates and
- ✗ *location* and *location class* will refer to a place defined by some abstract location information, i.e. defined on the basis of an abstract spatial correlation, e.g. *close to a specific object.*

The methods and considerations described in this section have to be applied for each position sensor or rather for each hand separately.

5.2.1 Manual location definitions

In order to define the location of interest for a specific activity one might use the coordinates of the object which is being manipulated. There are activities where this strategy might be effective, e.g. *tightening a screw.* The coordinates of the screw are known and its spatial expansion is approximately within the accuracy of the position measurements. Thus the coordinates of the screw head's center of gravity might be sufficient to define the location of this action. Unfortunately, we are not able to measure the cone end of the screwdriver, but the position of the wrist of the user's hand. Depending on the way the user holds the tool and on the

size of the tool itself there is a distance of up to 20 or 30cm between the location of interest and the measured position and, what is more, the wrist is not standing still but performing a circular movement. However, for this simple task it is possible to define a point around which the wrist will be rotating most likely; but for activities with more complex motions, e.g. *turning a wheel* or *pumping*, this manual location class definition will most likely fail because the location of interest

- ✗ is not equal to the positioning of the manipulated device and
- ✗ has got a spatial expansion, which is far from being point-shaped or even sphere-shaped.

5.2.2 Supervised location definitions

Much more promising seem to be approaches where the locations can be *learned* in a semi-supervised or supervised manner. Such a supervised way was already described in [SOJ+06]. Each hand gesture (*gesture class*) is manually assigned to one of the manually defined *location classes*, see e.g. Table 5.1 on page 75 columns 2, 3, 6, and 7. For both hands mean and variances are modeled for these locations according to the training data; i.e. each location is modeled as a *multivariate normal distribution* which better considers the nature of the locations of interest.

Although a multivariate normal distribution can model the spatial expansion of the location of interest it cannot model arbitrary shapes. Thus other distributions could also be helpful. To this end we will test multivariate Gaussian mixture distributions.

A n-component Gaussian mixture distribution is defined according to

$$f(\mathbf{x}) = \sum_{k=1}^{n} a_k f_k(\mathbf{x}) = \sum_{k=1}^{n} a_k \Phi(x|\mu_\mathbf{k}, \mathbf{\Sigma_k}) \tag{5.1}$$

where $f_k(\mathbf{x}) = \Phi(\mathbf{x}|\mu_\mathbf{k}, \mathbf{\Sigma_k})$ is the k^{th} normal distribution[1] and a_k its *prior probability*. In case the number of mixture components is high enough, such a distribution is able to simulate any possible distribution and thus any possible *shape*.

For consistency reasons we will use the following denotations:

- ✗ 1-*component Gaussian mixture distribution (supervised)* (1-m-s) will refer to the first and
- ✗ n-*component Gaussian mixture distribution (supervised)* (n-m-s) to the second supervised location definition method.

To fit a mixture distribution to the training data for one location class the *expectation-maximization (EM)* algorithm [Bil98] is used. The EM algorithm has to fit the prior probability a, the mean μ and the covariance matrix $\mathbf{\Sigma}$ for any component. In addition we do not use a constant value for n but a range of values. Hence for each permitted number of components a separate mixture distribution has to be fitted. Furthermore it has to be decided which distribution fits best according

[1] In general mixture distributions can use any distribution function, thus $f_k(\mathbf{x})$ does not have to be a normal distribution but can also be of any other kind.

to a given criterion. The *Akaike* or the *Bayes* information criteria are typically used as determining criteria.

For the presented case study the number of components n is restricted to the range of 1 to 5. To decide which component fits best we apply the Bayes information criterion. Before fitting the distribution with the EM algorithm, the means are initialized with the k-means cluster algorithm. Different additional information is further defined manually, e.g. which activity is performed with the right, the left, or both hands, see Table 5.1 columns 4, 5, 8, and 9.

5.2.3 Semi-supervised location definitions

The semi-supervised location class definition is based on an automatic clustering method. The location training data is clustered using a Gaussian mixture cluster algorithm and furthermore each resulting location class is assigned one or more gesture classes in a probabilistic manner.

This cluster algorithm fits a n-component Gaussian mixture distribution to the training data. To fit this mixture distribution the EM algorithm is used. Once again we do not use a constant value for n but a range of values. Hence for each permitted number of components a separate mixture distribution has to be fitted.

In case we decided on a mixture distribution, we need an assignment between gesture classes and location models. Each training sample $\mathbf{x}$ of gesture class i is assigned to a specific component of the fitted mixture distribution by means of assigning each sample $\mathbf{x}$ the component with the largest posterior probability of $\mathbf{x}$. Accumulating the assignments per component results in a n-fold assignment histogram for each gesture class i, and hence in a $g \times n$ matrix $\mathbf{A}$, with g being the number of gesture classes and n being the number of components of the Gaussian mixture.[2]

For the presented case study the number of components n is restricted to the range $n \in [10, 25]$. The number of components finally used is then defined according to

$$n_{final} = \arg\max_k(\min(\{\max_i(\mathbf{A}^k_{i,1}), \ldots, \max_i(\mathbf{A}^k_{i,n})\})) \tag{5.2}$$

where $\mathbf{A}^k$ denotes the assignment table corresponding to a k-component-mixture-distribution. Before fitting the distribution with the EM algorithm the means are initialized with the k-means cluster algorithm. The algorithm is further fed with knowledge about whether an activity is accomplished with the right, the left, or both hands. In case an activity is performed with only one hand, location training data of the appropriate other hand must not be presented to the cluster algorithm. Hence again we apply a two-way location definition, one for each hand. The semi-supervised location modeling is done in two different schemes: one using the entire gesture class resolution and the other using the reduced resolution for the training. For spotting and classification both schemes are using the entire resolution. We will refer to these schemes as *semi-supervised Gaussian mixture location modeling*

[2]This assignment table $\mathbf{A}$ can be seen as a *probabilistic* or *continuous* generalization – with values ranging from 0 to 1 – of the manually defined *binary* or *discrete* assignment table, see Table 5.1 on page 75, columns 2 and 6.

(s-m-m) and *semi-supervised Gaussian mixture location modeling using the reduced gesture class resolution (s-m-r)*, respectively.

5.3 Distance measures

To assign each position sample or a sequence of samples a specific location class a distance measure is needed. It is quite obvious that the *manually assigned locations* simply afford the use of an *Euclidean distance* measure whereas both the *semi-supervised* and the *supervised* method make better use of more complex distance measures.

The Mahalanobis distance is able to consider the spatial expansion – specified by the trained distribution – of the location of interest. Thus we will apply the *Mahalanobis distance* for the 1-m-s approach, but also use the *Euclidean distance* for reasons of comparison. The 1-m-s method in combination with the Euclidean distance measure is used to simulate best-case *manual* location modeling. We refer to this method as *pseudo-manual* because it is an ideal version of the manual method defined in Section 5.2.1.

For the n-m-s approach we define the distance of sample $\mathbf{x}$ to location i analogous to the Mahalanobis distance according to

$$d_i(\mathbf{x}) = -ln(pdf_i(\mathbf{x})) \tag{5.3}$$

with $pdf_i(\mathbf{x})$ being the probability density function of the location model i evaluated at position sample $\mathbf{x}$.

For the semi-supervised location approach we calculate the posterior probability of component j given position sample $\mathbf{x}$ for all n components of the location model, thus we end up in a posterior probability vector $\mathbf{p}(\mathbf{x})$. The distance to *gesture* i of position sample $\mathbf{x}$ is then defined according to

$$d_i(\mathbf{x}) = -ln\left(\sum_{j=1}^{n}(p_j(\mathbf{x}) \cdot A_{i,j})\right) \tag{5.4}$$

with $\mathbf{A}$ being the $g \times n$ probabilistic gesture-to-location assignment table, as defined above; g being the number of gesture classes.

5.4 Decision boundary

To decide whether a specific position reading originated in a specific location or not, a decision boundary has to be defined. This decision boundary is represented by a threshold ϑ_{dist}, which can be assigned

- ✗ *manually* using a static threshold, or
- ✗ *automatically* during the training process resulting in a class-wise threshold.

In [SOJ+06] we trained the thresholds according to

$$\vartheta_{dist,i} = \mu_i + f \cdot \sigma_i \tag{5.5}$$

where μ_i is the mean value of all distances calculated during the training of location i and σ_i the corresponding standard deviation. f is a constant factor. Hence to decide in favor of location class i in case of position sample $\mathbf{x}$ the following constraint must be fulfilled:

$$d_i(\mathbf{x}) < \vartheta_{dist,i} \tag{5.6}$$

The constant factor f is optimized during a spotting test run on the training data set by applying the evaluation metric defined by Ward *et al.* [WLT06, War06], see also Appendix B on page 139.

For the results presented in this thesis we define – no matter which location method is applied – the decision boundary for sample $\mathbf{x}$ and class i according to the Mahalanobis distance

$$\sqrt{\left(d_i\left(\mathbf{x}\right) - \mu_i\right)^T \mathbf{\Sigma}_i^{-1} \left(d_i\left(\mathbf{x}\right) - \mu_i\right)} < \vartheta_{dist,i} \tag{5.7}$$

Where μ_i is the mean of all distances of the training samples and $\mathbf{\Sigma}_i$ the respective covariance matrix. $\vartheta_{dist,i}$ is defined according to

$$\vartheta_{dist,i} = f_i \tag{5.8}$$

with f_i being a constant factor trained as above but separately for each location class.

5.5 Recognition approach on pre-segmented activities

As the focus of this work lies on how subsidiary context sensing techniques can support motion sensing based gesture recognition considering a continuous sensor data stream, evaluations on pre-segmented activities, i.e. activity segmentation based on the ground-truth annotation during the experiment, can help supply proof of concept. Due to that this section summarizes the recognition approach as applied in this special evaluation scheme.

5.5.1 HMM based motion classification

We have picked *Hidden Markov Models (HMMs)* as a state-of-the-art classification approach to the motion classification problem. HMMs have proven [JLT04a, Jun05] to be an appropriate choice for modeling and recognition of the dynamically changing motions in our experiment.

HMMs [Rab89, Gha98, Gha01] became a sort of state-of-the-art method for (on-line) analysis of sequential time series data. HMMs have first been used in the field of speech recognition [Rab89] and are now used in many other domains. HMMs are probabilistic models that are defined over a finite number of states, their observations, and state transition probabilities. The advantages of HMMs are time invariance and their ability to deal with real world data. But with an increasing number of states, the complexity of such models can result in ineffectiveness due to an explosion of the number of parameters, unless the parameters can be reduced

through exhaustive planning, which demands an exact knowledge of the system that should be modeled.

Although HMMs are quite time-invariant they are unresistant regarding to underfill and overfill errors (see Appendix B.1 on page 140) in particular. While underfill errors can be handled a bit more easily, through assigning the states of the HMM equally distributed *prior* probabilities – what makes the model of course less restrictive – overfill errors may totally confuse the recognition process because unknown states are presented to the model. Thus this method seems inadequate for the recognition of gestures in the preliminary spotted segments:

- ✗ The spotting stage aims to be non-restrictive to ensure that *all* segments potentially containing a gesture are detected. Thus it will introduce a high overfill error rate.
- ✗ The idea of *location* spotting itself implies that the result will contain a high overfill error rate: the hand location is a strong indicator for a gesture to start OR a gesture to start in the near future and a spotted segment might also contain two or more gestures.

Nonetheless, for pre-segmented activities HMMs are a viable approach.

For the presented evaluation HMMs have been applied to the motion data according to the following principles. Each manipulative gesture in our experiment corresponds to an individually trained HMM. The analysis and evaluation of the number of states per model ranging from two to twelve resulted in determining the number of states from five to seven for the manipulative gestures explored in the given case studies. The number of states reflects the complexity of the respective manipulative gesture. We exclusively used so-called *left-right* models. A characteristic property of left-right models is that no transitions are allowed to states whose indices are lower than the current state.

As features for the HMMs, only raw inertial sensor data has been used. The employed set of features comprises the following subset of available sensor signals: three acceleration and two gyroscope signals from the user's right hand and three acceleration and two gyroscope signals originating at the user's right upper arm. The observations of the used HMMs correspond to the raw sensor signals or features. Their continuous nature is modeled by a single normal distribution for each state in all models.

5.5.2 Location classification

When recognizing pre-segmented activities the trained location models are the basis for location classification. A frame-based classification is applied based on the distances defined in Eqs. (5.3) and (5.4). We calculate the median distance for all position samples of the segment. The location class with the minimum median distance is taken as the location class result. The location class ranking is derived from the median distances to all location classes. For the supervised and the pseudo-manual method a gesture class ranking can be derived by assigning each gesture the same rank its appropriate location class has got.

5.5.3 Classifier fusion

This section summarizes different approaches to fusing motion and location based classification results in the pre-segmented recognition approach.

5.5.3.1 Ranking based fusion

A simple fusion method combines the rankings of both the position and the motion classifier. In general both classification stages produce a ranking starting with the most likely and ending with the least likely class. The final classification is then based on a combination of these two rankings. This combination can be done as follows:

- ✗ *average of best matching classes (aob)* comparing the average ranking of the top choices of both classifiers.
- ✗ *average (avg)* comparing the average ranking of all gesture classes.

5.5.3.2 Plausibility analysis

The most obvious fusion method is the use of wrist position information to constrain the search space of the motion based classifier. The motion based HMM classifier results in motion class ranking. We test the four most likely gesture classes according to the following criteria. Beginning with the most likely gesture concerning the motion result we analyze the plausibility concerning the location of this gesture class. In case the plausibility result fits the location class, the result is assumed to be the correct gesture class, otherwise the next candidate is tested. In case the whole set of possible candidates is tested and we end up with no plausible class, the position estimation is assumed to have failed for whatever reason and the most likely motion result is taken as a final gesture result.[3]

Whether the current gesture hypothesis is plausible concerning current position estimations is decided by calculating the median distance of all position samples of the tested segment to the trained location for the gesture class that is currently tested for plausibility. In case the median of these distances is below a certain threshold level, it is assumed to be a plausible gesture result.

5.5.3.3 Naive Bayes classifier fusion

This fusion method considers the *confusion matrix (CM)* that is achieved by running a *test run* on the pre-segmented data used for training.

In order to describe this in more detail we apply the following denotation: ω denotes an event. ω can be one out of the n relevant hand gestures, thus $\omega \in [\omega_1, \ldots, \omega_n]$. α denotes the result of classifier A for the spotted gesture and β the result of classifier B. The confusion matrices CM_A and CM_B belong to classifier A and B, respectively. From these confusion matrices we obtain an estimation for the probability that a classifier recognizes class ω_i although class ω_j is true, i.e. the probability $P(\alpha = \omega_i|\omega = \omega_j) = CM_{A,ij}$ and $P(\beta = \omega_i|\omega = \omega_j) = CM_{B,ij}$. Assuming that the classifier results are independent, one can show that

$$c = \arg\max_i([P(\alpha|\omega_i) \cdot P(\beta|\omega_i)]) \tag{5.9}$$

[3]For the continuous evaluation scheme we apply a similar concept also called plausibility analysis. The major difference is that in case of no plausible class the spotted location is assumed to contain no relevant gesture and thus the segment can be treated as falsely inserted.

Table 5.1
Gesture list for the bicycle maintenance case study. The table also lists additional information used to configure the recognition process.

	RIGHT HAND				LEFT HAND			
class ID	loc ID	loc description	loc relevant	act relevant	loc ID	loc description	loc relevant	act relevant
1	1	fw (bottom)	√	√	1	fw (bottom)	√	-
2	2	bw (bottom)	√	√	2	bw (bottom)	√	-
3	3	screw A	√	√	*N*	-	-	-
4			√	√	*N*	-	-	-
5	4	screw B	√	√	*N*	-	-	-
6			√	√	*N*	-	-	-
7	5	screw C	√	√	*N*	-	-	-
8			√	√	*N*	-	-	-
9	6	pedal	√	√	*N*	-	-	-
10			√	√	3	chain	√	-
11			√	√	4	gear-switch	√	-
12	7	bw (top)	√	-	5	pedal	√	√
13	8	fw (center)	√	√	6	fw (center)	√	√
14			√	√			√	√
15	*N*	-	-	-	7	fw (left)	√	√
16	9	bw (right)	√	√	*N*	-	-	-
17	*N*	-	-	-	8	bell	√	√
18	10	seat	√	√	9	seat	√	√
19			√	√			√	√
20	11	pedal (center)	√	√	10	pedal	√	-
21			√	√			√	-
22	12	close to bw	√	√	11	close to bw	√	√
23			√	√			√	√

legend:
loc ... location *fw ... front wheel*
act ... activity *bw ... back wheel*
N ... NULL

is the best fused result from the Bayesian point of view. Due to the assumption that has to be made this fusion method is called *naive* Bayesian classifier fusion, see also [KBD01]. They refer to this method as *Naive Bayes combination*.

5.6 Experimental results

5.6.1 Introduction

This section presents evaluations when working with pre-segmented activities, to fortify the usefulness of the combination of location and motion in the gesture recognition process. In addition to the fusion methods described in Section 5.5.3 on the next page we apply additional pseudo-fusion methods to visualize the potential capacity of the classifier fusion approach, see next section. Moreover the performance of different location modeling approaches are contrasted.

We use the experiment described in Section 3.2 on page 30 to evaluate the methods described within this chapter. Table 5.1 gives the list of gestures (as already named in Table 3.1 on page 33) and any additional information used to configure the spotting and recognition algorithms. This information comprises the manually assigned *location classes* (only used for methods *p-m*, 1-*m-s*, and *n-m-s*), and whether hand location or hand activity is relevant for this specific manipulative hand gesture.

To evaluate the ability of the approach to deal with inter-subject training and

recognition and in particular to fortify the assumption that location awareness is a comprehensively user-independent feature and therein outperforms motion based activity recognition, we adopt a threefold evaluation scheme:

- *Intra:* denoting the user-dependent training and recognition, i.e. the location and gesture recognition for a specific subject is adopting models trained only on test data of this subject.
- *Inter:* denoting the inter subject test and recognition scheme, i.e. the models are trained using data from *all* subjects.
- *External:* in this evaluation scheme the models used for recognizing gestures of a specific subject are trained using exclusively test data of all *other* subjects.

To ensure a comparable training of these three evaluation schemes and, what is more, to avoid over-fitting in the inter and external schemes, we use just a subset of all recorded training instances according to the following rules:

- *Intra:* the first 18 recorded instances per gesture and subject $\rightarrow 1 \times 18 = 18$ train instances per model.
- *Inter:* the first three recorded instances per gesture and subject $\rightarrow 6 \times 3 = 18$ train instances per model.
- *External:* the first four recorded instances per gesture and subject $\rightarrow 5 \times 4 = 20$ train instances per model.

The ultrasonic distance readings of both wrist-worn ultrasonic transmitters and the orientation, acceleration, and gyroscope readings of both MT9B devices worn on the lower arms (see also Figure 3.1 on page 32) are processed as described in Chapter 4 using the filter structure as depicted in Figure 4.2 on page 61 resulting in two wrist trajectories as exemplified in Figure 4.3 on page 64.

5.6.2 Classification results for pre-segmented activities

Table 5.2 on the next page summarizes the classification results, see Section 5.6.1 on the preceding page. The HMM classification of the motion readings is fused with different location results. The fusion methods are *average (avg)*, *average of best (aob)*, and *location plausibility analysis (pa)*, see Section 5.5.3 on page 74.

Note that HMMs are trained *gesture-class-wise* and locations are trained *location-class-wise*. Thus the performance is evaluated in the same manner. Due to that the recognition results of mere location recognition cannot be directly compared with either mere motion based or fused results.

Two additional fusion methods are applied. *Almighty* identifies a fake fusion algorithm: in case either one classifier is correct the result is counted as *correctly classified*. Whereas *binary* is a rather strict classifier fusion method: both classifiers *must* agree, otherwise it is assumed that the class cannot be correctly classified and thus no valid result is available. The almighty and the binary fusion results are given as a clue on the upper and lower bounds of the classifier fusion capacities, i.e. any fusion method is expected to be significantly better than the binary result, but hardly any fusion method will be better than the almighty result.

Moreover, the table also summarizes the results for the reduced gesture class resolution, see also Table 3.1 on page 33. Finally the label 1+2 gives the classification

Table 5.2

Classification results for pre-segmented activities – *The table summarizes the classification results for pre-segmented activities. The HMM motion based classification is fused with different location results. The fusion methods are* avg, aob, *and* pa. *The table also summarizes the results for the reduced gesture class resolution (results in brackets). Finally the label* $1+2$ *gives the classification results for taking the two most probable predictions of the classifier or classifier fusion into account. All numbers are given as a percentage of the overall amount of gesture events.*

Note that the location results are given in location class resolution whereas the motion and the fusion results are given in gesture class resolution. Thus the location recognition rates are not directly comparable with the other results.

		intra		**inter**		**external**	
loc method		corr (red)	1+2 (red)	corr (red)	1+2 (red)	corr (red)	1+2 (red)
	motion HMM	83.5 (86.8)	90.5 (91.8)	81.5 (89.7)	94.6 (96.7)	69.8 (81.0)	84.6 (87.7)
p-m	location	85.0 (85.0)	96.3 (96.3)	86.6 (86.6)	96.2 (96.2)	85.9 (85.9)	96.0 (96.0)
	almighty	98.1 (98.2)	99.9 (99.9)	98.2 (98.7)	99.8 (99.8)	95.6 (96.6)	98.9 (99.1)
	binary	70.4 (73.6)	86.9 (88.2)	69.9 (77.5)	90.9 (93.1)	60.1 (70.3)	81.7 (84.6)
	avg	90.2 (94.7)	95.9 (96.3)	87.8 (98.1)	99.2 (99.7)	77.4 (90.9)	92.3 (94.0)
	aob	90.5 (95.0)	- (-)	88.7 (99.5)	- (-)	79.4 (93.3)	- (-)
	pa	88.9 (93.1)	- (-)	86.4 (96.4)	- (-)	76.4 (89.4)	- (-)
1-m-s	location	77.8 (77.8)	92.3 (92.3)	81.4 (81.4)	96.0 (96.0)	81.1 (81.1)	94.2 (94.2)
	almighty	95.5 (96.6)	98.8 (98.8)	95.9 (97.5)	99.2 (99.7)	94.1 (96.0)	98.0 (98.4)
	binary	65.9 (67.9)	84.0 (85.3)	67.0 (73.6)	91.4 (93.0)	56.8 (66.0)	80.8 (83.5)
	avg	89.9 (94.4)	95.4 (96.0)	87.8 (98.4)	99.0 (99.7)	78.4 (91.6)	92.6 (94.5)
	aob	88.2 (92.3)	- (-)	87.7 (97.7)	- (-)	79.2 (92.1)	- (-)
	pa	88.9 (93.1)	- (-)	86.7 (96.7)	- (-)	77.0 (90.1)	- (-)
***n*-m-s**	location	80.8 (80.8)	91.8 (91.8)	87.4 (87.4)	95.6 (95.6)	85.5 (85.5)	94.2 (94.2)
	almighty	96.7 (97.3)	99.6 (99.6)	97.8 (98.4)	99.7 (99.8)	95.6 (96.2)	98.7 (98.9)
	binary	67.6 (70.3)	82.6 (84.0)	71.1 (78.7)	90.4 (92.5)	59.8 (70.3)	80.1 (83.0)
	avg	89.5 (93.9)	95.6 (95.9)	87.6 (98.4)	98.9 (99.5)	77.9 (91.2)	92.6 (94.0)
	aob	88.4 (92.8)	- (-)	87.8 (98.6)	- (-)	78.9 (92.5)	- (-)
	pa	88.5 (92.7)	- (-)	86.8 (97.0)	- (-)	76.6 (89.7)	- (-)
s-m-m	location	43.7 (59.6)	74.5 (80.2)	47.1 (61.9)	76.8 (85.1)	47.7 (65.2)	81.8 (88.0)
	almighty	89.6 (94.2)	97.6 (98.6)	88.5 (95.5)	98.7 (99.5)	81.5 (92.1)	97.6 (99.1)
	binary	37.6 (52.2)	67.4 (73.5)	40.1 (56.1)	72.6 (82.2)	36.1 (54.1)	68.8 (76.5)
	avg	82.9 (93.0)	95.5 (96.0)	83.4 (97.6)	99.4 (99.6)	75.2 (91.4)	92.7 (94.6)
	aob	76.1 (90.4)	- (-)	78.5 (94.5)	- (-)	72.6 (90.9)	- (-)
	pa	86.0 (89.7)	- (-)	85.1 (96.1)	- (-)	74.4 (86.5)	- (-)
s-m-r	location	64.2 (64.2)	88.0 (88.0)	66.6 (66.6)	90.0 (90.0)	65.9 (65.9)	89.1 (89.1)
	almighty	91.3 (93.0)	98.8 (98.9)	92.4 (96.1)	99.5 (99.5)	88.5 (91.8)	97.7 (98.6)
	binary	56.4 (58.0)	79.6 (80.9)	55.7 (60.1)	85.1 (87.2)	47.2 (55.0)	76.0 (78.3)
	avg	88.9 (93.0)	94.9 (96.0)	86.9 (97.6)	99.1 (99.5)	78.0 (91.4)	92.8 (94.4)
	aob	88.7 (92.8)	- (-)	84.6 (95.4)	- (-)	77.2 (90.6)	- (-)
	pa	86.3 (89.9)	- (-)	85.1 (95.7)	- (-)	74.4 (86.2)	- (-)

legend:

1-*m-s ... 1-component Gaussian mixture distribution supervised location modeling*
p-m ... pseudo-manual: 1-*m-s using* Euclidean *distance*
n-m-s ... n-component Gaussian mixture distribution supervised location modeling
s-m-m ... semi-supervised Gaussian mixture location modeling
s-m-r ... s-m-m the reduced *gesture class resolution*
HMM ... Hidden Markov Model
avg ... fusion method based on average *class ranking*
aob ... fusion method based on average *ranking of winner classes*
pa ... fusion method based on location plausibility *analysis*
binary ... strict fusion method: both *classifiers must agree*
almighty ... fake fusion: either one *result is correct*
red ... reduced gesture class resolution
1+2 ... a gesture is counted as correctly recognized in case it came in first or second rank

results for taking the two most probable predictions of the classifier or classifier fusion into account. All numbers are given as a percentage of the overall amount of *ground truth* gesture events.

The results given in Table 5.2 suggest that the pseudo-manual and the supervised location modeling methods outperform both semi-supervised location modeling techniques. But considering also the second ranked gesture class or the reduced gesture class resolution the benefit of supervised location-gesture assignment decreases, arguing that semi-supervised location modeling is also a viable approach.

Moreover, the table suggests that location and motion provide complementary information for the contemplated gesture recognition task. The final recognition rates are between 86% and 97% when considering the reduced gesture class resolution and the *pa* fusion strategy.

In addition the results depict that *external* modeling has significant downsides in terms of decreasing recognition rates in case the system is based on mere motion sensing. The shortcoming is almost compensated by adding the location information. Thus this result confirms the *user-independence* of the mixed motion-location approach.

5.7 Conclusion

In this chapter we demonstrated that despite all its problems ultrasonic hand tracking is a valuable addition to motion sensor based recognition of manipulative gestures. Different location modeling techniques were proposed and thereafter contrasted by experimental evaluations. Summarizing, the following conclusions can be drawn:

- ✗ The results show that the supervised approaches including the manual approach outperform the semi-supervised location modeling techniques. But considering also the second ranked gesture class or the reduced gesture class resolution the advantage of supervised location modeling decreases. Thus – depending on the contemplated application – semi-supervised location modeling may still be a viable approach.
- ✗ We have shown that location and motion provide complementary information for the contemplated gesture recognition task, with (reduced) gesture recognition rates around 90% for pre-segmented activities – around 80% for the external case.
- ✗ In addition we have shown that bad recognition rates of mere motion based sensing when performing an *external* modeling scheme are compensated by adding position sensing and location information. Thus this result confirms the *user-independence* of the proposed mixed motion-location approach.

Chapter 6

Location based spotting and recognition

This chapter presents in detail our approach on combining location and motion information for activity spotting and recognition.

The proposed approach comprises location based spotting enhanced by means of a location trajectory based spotting step. Finally a strategy for fusing intermediate, class-wise results is described. The methods are tested and evaluated on the bicycle maintenance test scenario.

Finally, the chapter gives also a comparison with the approach applied in [SOJ+06] and contrasts results achieved by both methods.

6.1 Introduction

This chapter deals with the use of the proposed hand tracking approach (see Chapter 4) to identify which parts of the machinery are being manipulated during an assembly or maintenance task in order to solve the gesture spotting problem. The advantage of this approach is that it requires only minimal instrumentation of the environment. All that is needed are at least four ultrasonic *base stations* (receivers) placed at predefined locations, two ultrasonic *transmitters* attached to both wrists of the user, two wrist-worn orientation sensors, and data on the dimensions and layout of the machinery, which today is in most cases available in electronic format, at least in industrial environments.

This chapter demonstrates how hand tracking can be used to improve the accuracy of continuous recognition of manipulative gestures. Specifically this chapter presents the following contributions

- ✗ To prove that the suggested approach can handle the *continuous* case, we apply and expand a recently presented motion trajectory based spotting and recognition approach on the position trajectories derived by means of the Kalman fusion filter described in Chapter 4.
- ✗ We present the results of an experimental validation of our method. It is based on a bicycle maintenance tasks that has been repeatedly performed by six volunteers, see also Section 3.2 on page 30. Complex NULL gestures where inserted accounting for approximately 50% of the overall number of gestures (68.2% of the total data length) in the *sequence* data set.
- ✗ We demonstrate a recognition process that is initiated with a high recall location based spotting approach and continuously increases precision by fusing additional information without decreasing the initial recall rate significantly.
- ✗ What was already shown for pre-segmented activities (see Section 5.6.2) becomes even more obvious in the continuous case: our method can handle user-dependent training as well as user-independent without a significant difference in the performance.

6.2 Basic idea

One possible approach to continuous recognition of manipulative hand gestures is to correlate arm gestures with the user's hand location – or rather wrist location. More precisely, we consider the location with respect to the object being maintained or assembled. The assumption is that the probability of a gesture resembling a certain maintenance activity to be accidentally performed at the location corresponding to this activity is very low.

Moreover position based spotting seems to be promising because the user's location or, even more, the location of the hands is a strong indicator for starts or stops of manipulative hand gestures, see also Section 1.3 on page 4.

Figure 6.1 on the next page gives an overview of our implementation of this idea. We use the hand position information derived from ultrasonic and orientation sensor based hand tracking as described in Chapter 4 to select data segments most likely containing an activity of interest. In each segment we perform motion trajectory

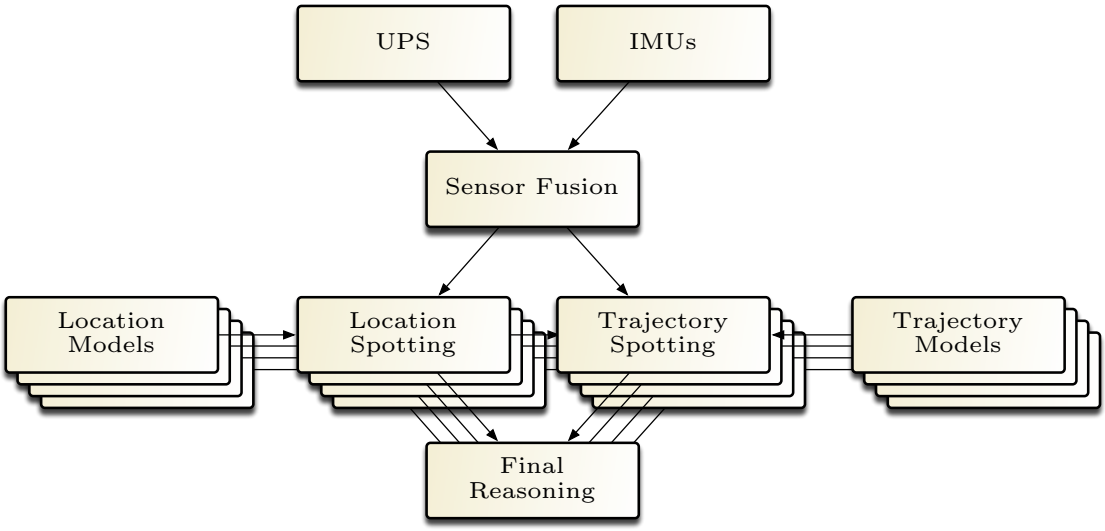

Figure 6.1
Overview of the recognition process.

based spotting already considering the gesture hypotheses derived from the location spotting step, aiming at further refinements of the spotting results. The final merge step includes additional decision making on the spotted gestures.

The locations of interest have to be defined in advance. This is done either in a manual or in a (semi-) supervised manner in the training stage, see Section 5.2 on page 68. The section also defines a distance measure for each location definition method.

6.3 Location based spotting

This spotting method is based on the detection of segments within the location stream, whose position samples lie within the location decision boundary ϑ_{dist} defined in Eq. (5.7). For bi-manual gesture classes the locations of either hand must fulfill this criterion. This step results in a class-wise binary spotting stream, see also Figure 6.2 on the next page.

The location spotting is done for each of the trained location classes of both the left and the right hand, i.e. we end up with two multi-dimensional spotting results with the dimension being the number of defined location classes per hand. Combining left and right hand locations considering the mapping of the gesture classes to the three type of gestures – *left, right, bi-manual* – still results in a multi-dimensional spotting but with the dimension being the number of gesture classes.

After this first spotting stage these detected segments potentially holding a specific gesture also have to fulfill the following criteria, aiming to delete falsely inserted segments:

- ✗ *Temporal criterion* First we define a temporal criterion, i.e. the segments must be of a certain length. For that, segments with a $length < \vartheta_{len}$ are filtered out. Preliminary – in order to handle sample-wise measurement errors – we

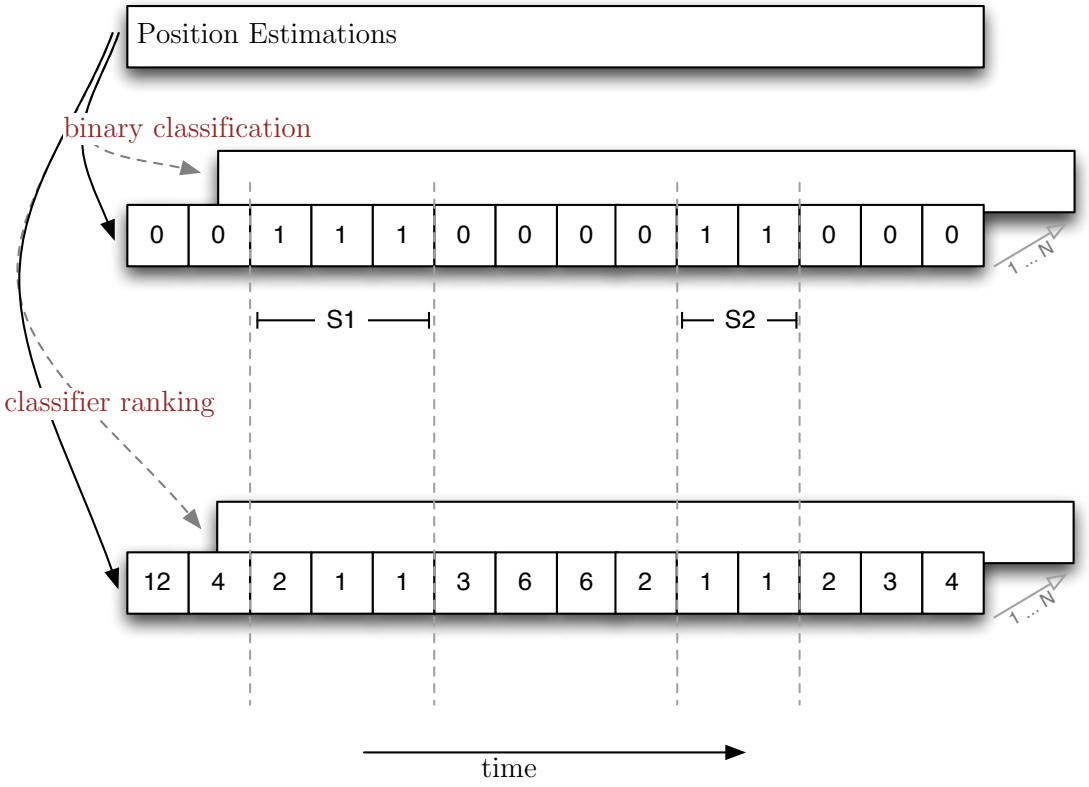

Figure 6.2
The location spotting process is tuned and controlled by criteria defined on the class-wise distance calculations.
The higher ϑ_{dist} is set, the higher the median grade of a specific segment most likely gets. Consider segment $S1$ in the illustration: in case ϑ_{dist} is increased, additional samples are most likely included within this segment. In case one sample on both sides gets included then the median grade of $S1$ would increase from 1 to 2.

filter out gaps of *length* $< \vartheta_{NULL\ len}$ between segments with identical class hypotheses.

✗ *Grade criterion* In addition we calculate the *median grade* of each segment: Each position sample $\mathbf{x}$ results in an n-dimensional vector of distances $\mathbf{d}(\mathbf{x})$, e.g. according to Eq. (5.3), with n being the number of location classes. Sorting this distance vector results in an n-dimensional vector $\mathbf{r}(\mathbf{x})$ of ordinal numbers where the i^{th} entry $r_i(\mathbf{x})$ assigns location class i a grade given position sample $\mathbf{x}$ in such a way that $r_i(\mathbf{x}) \leq r_j(\mathbf{x}) \Leftrightarrow d_i(\mathbf{x}) \leq d_j(\mathbf{x}) \forall \{(i,j) \in [1,\ldots,n]\}$. Thus $\mathbf{r}(\mathbf{x})$ is the *location class ranking* given position estimation $\mathbf{x}$. Since each detected segment consisting of position samples $[\mathbf{x}_1,\ldots,\mathbf{x}_m]$ has got a specific location class hypothesis i, we can assign each segment a series of grades $[r_i(\mathbf{x}_1),\ldots,r_i(\mathbf{x}_m)]$. We will refer to the *median* of this series as *median grade*.

In case this *median grade* is above a certain threshold ϑ_{grade} the segment is assumed to be an insertion and gets deleted. In case of a bi-manual gesture the *median grades* for both location classes must be below a certain threshold.

The choices for ϑ_{dist} and ϑ_{grade} are the tuning factors to minimize both deletion

and insertion rate of the spotting stage: The higher ϑ_{dist}, the lower the number of deletions, but the higher the number of insertions; the same is true for ϑ_{grade}. On the other hand, the higher ϑ_{dist} is set, the higher the *median grade* of a specific segment gets because the segment boundaries are spread due to the lower restrictive decision boundary. Thus we most likely include more samples with a higher grade which increases the *median grade* of this segment; for an example see Figure 6.2. By increasing ϑ_{dist} at a constant value for ϑ_{grade} we both add segments to the spotting result but also exclude some other segments. We now apply a bi-modal spotting stage, the first stage is using a high value for ϑ_{dist} and the other stage is applying a low value, both using the same value for ϑ_{grade}. The final output is the union of both spotting stages. This has turned out to be an effective method to increase the precision without significantly lowering the recall.

6.4 Trajectory based spotting

Up-sampling the position data by means of a complementary Kalman filter, see Section 4.4.5 on page 58, does not only result in position and thus location estimation with higher resolution in time, but also in a more dynamical position estimation, i.e. the position estimation is aware of subtle and fast motions of the hand. Hence, these resulting hand trajectories are better applicable for gesture recognition.

6.4.1 Introduction

A class-wise motion trajectory based spotting approach was already presented by Stiefmeier *et al.* [Sti08, SRO+08, SRT07b, SRT07a], see also Section 8.2.3 on page 116. It is based on a discretization of the motion trajectories of the lower arms or the hands resulting in a motion trajectory alphabet. Furthermore, for each gesture class an *optimal* string, which best represents the training set, is trained. For methods how to train these optimal string patterns refer to [Sti08].

In the spotting stage a string matching function is used to retrieve similar patterns within the discretized motion trajectories. This string matching function results in matching cost series with minima at the end of potential gesture segments. Cost function results for trajectories of different body parts are finally fused to handle left, right, and bi-manual gestures and to handle gestures that are more related to the arms or more related to the hands.

The approach has been tested on trajectories recorded using the *Motion Jacket* [SRO+08]. The Motion Jacket uses four or six orientation sensors on the upper body limbs (upper arms and lower arms or upper arms, lower arms, and hands) to track their motion trajectory. All trajectories are referenced to the motion trajectory of an additional orientation sensor fixed on the torso – either on the chest or on the back – of the user. By compensating the heading of the user by means of the measured heading of the torso orientation sensor, the trajectories are calculated according to a reference system *translating but non-rotating* compared to the global reference system; i.e. the resulting trajectories are given in a user reference system with the x-axis[1] rotated to magnetic north; thus the user always seems to

[1] The x-axis is defined in such a way that it is always pointing into the same direction the user is facing.

face magnetic north. In such a way the trajectories are invariant to global user orientation and global user position. For a visual comparison of the two trajectory estimation approaches (mere orientation sensing and mixed location and orientation sensing) see Figure 6.3 on the opposite page.

6.4.2 Approach

This work aims at applying this technique to the trajectories that can be derived from the position estimation of the fusion of the absolute position measurements with the relative position estimation of the inertial measurement units described in Section 4.4.5 on page 58.

This mixed location-orientation sensor approach should outperform the mere orientation approach in the following points:

- ✗ The orientation sensors used in [SRO+08] are inertial measurement units mixed with a three-dimensional magnetic field sensor as an absolute reference source. This is a decent approach as long as the magnetic field is non-changing over time and homogeneous, otherwise this results in drifting orientations. Evidently, ultrasonic positioning as an aiding source has got its disadvantages as well, in industrial environments in particular. However, magnetic aiding will most likely cause slightly drifting errors also confusing the estimations based on the relative measurements whereas location aiding sources will mainly cause missing position readings, which can be for a short term compensated by the relative measurement system.
- ✗ The mere orientation approach uses at least three on-body sensors to measure the orientation of the wrist. Each sensor has to be fixed separately on a single body limb. In case both wrist trajectories are of interest the sensor setup demands five sensors fixed on different locations on the user. In case a position sensor is used, there is need for two motion sensors and two position sensors for both wrists. A final implementation could even integrate motion and position sensors to one sensor unit.

As stated above, the mere motion trajectory approach results in rotation and position invariant trajectories. An additional orientation sensor would be necessary to achieve the same in the mixed position and orientation approach.

We now apply this trajectory based spotting approach to further refine the location based spotting result. More precisely, we apply this method only on preliminary spotted segments. Every segment is assigned a specific location hypothesis and one or more gesture hypotheses. The string trajectory patterns trained for each of these expected gestures are then used to search in these spotted segments in order to recognize matching trajectories and thus to locate sub-segments containing the actual gesture. This results in a (*multivariate*, in case of multiple gesture hypotheses) series of matching costs for every single location segment.

6.4.3 Matching cost minima spotting

As stated above the matching costs have a minimum at the end of a potential gesture segment. Thus [SRO+08] suggests a class-wise trained threshold. Subsequently, any

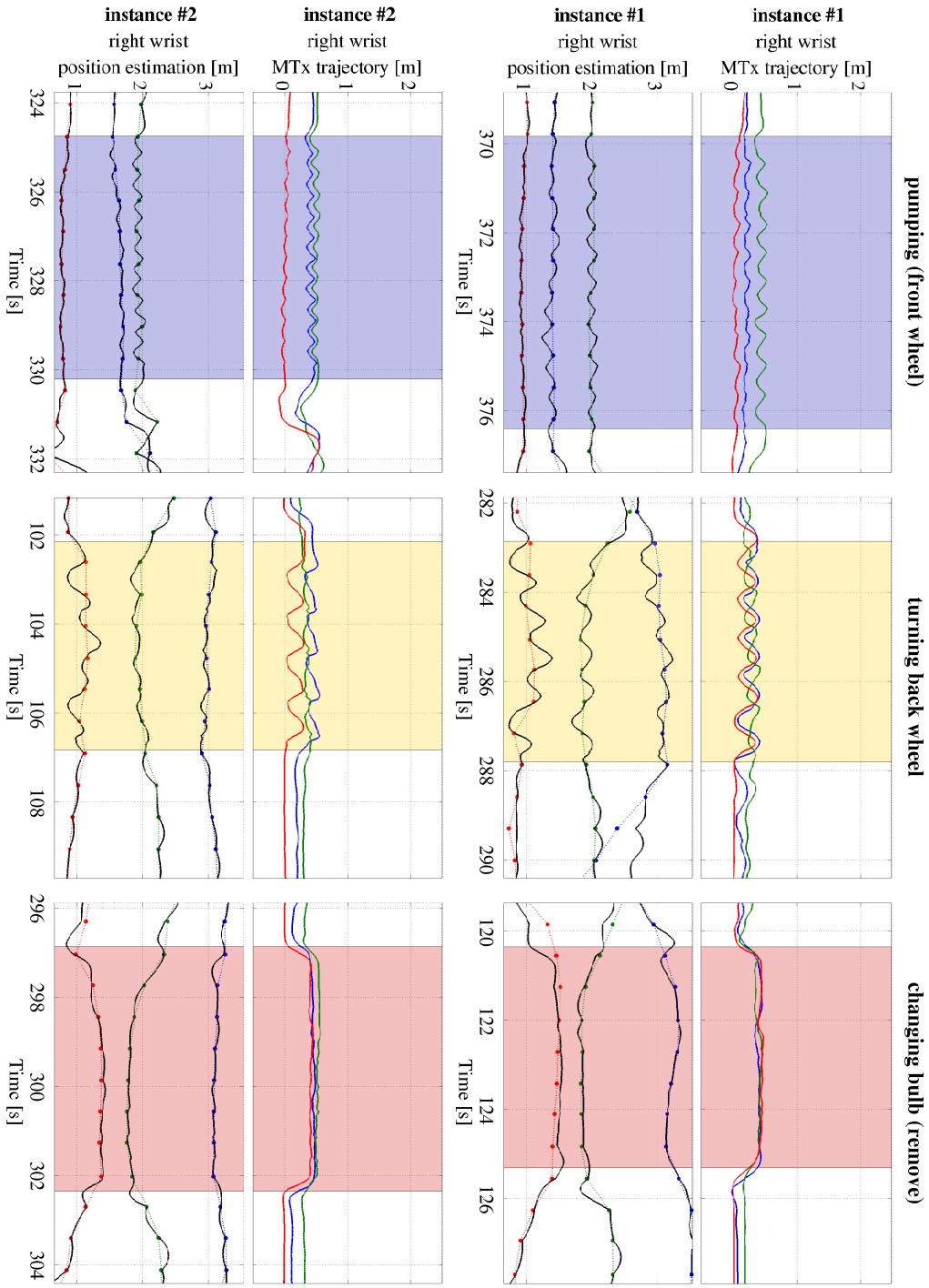

Figure 6.3

The figure depicts the results of two different sensing approaches to obtain a wrist trajectory. The MTx based trajectories display those trajectories calculated from three on-body orientation sensors: chest, upper and lower arm. The resulting trajectories are given in the user reference system with the x-axis rotated to the magnetic north. The position estimation depicts the coordinates of the Kalman filter based fusion of motion and position readings, see also Figure 4.3 on page 64. The resulting trajectories of the second approach are given in the global reference system.

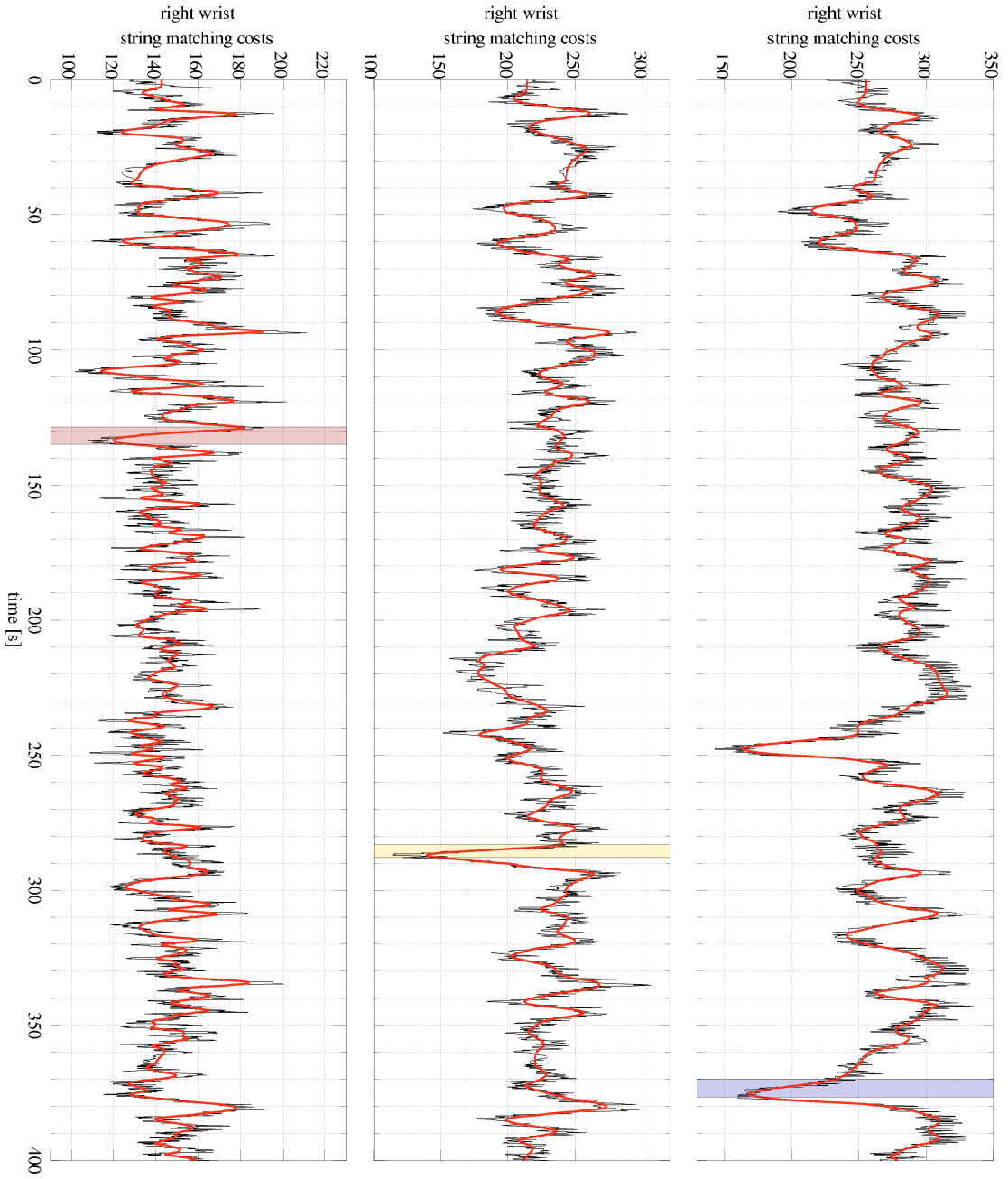

Figure 6.4

Matching cost time series – *the plots depict the matching cost time series for three different trained string patterns. All three matching cost time series depict the result for the same data set of the type* sequence. *The colorized regions depict the ground truth annotation for the respective manipulative hand gesture. Black depicts the raw matching costs and red the filtered matching cost series after applying a low-pass filter to the raw matching costs. The low-pass filter uses a cosine-tapered window of the size of the pattern that was trained for the respective gesture class.*

decreasing matching cost segment with an endpoint below this threshold is assumed to contain the respective gesture.

Figure 6.4 on the next page exemplifies matching cost series as calculated for three different gestures on the same sequence. The colorized regions mark the ground truth annotations of the respective gesture. As expected, the matching costs reach a minimum in all three examples. In case of gesture *pumping (front wheel)* this minimum is outstanding compared to the other minima peaks, the other outstanding minimum results from an instance of gesture *pumping (back wheel)*. A contrary result can be seen in the third plot. Though the matching cost series for gesture *changing bulb* does have a minimum at the end of the annotated gesture, this minimum is not outstanding at all. Thus a strategy based on mere minima thresholding does not seem to be viable for all gesture classes. In fact it turned out to be applicable for approximately the half of the gesture class set of the bicycle maintenance case study. For the other gestures this approach has though a decent recall but is not restrictive enough, i.e. these patterns are not specific enough to further improve the location based spotting result.

6.4.4 Multivariate analysis of matching costs

Nevertheless Figure 6.4 implies that a more accurate analysis of the matching cost series can enhance the performance of the string matching approach further. As a straightforward starting approach additional polynomial features[2] of the matching cost series are evaluated; for examples see Figures 6.5 on the following page to 6.8 on page 89. Figures 6.5 and 6.7 exemplify parameter results for data of two activities used for training and Figures 6.6 and 6.8 for the respective activities but calculated on data used for testing. The five features are *minimum*, *maximum*, *mean*, *gradient*, and *curvature*. The figures depict feature values for two different gesture classes, one most likely resulting in a good class separation (Figures 6.5 and 6.6). The others exemplify a really challenging gesture class (Figures 6.7 and 6.8).

These additional features are included into the spotting process by means of gesture-class-wise LDA classifiers. Each classifier is trained by applying the respective trajectory string pattern on training data sets of all gesture classes. The resulting matching cost series is segmented by means of minima and maxima recognition. Finally this results in two sets of polynomial features for each gesture class, one set comprising correctly spotted gestures, the other comprising the NULL class examples, see also Figures 6.5 to 6.8.

The LDA performs a dimensionality reduction for a given n-dimensional feature space resulting in a $(c-1)$-dimensional feature space with c being the number of classes. Hence in our case the new feature space is one-dimensional and a single scalar threshold can be trained for each gesture.

Note that LDA based classification has got its limitations in this case. The classifiers are trained to separate two classes: one specific gesture class and the NULL class consisting of any imaginable human hand gesture. Modeling the NULL class by means of just using any other gesture of interest is not sufficient. The resulting projection matrix and the appropriate threshold will only discriminate

[2] Refer to [EGK91] for a fast polynomial least-squares approximation. Applying such a method on the cost function remains future work.

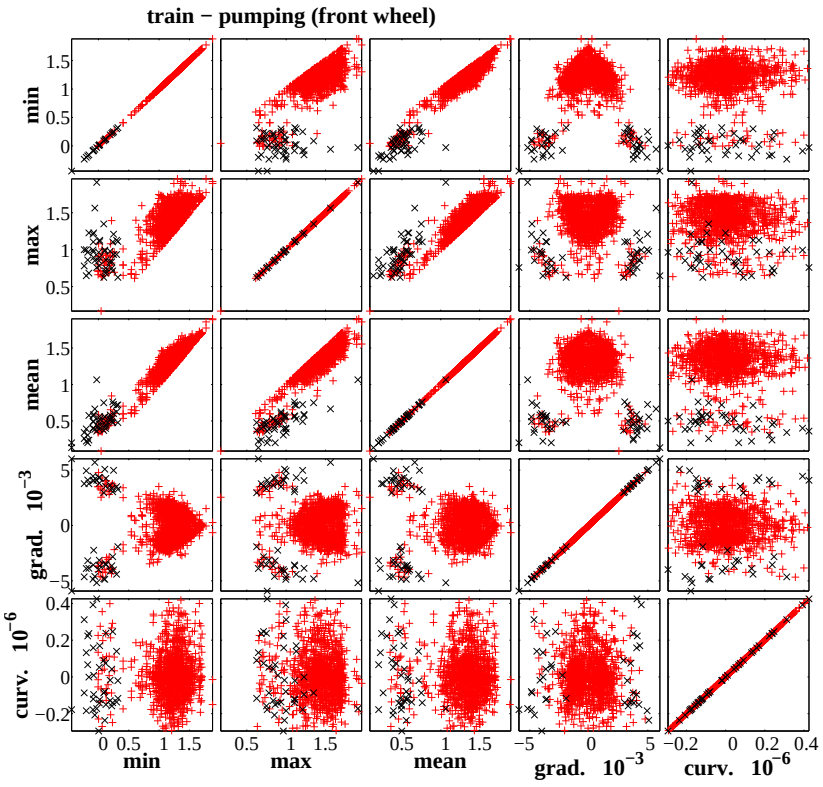

Figure 6.5 *Polynomial parameters calculated on matching cost segments found within the overall set of recordings used for training. The applied trajectory string pattern was trained for class* pumping at front wheel. *A black* **x** *reflects a polynomial feature calculated on a segment being part of an instance of the correct gesture class, a red + reflects a feature calculated from any other segment.*

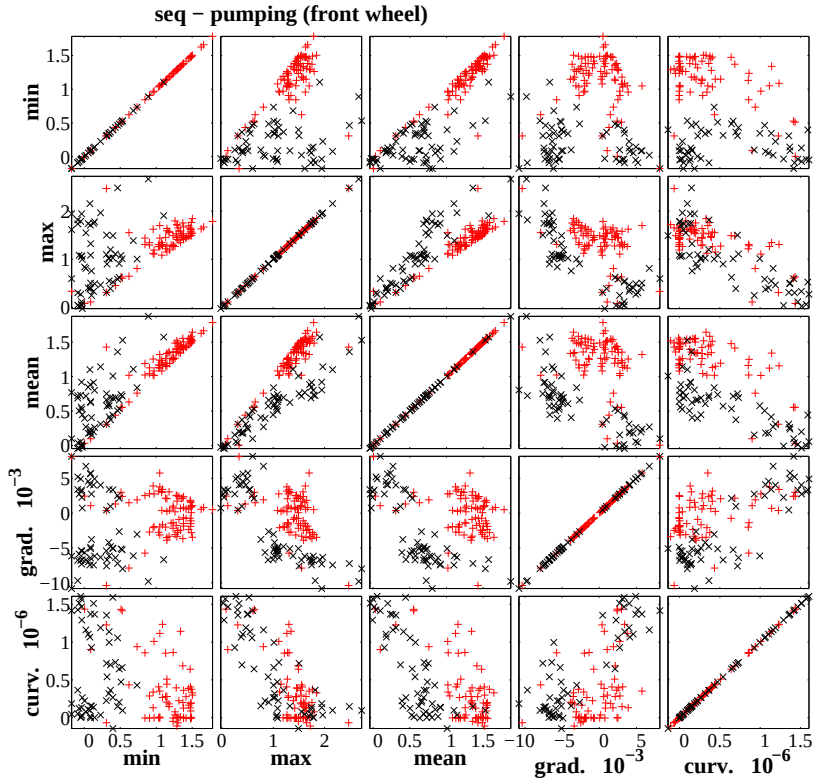

Figure 6.6 *Polynomial parameters calculated on matching cost segments found within the sequence data after the location spotting process using trajectory string pattern trained for class* pumping at front wheel. *A black* **x** *reflects a polynomial feature calculated on a segment being part of an instance of the correct gesture class, a red + reflects a feature calculated from any other segment.*

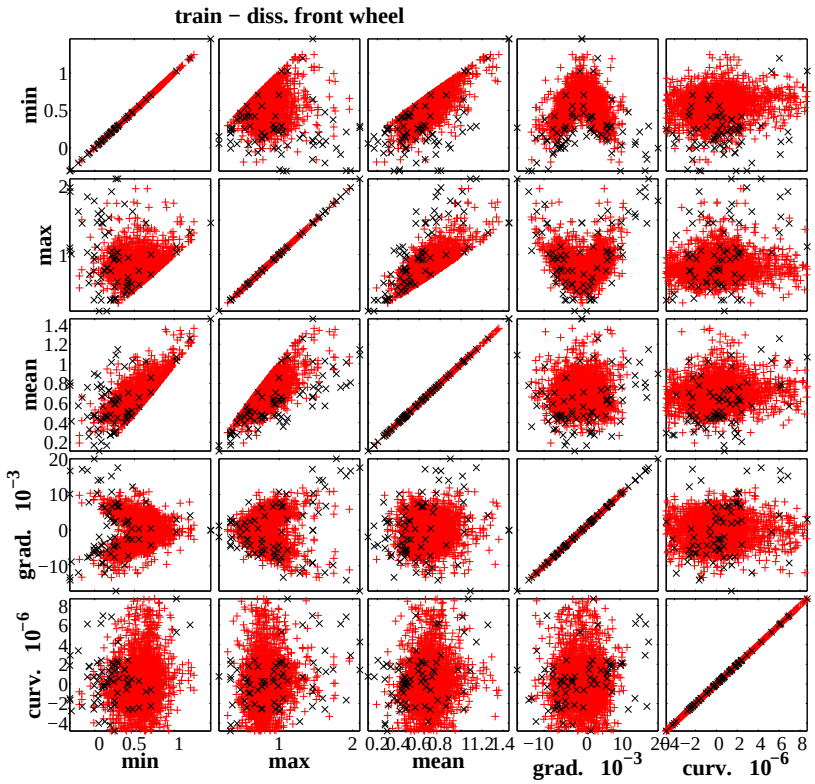

Figure 6.7 *Polynomial parameters calculated on matching cost segments found within the overall set of recordings used for training. The applied trajectory string pattern was trained for class* disassembling front wheel. *A black* **x** *reflects a polynomial feature calculated on a segment being part of an instance of the correct gesture class, a red + reflects a feature calculated from any other segment.*

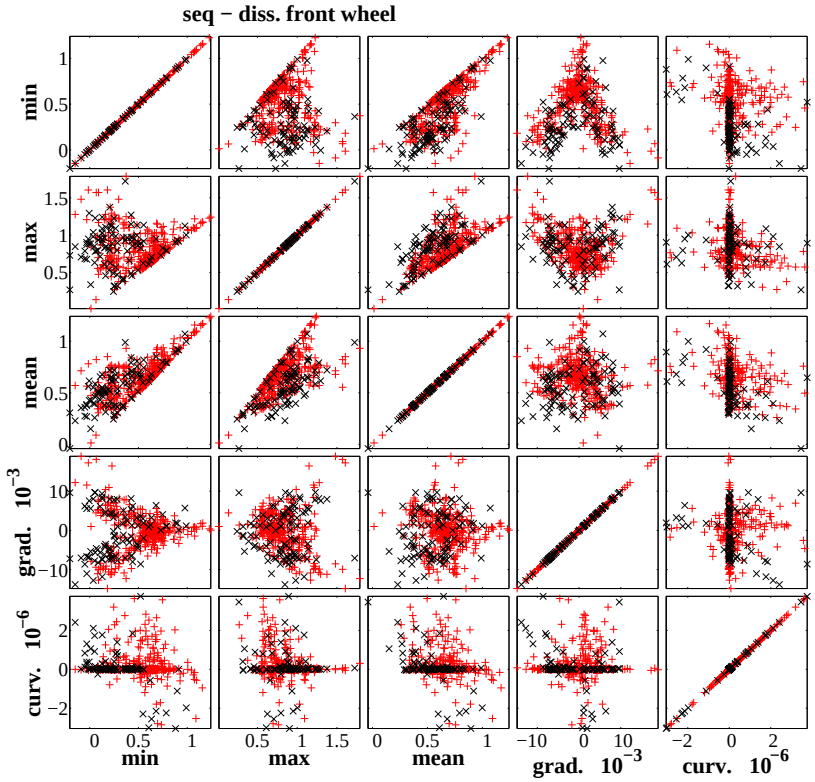

Figure 6.8 *Polynomial parameters calculated on matching cost segments found within the sequence data after the location spotting process using trajectory string pattern trained for class* disassembling front wheel. *A black* **x** *reflects a polynomial feature calculated on a segment being part of an instance of the correct gesture class, a red + reflects a feature calculated from any other segment.*

between the gesture and the set of all other trained gestures where it should obviously discriminate between the gesture and *any* other possible gesture.

6.5 Final decision making process

Trajectory based spotting result in a set of segments, that has to be further processed for the purpose of

- ✗ excluding falsely predicted gesture events (insertions) and thus increase the final precision;
- ✗ deciding on gesture class hypotheses that have temporal concurrencies. These temporal concurrencies result from the class-wise processing approach, i.e. all preceding processing stages operate separately in a class-wise manner unaware of the concurrent recognition results.

In order to cope with these two issues, we apply a *plausibility analysis* stage and a simple strategy aiming to resolve recognition concurrencies.

6.5.1 Plausibility analysis

The plausibility analysis stage argues that after the trajectory based spotting step more restrictive location criteria than in the mere location spotting stage can be applied.

This is because the result of the trajectory based spotting stage further confines each segment from location class resolution to gesture class resolution. Thus in case of a correctly spotted gesture the position samples of this *gesture segment* should on average fit the location model better than the overall location segment's samples do: the trajectory based spotting stage excludes samples at the beginning and the end of the location segment that might be *close* to the location but are not part of the actual activity, whereas position samples remaining in the gesture segment should be even closer to the respective location model than the excluded position samples are. In case of an incorrectly spotted gesture this is obviously not true.

The criteria – calculated on the gesture segments – on which this analysis is based on are:

- ✗ the fraction of position samples with grade $r_{l(i)} = 1$,
- ✗ the fraction of position samples with grade $r_{l(i)} \leq 2$,
- ✗ the minimum distance value to location $l(i)$, and
- ✗ the median distance value to location $l(i)$

with i being the gesture class hypothesis, $l(i)$ being the location class of gesture i, and r being the ranking vector as defined in Section 6.3. The above listed criteria are used to further exclude potentially falsely recognized gestures.

Additional criteria based on the trajectory matching cost function would make a class-comprehensive normalization of the matching costs necessary. This has remained an open issue so far.

6.5.2 Concurrency resolving

Finally the remaining gesture recognition concurrencies have to be resolved. For this purpose we calculate a confidence value derived from the above listed criteria, according to the product of all four criteria – all eight in case of a *bi-manual* gesture. In case of a concurrency (i.e. *a list of segments overlapping one another*), we search for all possible sub-lists containing no concurrency and decide in favor of the sub-list that contains the gesture segment with the highest confidence value, as already proposed in Section 2.3.2 on page 25. Evidently, this strategy will fail and cause a considerable amount of deletions in case of heavily nested concurrencies; which is not the case for the predicted gesture event streams after the plausibility analysis stage at least in the presented case study.

Before this strategy is applied, some *minor* concurrencies are pseudo-resolved, i.e. in case of a concurrency of two segments with the overlapping fraction of both segments being smaller than a certain threshold – this threshold is set to 30% for the presented evaluations – we shorten these segments by half of the respective overlapping fraction. For any given segment repeated pseudo-resolving is allowed while the deleted fraction on each side of the segment is equal or smaller than half of the threshold.

6.6 Summary of the previously published approach

Section 6.7 will also compare the current results with the results achieved in [SOJ+06]. Thus this section summarizes this preliminary applied approach. Location based spotting is based on position data derived from least squares optimization. In the training stage gestures are manually grouped into a set of eleven locations, similar to Table 5.1 on page 75. For both hands, mean and variances are modeled for these locations according to the training data.

In the gesture spotting stage, the Mahalanobis distance is used to estimate the probabilities for each sample to be part of a specific location, analogous to method 1-m-s, resulting in a parallel spotting stream for each location of interest.

In a next step a Mahalanobis classification similar to the location spotting stage itself is done, despite its being trained for all gesture classes instead of the location classes. For each spotted location segment each sample out of this segment is classified using the Mahalanobis distance. A majority vote over all samples assigns then a final gesture class hypothesis.

For motion based classification HMMs have been chosen, similar to Section 5.5.1. The features for the HMMs are raw inertial sensor data, i.e. acceleration and rate of turn, on the one hand. On the other hand, we derived orientation information from the set of inertial sensors in form of Euler angles to complement the raw sensor data features. The employed set of features comprises the following subset of available sensor signals and derived quantities: two acceleration and one gyroscope signal from the right hand, pitch angles from right lower and upper arm, two acceleration signals from the left hand and the pitch angle of the left upper arm.

Finally fusion algorithms as described in Section 5.5.3 are used to combine both classifier results and to exclude possibly falsely spotted gestures. The fusion stage constitutes the major difference to the current modular approach conceptually

explained in Chapter 2. Any decision on rejecting or accepting a specific gesture hypothesis is always made in comparison to another gesture hypothesis or even a set of hypotheses. Hence these stages have to be trained on the overall set of gestures of interest and what is more the classes cannot be be optimized individually. In both approaches an activity class with low precision rate most likely decreases the recall rate of other classes. But in case the recognition is optimized in a class-wise manner this negative inter-class effect is kept to a minimum.

6.7 Experimental results

6.7.1 Introduction

The presented evaluations test the suggested hand tracking (see Chapter 4) and location modeling approaches (see Chapter 5) on its ability to spot sporadic hand gestures by means of applying the approaches presented within this chapter. Thus this section presents results of the **continuous** case.

Once again we use the experiment described in Section 3.2 on page 30. Additional information and a description of the applied evaluations schemes (*intra*, *inter*, and *external*) was already given in Section 5.6 on page 75.

Ultrasonic distance readings of both wrist-worn ultrasonic transmitters and orientation, acceleration, and gyroscope readings of both MT9B devices worn on the lower arms (see also Figure 3.1 on page 32) are processed as described in Chapter 4 using the filter structure as depicted in Figure 4.2 on page 61 resulting in two wrist trajectories as exemplified in Figure 4.3 on page 64.

6.7.2 Spotting results

Location based spotting and location trajectory based spotting results for different location methods and the three different evaluation schemes are given in Table 6.1 on the next page. In order to count event errors the evaluation metric proposed by [WLT06, War06] was applied, see also Appendix B on page 139.

Note that all errors have to be counted *gesture*-class-wise, thus merge errors are impossible because no recorded sequence contains more than one instance of a certain gesture class. On the other hand, insertion errors are overestimated, e.g. in case the location spotting stage correctly detects location class *screw A* because the subject is *tightening screw A*, this will account for a correct event for this gesture class but also for an insertion error for the respective other *screw A* class.

The results of the spotting evaluations can be summarized as follows.

- ✗ Table 6.1 on the next page shows that the semi-supervised location modeling methods are outperformed in terms of correctly recognized location event rates and substitution error rates by the supervised and the pseudo-manual methods.
- ✗ Moreover, the results point out that the mere *intra* location modeling seems to provide too little diverse location training data. Whereas for motion based gesture recognition *intra* training schemes usually outperform *inter-subject* trained models, this does not seem to be valid for location features. Thus –

Table 6.1

Location and location trajectory spotting results for different location methods and the three different evaluation schemes. The results are given as a percentage of the overall amount of gesture events. The results for overfill and underfill errors are given as a percentage of the overall amount of correctly predicted events.

eval. scheme	spotting stage	correct	del	subst	ins	frag	mer	over	under
intra	loc (p-m)	93.8	2.3	3.9	269.9	0.1	-	98.9	42.3
	traj (poly)	90.8	5.7	3.0	89.7	0.5	-	97.7	47.9
	traj (min)	*89.0*	*4.7*	*4.7*	*133.9*	*1.6*	-	*82.6*	*64.6*
	loc (1-m-s)	97.3	1.4	1.2	286.4	0.1	-	98.7	38.9
	traj (poly)	94.2	3.0	2.4	110.1	0.5	-	97.2	45.5
	traj (min)	*93.4*	*2.7*	*2.1*	*161.6*	*1.7*	-	*83.4*	*61.8*
	loc (*n*-m-s)	90.7	4.2	4.8	206.1	0.3	-	97.9	49.6
	traj (poly)	87.8	8.4	3.3	75.9	0.6	-	96.1	54.3
	traj (min)	*86.2*	*7.8*	*4.4*	*107.1*	*1.6*	-	*79.8*	*68.5*
	loc (s-m-r)	84.9	1.4	13.6	278.4	0.1	-	96.4	47.6
	traj (poly)	82.3	10.6	6.9	127.7	0.2	-	94.6	53.2
	traj (min)	*80.8*	*9.4*	*8.1*	*138.0*	*1.8*	-	*79.3*	*69.5*
	loc (s-m-m)	79.6	0.1	20.0	262.4	0.2	-	96.6	50.6
	traj (poly)	77.6	8.4	13.8	114.5	0.2	-	95.2	54.7
	traj (min)	*75.9*	*6.8*	*15.8*	*125.5*	*1.5*	-	*78.8*	*67.6*
inter	loc (p-m)	96.8	1.1	2.1	315.5	0.1	-	99.2	36.1
	traj (poly)	93.8	3.1	2.7	112.3	0.4	-	97.8	42.3
	traj (min)	*92.2*	*3.0*	*3.1*	*154.7*	*1.7*	-	*85.7*	*61.9*
	loc (1-m-s)	97.4	1.1	1.5	313.1	0.1	-	99.2	33.2
	traj (poly)	94.5	2.5	2.5	129.6	0.5	-	98.1	41.5
	traj (min)	*93.5*	*2.1*	*2.5*	*176.0*	*1.8*	-	*85.9*	*59.7*
	loc (*n*-m-s)	95.2	1.5	2.6	264.5	0.7	-	99.5	42.7
	traj (poly)	92.0	3.8	3.2	106.8	1.0	-	97.2	49.3
	traj (min)	*90.9*	*3.3*	*3.7*	*144.9*	*2.1*	-	*84.3*	*65.5*
	loc (s-m-r)	81.3	1.6	17.0	200.2	0.1	-	97.5	49.8
	traj (poly)	79.6	9.4	10.7	88.2	0.2	-	95.9	53.8
	traj (min)	*77.4*	*7.1*	*14.0*	*95.1*	*1.5*	-	*77.7*	*68.9*
	loc (s-m-m)	80.1	1.0	18.8	223.7	0.1	-	97.2	49.1
	traj (poly)	78.4	12.1	9.3	104.0	0.2	-	95.2	53.4
	traj (min)	*76.5*	*8.6*	*13.4*	*116.8*	*1.5*	-	*78.1*	*70.4*
external	loc (p-m)	96.1	1.2	2.6	319.0	0.1	-	99.2	37.5
	traj (poly)	93.2	3.9	2.6	111.5	0.3	-	97.8	43.4
	traj (min)	*91.5*	*3.5*	*3.4*	*158.4*	*1.6*	-	*85.4*	*63.3*
	loc (1-m-s)	97.4	1.2	1.3	315.5	0.2	-	99.2	35.0
	traj (poly)	94.3	2.8	2.4	124.8	0.5	-	97.9	42.2
	traj (min)	*93.4*	*2.3*	*2.5*	*174.0*	*1.8*	-	*85.6*	*60.8*
	loc (*n*-m-s)	94.9	1.9	2.7	263.3	0.5	-	99.3	44.9
	traj (poly)	91.9	4.4	3.0	104.6	0.7	-	97.2	50.7
	traj (min)	*90.4*	*3.9*	*3.9*	*142.4*	*1.8*	-	*83.5*	*66.9*
	loc (s-m-r)	86.9	0.3	12.7	265.6	0.1	-	97.2	45.4
	traj (poly)	84.6	5.7	9.2	114.2	0.5	-	96.0	50.7
	traj (min)	*82.7*	*3.8*	*12.0*	*109.9*	*1.5*	-	*80.4*	*68.6*
	loc (s-m-m)	81.3	1.3	17.3	221.1	0.0	-	97.7	46.4
	traj (poly)	79.3	9.0	11.5	113.1	0.2	-	96.9	51.0
	traj (min)	*77.8*	*6.2*	*14.5*	*110.9*	*1.6*	-	*79.9*	*68.0*

legend:

del ... deletion error
subst ... substitution error
ins ... insertion error
frag ... fragmentation error
mer ... merge error
over ... overfilled event
under ... underfilled event
loc ... location based spotting stage
traj ... location trajectory based spotting stage either using minimum matching cost search (min) or using the LDA classifier trained on additional polynomial features (poly)

1-m-s ... 1-component Gaussian mixture supervised location modeling
p-m ... pseudo-manual: 1-m-s using Eucl. *distance*
n-m-s ... n-component Gauss. mixture supervised location modeling
s-m-m ... semi-supervised Gaussian mixture location modeling
s-m-r ... s-m-m using reduced *gesture class resolution*

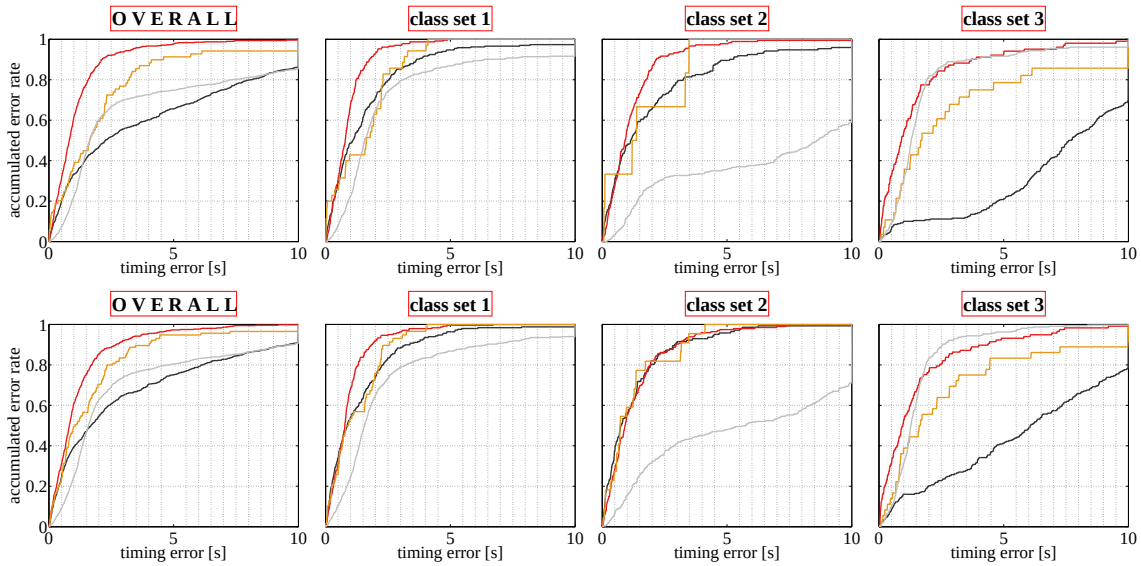

Figure 6.9

Accumulated timing error rates – The plots in the upper row depict the results for the location spotting stage whereas the second row depicts the timing errors of the location trajectory based spotting stage. Class set 2 comprises all opening classes and class set 3 summarizes all closing classes. Class set 1 comprises all other gesture classes. Four types of timing errors are depicted: pre-overfill (dark gray), post-overfill (light gray), pre-underfill (red), post-underfill (orange). Errors are given in absolute time.
See also Figures 6.11 and 6.10.

as expected – location appears to be truly user-independent and moreover locations can and should be modeled by means of a set of recordings providing a decent diversity.

- ✗ The event results in Table 6.1 also include the results of the location trajectory based spotting stage using mere minima search in the string matching cost function. The results clearly show that the use of additional polynomial features to evaluate the string matching costs enhances the spotting performance regarding its insertion rate, arguing that the mere minima search is too little selective although it results in decent recall rates.
- ✗ In Table 6.1 an overfill error accounts for a correct prediction event that exceeds its appropriate ground truth event whereas an underfill error accounts for the opposite, see also Appendix B on page 139. Evidently, a correct prediction event can also account for both types of errors. Figures 6.9 to 6.10 give additional information on these timing errors for the spotting results. The figures show the timing errors for external evaluation scheme and n-m-s location modeling. The plots depict the accumulated timing errors for these types of errors: *pre-overfill*, *post-overfill*, *pre-underfill*, and *post-underfill* (as defined in Appendix B on page 139). Note that the timing errors in Table 6.1 are counted event-wise and given in fraction of the overall amount of correctly predicted events, whereas the error rate in Figures 6.9 to 6.10 is given either in fraction of the length of the appropriate ground-truth element (fgel) or of the appropriate prediction event (fpel) or in absolute time and accumulated over

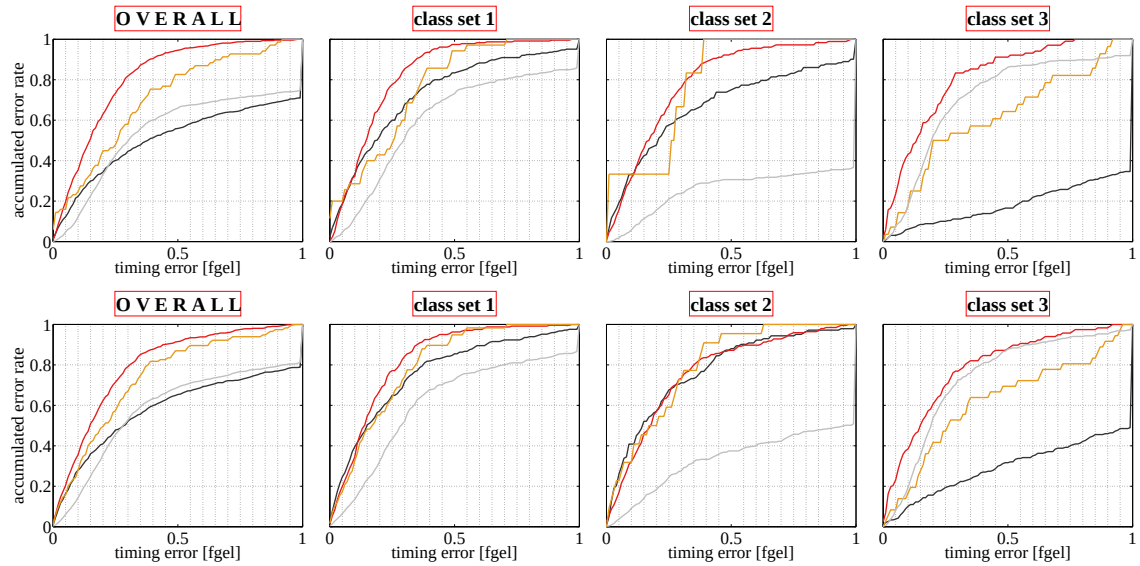

Figure 6.10
Accumulated timing error rates – see also Figure 6.9. Errors are given in fraction of the length of the appropriate ground-truth event (fgel).

all timing error events. Thus the accumulated error rate displays the fraction of timing error events that are smaller or equal a certain time span, fgel or fpel.

Whereas the event-based evaluation of the timing errors suggests that the location trajectory based spotting stage did not achieve an improvement, the accumulated timing errors prove the spotting optimization: the overfill errors decrease whereas the underfill errors do not significantly increase.

- The plots also depict that merge errors are actually present in the spotting result – although not shown in Table 6.1. Though the gesture sequence order was randomized during recording of the sequence data sets, specific pairs of gestures, namely *open/close* gesture pairs, were always accomplished consecutively. Thus both the *post-overfill* timing errors for *opening*-gestures and the *pre-overfill* timing errors for *closing*-gestures are far above the amount of the respective other three error types. This result suggests that similar gestures performed at the same location can hardly be distinguished and separated by the suggested approach.

6.7.3 Continuous recognition results

The results for the continuous recognition case, i.e. the final results, are given as *precision and recall* plots. Transferring the precision and recall definition as used in the *information retrieval* domain to our recognition problem results in the following definitions:

- *precision* is the ratio of correctly predicted events and overall number of predicted events whereas
- *recall* is the ratio of correctly predicted events and overall number of ground truth events.

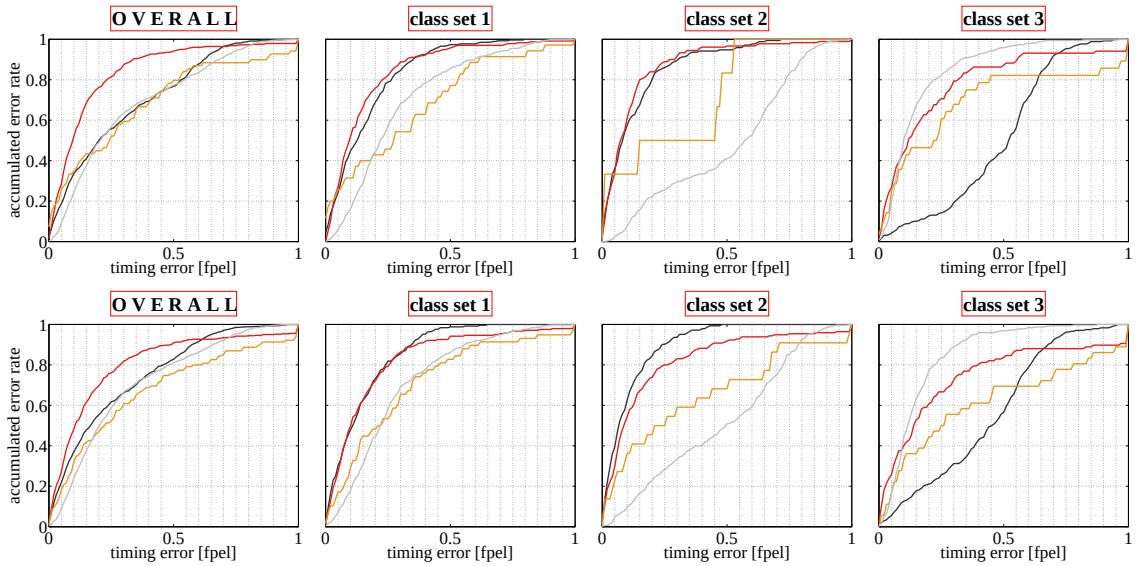

Figure 6.11
Accumulated timing error rates – see also Figure 6.9. Errors are given in fraction of the length of the appropriate prediction event (fpel).

Once again, the event error count directive defined by [WLT06, War06] was used to achieve these error rates. For an exact definition of how precision and recall are defined in our case see Appendix B on page 139.

Figures 6.13 on page 99, Figure 6.12, and Table 6.2 on the opposite page summarize parts of the results of both the final and intermediate processing stages.

As depicted in these figures, the proposed spotting and recognition procedure is initiated by means of a high recall location based spotting stage. Any additional refinement stage succeeds in increasing the precision without decreasing the recall significantly. The final concurrency resolving stage does not introduce further deletions at a significant rate. For open/close gesture pairs – these are gesture pairs with only a small amount of diversity in their motion pattern performed at exactly the same location – the reduced class resolution evaluation scheme suggests that insertion and deletion errors are mainly due to substitutions between these gesture pairs.

In addition Table 6.2 and Figure 6.12 compare the results achieved by the current approach with the method used in [SOJ+06] as described in Section 6.6 on page 91. Evidently, the current approach outperforms the previous approach in terms of a better recall and a far better precision.

6.8 Conclusion

We demonstrated that ultrasonic hand tracking might help solving the gesture spotting problem in the contemplated industrial production and maintenance scenario. Summarizing, the following conclusions can be drawn:

- ✗ As already shown for pre-segmented activities, the location context of the user's hand proved to be user independent. Moreover mere *intra* location modeling seems to provide too little diverse location training data. Whereas

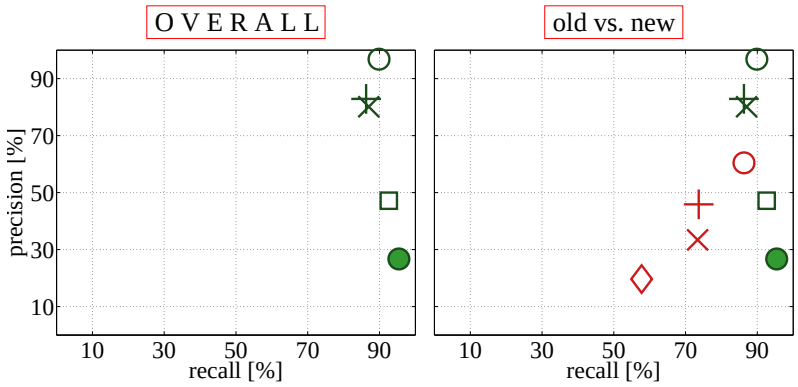

Figure 6.12
Overall precision and recall, see also Figure 6.13. The depicted processing stages are: location based spotting (green disk), location trajectory based spotting using polynomial matching cost features (square), plausibility analysis (x), prediction concurrency dissolving (+), *and* prediction concurrency dissolving evaluated using the reduced class resolution (white disk).
The plot to the right also compares current results with the results achieved in [SOJ+06], see also Table 6.2. The depicted processing stages are: motion (red diamond), location (red x), final (red +), *and* final using the reduced class resolution (red circle).

Table 6.2
Overall precision and recall – results for different spotting stages and different location modeling methods using the external *evaluation scheme. The table to the right summarizes the results achieved in [SOJ+06] using the same evaluation scheme.*

	p-m		1-m-s		*n*-m-s	
	rec	**prec**	**rec**	**prec**	**rec**	**prec**
loc	97.5	23.6	96.2	23.2	95.4	26.7
traj (poly)	94.8	43.3	93.5	45.7	92.6	47.2
pa	89.9	77.5	88.4	79.9	87.0	80.1
diss	88.3	80.9	86.4	82.6	86.3	82.9
diss (red)	91.7	95.4	89.9	96.1	89.9	96.8

results from [SOJ+06]		
	rec	**prec**
motion	57.8	19.7
location	73.4	33.4
final	73.7	45.9
final (red)	86.3	60.4

legend:
rec ... recall
prec ... precision
loc ... location based spotting stage
traj ... location trajectory based spotting stage using the LDA classifier trained on polynomial matching cost features
pa ... post spotting plausibility analysis
diss ... prediction concurrency dissolving stage
diss (red) ... prediction concurrency dissolving stage, evaluated using the reduced class resolution

for motion based gesture recognition *intra* training schemes usually outperform *inter-subject* trained models, this does not seem to be valid for location features. Thus hand locations can and should be modeled by means of a set of recordings providing a decent diversity.

- ✗ Location and location trajectory spotting seem to be promising approaches, resulting in a correct rate above 95% for the location spotting stage and a correct rate above 90% for location trajectory based spotting refinement stage when applying inter-user, supervised location modeling.
- ✗ We have suggested a modification (LDA classification of polynomial matching cost features) for the motion trajectory based spotting approach presented by [Sti08, SRO+08, SRT07b, SRT07a]. This adaptation allows to apply this approach on all tested classes without the high insertion error rate of the original version of this approach.
- ✗ By means of consecutive motion and location information fusion we finally increase the precision of the continuous recognition process above 80% (above 95% for the reduced case) whereas the recall does not go under 86% (almost 90% for the reduced case).
- ✗ The results have shown that the substitutions are mainly due to do/undo gesture pairs, arguing that the proposed spotting and recognition approach still has great problems with similar gestures performed at the same location.

Figure 6.13

Precision and recall – results of the n-m-s *location method for different processing steps in the spotting and recognition procedure using the* external *evaluation scheme. The depicted processing stages are:* location based spotting (green disk), location trajectory based spotting using polynomial matching cost features (square), plausibility analysis (x), prediction concurrency dissolving (+), *and* prediction concurrency dissolving evaluated using the reduced class resolution (white disk).

Chapter 7

Muscle activity monitoring*

This chapter investigates the usefulness of muscular information of the lower arms. To this end an adequate sensing hardware comprising several FSRs is developed and implemented.

We then systematically investigate the usability of the FSR system to recognize different manipulative gestures. The aim is to test the limits of the system, compare them to established sensing modalities, i.e. three-dimensional acceleration and gyroscopes, and establish the advantages of combining FSRs with other sensing modalities.

*This chapter is partly based on reference [OKL07].

7.1 Introduction

Much research in wearable context recognition has gone into tracking and recognizing arm and hand actions. The bulk majority of this research is based on motion sensors placed at different arm and hand locations, see e.g. [WLTS06, BI04]. This has both advantages and disadvantages. On the positive side, motion sensors, in particular accelerometers, are cheap, small and low power. At the same time a significant amount of activity information is contained in motion patterns. On the negative side, acceleration sensors tend to contain a mixture of motion and orientation information that is difficult to separate with simple sensor setups. In addition they provide no information on the palm and finger activity.

This work investigates an additional source of information about arms and hand actions: the analysis of arm muscle activity with force sensitive resistors (FSRs). Section 7.2 on the opposite page already gave more details on this sensor system.

FSRs are thin piezoelectric plates (see Figure 7.1 on page 104) that change their electrical resistance as mechanical force is applied to their surface. They are cheap and can easily be integrated in garments. It has also been shown ([MLT06]) that such mechanical pressure sensors can be implemented directly into textiles.

The idea behind our work is to use such sensors in an elastic sleeve worn on the forearm. This is fostered by the fact that palm and finger motions are driven by muscles in the forearm. As those muscles contract, they change their shape which in turn results in mechanical pressure being applied to the sensors in the elastic sleeve.

7.1.1 Contributions

Preliminary work [LHSS06] has already demonstrated the general feasibility of using FSRs to monitor leg muscle activity. [AJL+06] shows that different arm actions such as holding a heavy object or making a fist produce distinct FSR signals.

This work goes beyond such basic signal examples; it aims to investigate the use of FSRs for the actual recognition of a set of non-trivial manipulative gestures, see Section 3.3 on page 36. The aim is not to demonstrate the ability to reliably a specific, application relevant activity set recognize. Instead we want to lay the foundation for other researchers to set up FSR based activity recognition systems. To this end we present the following specific contributions:

- ✗ We describe details of the hardware implementation of the final FSR system and its advantages over previous implementations. The herein suggested implementation addresses key problems that were identified this preliminary work, i.e. large variations in the attachment force and sensor placement accuracy issues. The new system allows unobtrusive attachment of the sensors and makes post-attachment hardware calibration unnecessary, independent of attachment force and placement.
- ✗ We systematically investigate the performance of a FSR system with different classifiers on a set of 320 manipulative gestures from 16 different classes performed by two subjects. The classes have been chosen to test the limits of the system rather than to be recognizable with high accuracy. Thus the set contains some very subtle gestures.

- ✗ We compare the FSR performance to well-established sensing modalities, i.e. three-dimensional acceleration and gyroscopes. To capture hand rather than only arm motions (as do the FSR through muscle monitoring) the additional sensors are mounted on the back of the hand rather than on the wrist. Wrist mounting is actually more common since attaching devices to the hand is usually considered burdensome and requires at least a glove. However, we know from previous work that wrist-mounted motion sensors are not good at recognizing gestures primarily defined by hand motions so that hand-mounted sensors offer a more challenging comparison.
- ✗ We investigate the benefit of combining FSRs with additional sensing modalities by testing different combinations of the three sensor types (accelerometer, gyroscope, FSR).

7.2 A wearable FSR sensing prototype

7.2.1 The sensing hardware design

This section gives a short overview of what has to be considered when designing a FSR measurement circuit. However, no matter which strategy is used to build such a system, a major challenge is the fact that it is just possible to measure the differential muscle force; i.e. it is not possible to measure the muscle pressure force directly but the difference of force applied by the muscles to the most inner layer and the most outer layer of the arm-mounted sensors.

FSRs – available e.g. from Interlink[1] – are *polymer thick film devices*. The resistance of such a device decreases with an increase in the force applied to its active surface. According to Interlink, see [Int], FSRs are ideal for use in human touch control but are not suitable for precision measurements. The specifications are as follows:

- ✗ size range maximum: $51 \times 61cm$
- ✗ size range minimum: $0.5 \times 0.5cm$
- ✗ device thickness: 0.20 to $1.25mm$
- ✗ force sensitivity range: $<100g$ to $>10kg$
- ✗ pressure sensitivity range: $<0.1kg/cm^2$ to $>10kg/cm^2$
- ✗ part-to-part force repeatability: $+-15\%$ to $+-25\%$ of established nominal resistance
- ✗ single part force repeatability: $+-2\%$ to $+-5\%$ of established nominal resistance
- ✗ force resolution: better than 0.5% full scale
- ✗ break force (turn-on force): $20g$ to $100g$
- ✗ stand-off resistance: $>1M\Omega$
- ✗ switch characteristic: essentially zero travel
- ✗ device rise time: $1-2ms$ (mechanical)
- ✗ lifetime: $>10^{-6}$ actuations

[1] http://www.interlinkelectronics.com

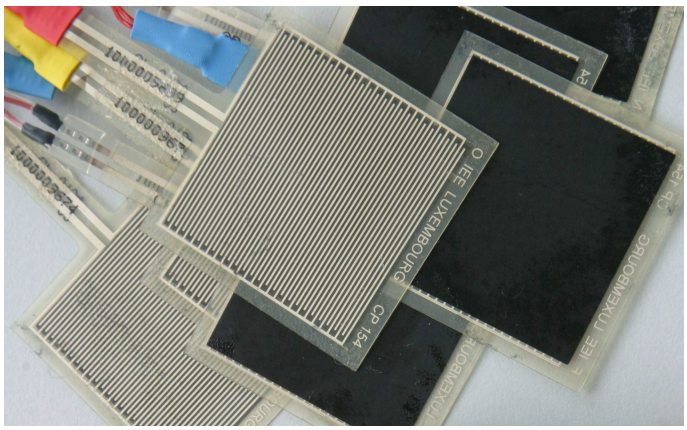

Figure 7.1
Examples of force sensing resistors (FSR). The picture shows square FSRs of size 46×46 mm as used throughout this thesis. FSRs are available in any other shape or size.

✗ sensitivity to noise/vibration: not significantly affected

7.2.1.1 Voltage divider solution

An obvious and straightforward method to measure a resistance value is simply using a voltage divider, see Figure 7.2 on the next page. Let V_{ref} be a known reference voltage and R_{ref} a known reference resistance. Let further be v_{out} the voltage drop along R_{ref} then v_{out} given a certain FSR resistance r_{fsr} is:

$$v_{out}(r_{fsr}) = \frac{R_{ref} \cdot V_{ref}}{R_{ref} + r_{fsr}} \tag{7.1}$$

From Eq. (7.1) output voltage vs. FSR resistance characteristics can be identified. Figure 7.3 on page 106 depicts different output voltage vs. FSR resistance graphs given different reference resistances and a certain reference voltage. The figure depicts also the quantification results assuming a 6 bit AD converter, i.e. 64 quantification steps.[2] What can be seen already from Eq. (7.1) becomes even more obvious in Figure 7.3: the correlation of resistance and voltage output is not linear at all. The quantification results illustrate that any given R_{ref} results in a decent resolution for only approximately two resistance decades. A typical FSR ranges from approximately 2Ω to approximately $2M\Omega$, i.e. the FSR resistance has a range of approximately 6 resistance decades. Thus the voltage divider solution is suboptimal for a force measurement where the entire sensor range is of interest. Though, this solution might still be useful for an initial test-run in case

✗ a narrow force range of interest $[F_1, F_2]$ can be identified,
✗ the resistance of R_{ref} is fine-tuned so that the highest resolution lies approximately in the middle of $[F_1, F_2]$.

This fine-tuning has to be done every time a subject is equipped with a FSR. In previous experiments we used a potentiometer for R_{ref}. Using a digital potentiometer an open loop controlled R_{ref} would be a decent solution as well, though inducing much higher demands on the micro-controller.

[2] Most implementations will of course use a higher resolution AD converter. 12 bit conversion is used in our case.

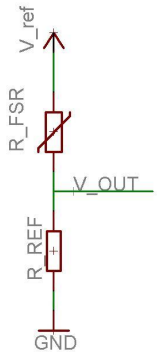

Figure 7.2
FSR resistance to voltage conversion using a voltage divider.

7.2.1.2 Current-to-voltage converter solution

The FSR generates different counter forces for differently attached sensors. When measuring the resistance of the FSR by means of a voltage divider according to Section 7.2.1.1 on the opposite page and [AJL+06, LHSS06] the initial counter force must be adapted for each sensor and each user individually due to the nonlinear relationship of resistance and output voltage. To circumvent this problem this approach is based on current measurement instead of voltage measurement; thus a current-to-voltage converter is required, see Figure 7.4 on page 107. This results in a linear relationship between resistance and output voltage and thus in an enhancement of the dynamic performance of the system. It furthermore scales better for different users and different ways of attaching the FSR.

7.2.2 Sensor placement and attachment

[LHSS06] describes the effect of sensor displacement on muscle activity monitoring. It was shown that a sensor displacement of just $1cm$ can lead to false or no signals. However, we have also demonstrated that this problem can be overcome by covering a larger area with the FSRs. This can either be accomplished by a matrix of sensors around the point of interest (as discussed in [LHSS06]) or by using large FSRs. In this work we combine both approaches. Both the lower part of the forearm (right above the wrist) and the upper part of the forearm (right below the elbow) are covered with a ring of four $46 \times 46mm$ FSRs. In such a way we circumvent the problem of slightly slipping sensors, see also Figure 3.3 on page 38.

The attachment of the sensors is another crucial question. FSRs require a moderate counter-force. On the other hand, the system should be *wearable*, i.e. easy to be put on and taken off and not too tight. As a consequence we opted for a three layer design: a thin inner layer (a thin stocking) on which the FSRs – the second layer – are fixed, and a third outer layer that is tight but stretchable. We considered an ordinary bicyclist's sleeve to be the right choice. That way we ended up with a sleeve that can be put on and taken off easily.

Future implementations of such muscle activity monitoring systems may rely on implementations that can be even more easily integrated into garments; e.g. Meyer *et al.* [MLT06] present a capacitive pressure sensor that can be integrated into textiles.

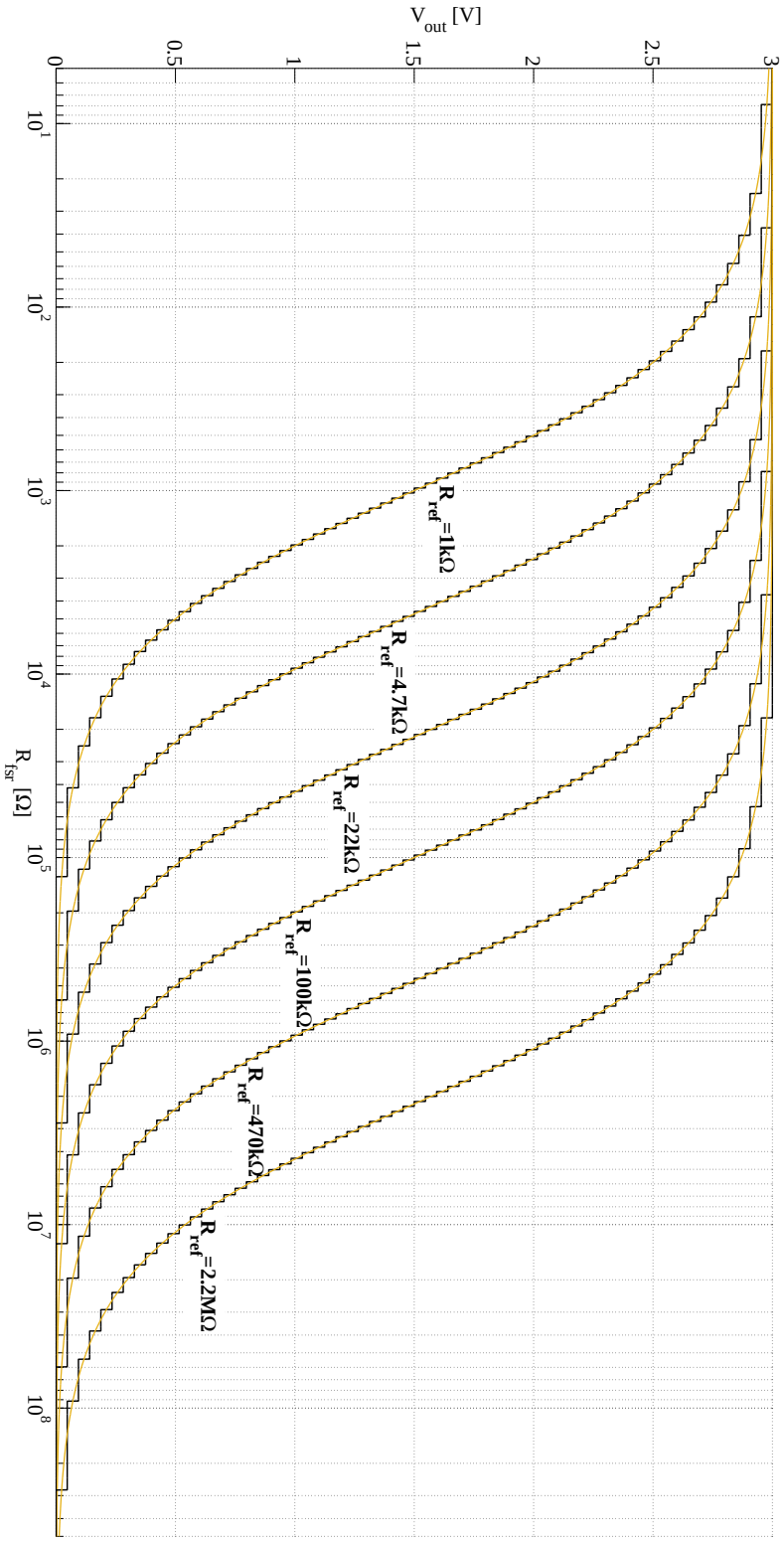

Figure 7.3
Voltage output v_{out} vs. FSR resistance r_{FSR} characteristics for the voltage divider solution as shown in Figure 7.2 on the previous page. The figure depicts the v_{out} vs. r_{FSR} characteristics for different values of reference resistance R_{ref} and a reference voltage $V_{ref} = 3V$. The solid cyan graphs depict the theoretical continuous results whereas the black graphs depict the discrete results after AD conversion using a 6 bit resolution AD converter.

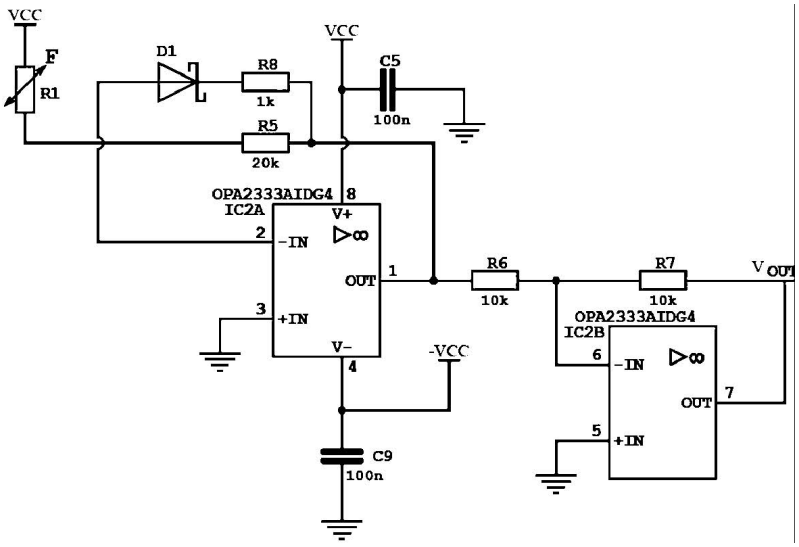

Figure 7.4
Schematic of the adopted current-to-voltage converter. Resistance $R1$ depicts the FSR, the first op-amp circuit comprises the actual conversion, the second op-amp circuit is needed for voltage level inversion.

7.2.3 The sensor system

The overall sensor system (Figure 7.5 on the next page) comprises a *tmote sky* from moteiv[3] with add-on boards featuring a current-to-voltage converter. To keep the size of the add-on board small we decided to multiplex the individual FSR channels, thus the add-on board also features an ADG708 able to multiplex up to eight FSRs. Tmote sky is featuring six AD converter channels thus parallel conversion would also be possible and would increase the sampling frequency by factor four in case of eight FSRs. Hence the actually applied strategy results in a significantly smaller maximum sampling frequency than the parallel approach; we adopted this approach anyway, due to the following considerations:

- ✗ For wearable and on-body sensors the size of the devices should be kept as small as possible, but the parallel approach would require a separate current-to-voltage converter circuit for each individual FSR channel.
- ✗ By multiplexing on one AD channel the other channels of the tmote sky platform are vacant for additional sensor modalities, e.g. by using six add-on boards we can multiplex 48 FSR channels. In such a way the system is ready to handle a matrix of FSRs, as is planned for future experiments.

The add-on board also comprises the LTC3455 from Linear Technology. The LTC3455 is a power management solution for battery driven USB devices, containing DC/DC converters, power controller and a Li-Ion battery charger. By means of this power management chip the final system can be both powered via USB and via Li-Ion battery. In addition the battery can be charged via USB during normal usage.

The following considerations are reasons for choosing the tmote sky platform – which is actually a *wireless sensor platform*

- ✗ handling different body parts without need for on-body wiring,
- ✗ well-documented, open source hardware platform (*tmote sky* is almost identical to *telos rev. B*[4] from UC Berkeley),

[3] http://www.moteiv.com
[4] http://webs.cs.berkeley.edu/tos/hardware/telos/telos-revb-2004-09-27.pdf

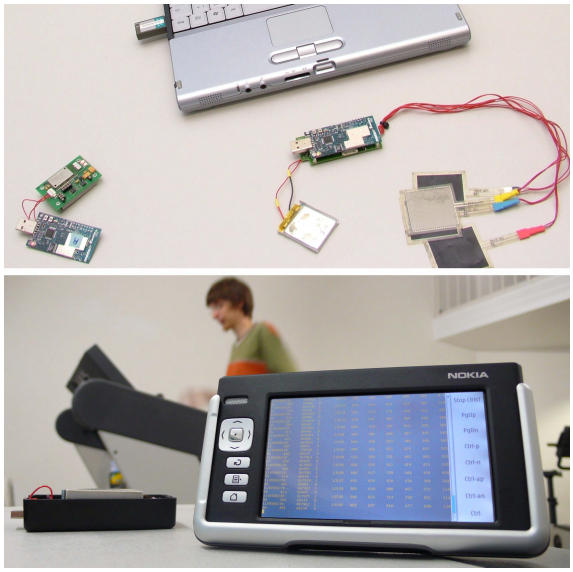

Figure 7.5
Two example setups comprising the tmote sky *based FSR system. The system can be interfaced via the IEEE 802.15.4 / Bluetooth-SPP bridge (both figures: the devices to the very left) to an ordinary computer (upper figure) or to a smart phone (lower figure). The smart phone was actually used as a recording and annotating platform for a sport experiment: a subject wearing the FSR system mounted on the thigh performing activities like running, walking, bicycling, hiking, etc.*

- ✗ open source embedded operating system, and
- ✗ availability of various interfaces, i.e. USB, I^2C, UART, SPI, and IEEE 802.15.4.

By means of using a wireless sensor platform we can easily distribute various FSR system on different body parts (legs, hands). A central node can collect the measurements and forward them to the recording system. The forwarding can be done in various ways, depending on the interfaces of the recording system. In case of an ordinary desktop-like computer being the recording system the central node will most likely just communicate the measurement readings by means of the USB interface. For various contemplated experiments – in sport scenarios in particular – lightweight platforms are more suitable like a smart phone or a wearable computer. These devices – as is the case with the Nokia device in Figure 7.5 – often lack a USB interface, thus we also built a *IEEE 802.15.4 / Bluetooth-SPP* bridge by simply attaching a Bluetooth-SPP device to the UART of the central node.

7.2.4 Calibration

In earlier versions of the sensing platform an experimenter had to – after equipping the test subject with FSRs – adapt one or more resistors to shift the voltage level of the FSR-resistor into a sensitive area, see e.g. Figure 7.3. As stated before the proposed new design makes hardware calibration unnecessary. The FSR-resistor is *always* in a sensitive area. Anyway calibration is needed to shift the baseline to the same level for any experiment and any test subject. Due to the fact that the data is only processed off-line the calibration was done off-line as well. What is more, test-runs usually last for only a couple of minutes and thus a static calibration was used.

7.3 Gesture recognition

For comparison, three classifiers have been tested. The tree based C4.5 classifier and the instance based k-Nearest-Neighbor (k-N-N) are used in a sliding window

Table 7.1

Recognition results per class for the k-N-N classification for different sensor modalities.

sensor	$\sum$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
acc	**81**	80	99	58	100	73	75	79	100	61	99	93	63	69	93	55	100
gyr	**65**	58	100	10	99	80	73	64	93	11	46	90	31	51	94	46	99
fsr	**76**	54	93	55	100	85	90	74	100	45	87	88	34	59	82	80	98
acc+gyr	**90**	91	100	63	100	94	92	81	100	80	97	98	83	89	100	82	100
fsr+acc	**86**	71	100	61	100	85	100	78	100	77	96	97	62	76	99	87	100
fsr+gyr	**84**	66	96	65	100	89	95	78	100	74	92	91	48	80	92	84	98

legend:
acc ... acceleration *gyr ... gyroscope*
fsr ... force sensitive resistor $\sum$ *... overall result*

Table 7.2

Recognition results per class for the HMM classification for different sensor modalities.

sensor	$\sum$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
acc	**83**	89	95	60	98	73	99	79	100	58	99	98	67	64	90	66	95
gyr	**72**	43	93	35	96	72	86	76	79	32	87	94	56	68	88	59	87
fsr	**73**	63	79	51	100	74	81	76	99	53	51	70	68	74	76	63	88
acc+gyr	**91**	87	96	69	100	92	99	88	100	74	98	100	85	95	100	74	97
fsr+acc	**84**	74	98	55	100	83	96	77	100	74	76	83	74	82	93	80	94
fsr+gyr	**81**	56	88	58	100	81	91	81	100	66	73	86	73	76	93	77	91

legend:
acc ... acceleration *gyr ... gyroscope*
fsr ... force sensitive resistor $\sum$ *... overall result*

approach: In a time window of fixed size, a set of features is computed using the raw sensor data. Then the sliding window is moved by an offset which determines the overlap with the last window. We use mean and variance as features, with window-size 30 and step-size 15. After that a majority decision is applied to the raw classification results. This yields a filtered decision for the particular gesture and constitutes the final result of the frame-based classification. We use the *YALE*[5] implementation of these classifiers.

In addition to this frame-based approach a Hidden Markov Model (HMM) based classifier is tested as well. For each manipulative gesture in our experiment a separate HMM is trained. During testing a single gesture is aligned with the most likely model. We use the HMM implementation in the *Bayes Net Toolbox*[6] for Matlab for our experiments.

7.4 Results

Due to the small data set we evaluate the data in a cross validation scheme. The classification results for the three classifiers are summarized in Table 7.4 on page 111, a per-class recognition rate for different sensor modalities for the k-N-N classification is given in Table 7.1, for the HMM classifier in Table 7.2. In addition Table 7.3 on page 111 gives the confusion matrices of the k-N-N classification for three different sensor combinations: acceleration sensors, acceleration and gyroscopes, and acceleration in combination with FSRs. The following points are the main observations:

[5] http://www-ai.cs.uni-dortmund.de/SOFTWARE/YALE/index.html
[6] http://www.cs.ubc.ca/~murphyk/Software/BNT/bnt.html

- ✗ The recognition is far from perfect for all combinations of sensors and all classifiers. This was to be expected, as the gesture set was chosen to test the limits of the recognition rather than to be fully recognizable. As can be seen from Table 7.3 most errors occur typically for do/undo gesture pairs. Such pairs of classes are: $C_{1,3}$, $C_{5,7}$, $C_{9,12}$, $C_{13,15}$.
- ✗ For all classifiers the overall accuracy of the FSR system is in the middle between the accelerometers (which is between 5% and 10% better) and the gyroscopes (which is between 2% and 11% worse). This is also not surprising, since the accelerometers are mounted on the hand rather than the wrist. Thus – just like FSRs – they provide information not just on motions, but also on grasping which causes vibration on the back of the hand. At the same time the FSRs have less clear signals due to placement issues and, what is more, lack some motion information. On the other hand, gyroscopes lack the grasping information (reaction to the vibrations is minimal).
- ✗ Adding FSRs to other sensors always leads to an improvement (between a minimal 1% for HMM and acceleration and 19% for k-N-N and gyroscope). This is clear for the gyroscopes. For accelerometers it indicates that there is indeed some information that the FSRs have and an accelerometer – even when hand-mounted – does not have. This is further confirmed by the fact that even in a single sensor case there are gestures for which FSRs perform better than accelerometers.

7.5 Conclusion

As force sensing resistors have not been used extensively for monitoring muscular activities, the chapter suggested a FSR platform intended to be wearable and applicable for different on-body sensing scenarios. Compared to previous approaches the suggested sensing platform makes no hardware calibration necessary. Moreover, the suggested platform allows a body-network of FSRs and provides state-of-the-art wireless communication.

The main lesson of this chapter is that FSR based muscle monitoring is indeed useful for the recognition of activities involving hand actions. While being inferior to accelerometers mounted on the hand for the overall accuracy on the 16 gestures, FSRs perform well for many individual gestures. In a few cases they are even better than the accelerometer, which confirms that there is some grasping related information that even hand-mounted motion sensors cannot detect. Note that in our experiment FSRs rely on much less obtrusive arm mounting which makes them preferable for many applications.

7.6 Limitations

Although the gesture set in the experiment has been chosen to be diverse, it cannot be claimed to be representative in any systematic way. In addition, even for the same gesture set the performance for a given sensing modality depends on fine-tuning of features, window sizes, etc. – factors that were not explored systematically in this

Table 7.3

Confusion matrices of the k-N-N classification for three different sensor combinations: acceleration sensors (upper table), accelerometers and gyroscopes (middle), and accelerometers in combination with FSRs (lower table).
The results are given in % of the overall amount of ground-truth events.

	Prediction \ Ground-truth	1 - open pen	2 - write	3 - close pen	4 - erase	5 - screen down	6 - point	7 - screen up	8 - remote	9 - open notebook	10 - type	11 - mouse	12 - close notebook	13 - open bottle	14 - glass	15 - close bottle	16 - drink
accelerometers	1 - open pen	80		9													
	2 - write	6	99	12			13	2									
	3 - close pen	2		58													
	4 - erase		1	7	100		9	1								2	
	5 - screen down	3		3		73	3	8		3							
	6 - point	3				5	75	1									
	7 - screen up	3		3		18		79		1			1	3			
	8 - remote			8		3		3	100								
	9 - open notebook									61		1	18		5		
	10 - type									6	99	5	4	8		2	
	11 - mouse							3		7	1	93	4			5	
	12 - close notebook									8			63				
	13 - open bottle							3					1	69		30	
	14 - glass	3								6			1	4	93	6	
	15 - close bottle									8		1	3	16	2	55	
	16 - drink	1				1							5				100
accelerometers + gyroscopes	1 - open pen	91		7													
	2 - write	1	100	16			7	2									
	3 - close pen			63				3									
	4 - erase	3		2	100												
	5 - screen down	3		1		94	1	6									
	6 - point						92										
	7 - screen up					6		81									
	8 - remote			10				5	100								
	9 - open notebook									80		2	4			2	
	10 - type									2	97		4	3			
	11 - mouse							3		8	3	98	4			5	
	12 - close notebook									5			83				
	13 - open bottle													89		9	
	14 - glass	2		1						1			4		100	2	
	15 - close bottle									4				8		82	
	16 - drink			1									1				100
accelerometers + FSRs	1 - open pen	71		10													
	2 - write	13	100	19													
	3 - close pen	8		61													
	4 - erase	2		5	100			1		1							
	5 - screen down	1				85		14									
	6 - point	1					100										
	7 - screen up	2				15		78									
	8 - remote	1		4				7	100								
	9 - open notebook									77			2				
	10 - type	1								5	96	3	7	4		9	
	11 - mouse									5	3	97	15	1	1		
	12 - close notebook									2			62	2			
	13 - open bottle									2	1			76		4	
	14 - glass									2			6	1	99		
	15 - close bottle	1								6				16		87	
	16 - drink			1									8				100

Table 7.4

Classification results in percentage of the overall amount of gestures.

classifier	acc	gyr	fsr	acc+gyr	fsr+acc	fsr+gyr
HMM	83	72	73	91	84	81
C4.5	76	57	62	82	79	68
k-N-N	81	65	76	90	86	84

legend:
acc ... acceleration — *gyr ... gyroscope*
fsr ... force sensitive resistor — *$\sum$ ... overall result*
HMM ... Hidden Markov Models — *k-N-N ... k-Nearest-Neighbor*

work. Thus the lessons and conclusions discussed above must be taken as indicative rather than proven beyond doubt.

7.7 Future work

So far the calibration is done offline in a static way. Thus an important next step will be to investigate in more detail and possibly to automate the calibration of the system to different users. An auto-calibration procedure could also compensate changes in the force signal baseline caused by slipping sensors.

Chapter 8

Multi-modal continuous spotting and recognition*

This chapter describes an approach to real-life task tracking using a multi-modal, on-body sensor system. The specific example that we study will be quality inspection in car production. This task is composed of up to 20 activity classes such as checking gaps between parts of the chassis, opening and closing the hood and trunk, moving the driver's seat, and turning the steering wheel. Most of these involve subtle and short movements and have a high degree of variability in the way they are performed.

To spot those actions nonetheless in a continuous data stream we use a wearable system composed of seven motion sensors, 16 FSRs for lower arm muscle monitoring and four UWB tags for tracking user position. We propose a recognition approach that deals separately with each activity class and then merges the results in a final reasoning step. This allows us to fine-tune the system parameters separately for each activity. It also means that the system can easily be extended to accommodate further activities.

In order to demonstrate the feasibility of our approach we present the results of a study with eight participants and a total of 2394 activities.

*This chapter is based on reference [OSLT08].

8.1 Introduction

In this chapter we look at a complex, realistic case study closely modeled after a real industrial application; including initial data recordings in the real production line at a Škoda car factory. We focus on how highly multi-modal sensor systems – 27 on-body sensors in our specific case study – can be used in a flexible, modular way. Thus the methods described in this work allow addition and removal of sensors with little changes to the rest of the recognition systems. Similarly new activities can be added without impact on the recognition system regarding the old activities. Finally the sensors and algorithms used for spotting can be selected and fine-tuned separately for each activity.

As described in Section 3.4 on page 38 the investigated scenario is closely modeled after a real-life quality assurance procedure in a car assembly factory, namely the Škoda factory in Mladá Boleslav. The aim is to track the progress of a quality inspection procedure at the end of the car production line. The procedure involves activities such as opening the trunk, door and hood, sliding the hands over parts of the car to detect gaps, and moving parts such as the steering wheel and the seats. For a complete list refer to Table 3.3 on page 40.

The sensor system used in that case study consists of seven motion sensors (each is a combination of a three-axis accelerometer, a three-axis gyroscope and a three-axis digital compass) monitoring the motions of different upper body parts, 16 FSRs to monitor arm muscle activity and a high accuracy indoor location system to determine the position of the user with respect to the car.

8.1.1 Contributions

- ✗ *Complexity and degree of multi-modality of the recognition architecture.* Our recognition system uses 27 sensors (seven motion sensors, 16 muscle activity sensors and four location tags at different upper body locations). The benefit of our approach is illustrated in the discussion of the experiment results (see Section 8.3.4 on page 124).
- ✗ *Modularity of the recognition system with respect to sensors and algorithms.* As described in Section 8.2 we use additional sensors in form of consecutive independent masking passes. This means that adding or removing a sensor modality or an additional algorithm has no impact on the rest of the recognition system.
- ✗ *Modularity with respect to activity classes.* Our recognition architecture has an independent spotting process for each activity (see Section 8.2 on the opposite page). New activities can be added without any impact on the recognition systems for the old ones. Also, sensors and algorithms used for spotting can be selected and fine-tuned separately for each activity.
- ✗ *Specific methods for individual sensing modalities.* We present two methods for removal of insertion errors from a pre-segmented signal stream. One uses a motion sensor based Bayesian filtering approach (see Section 8.2.5 on page 119), the other one is based on muscle activity information from the arm-mounted FSRs (see Section 8.2.4 on page 118).

  In addition we demonstrate how information from the individual activity

spotting processes can be consistently and efficiently combined into a single event stream (see Section 8.2.6 on page 121).

- ✗ *Complexity and realism of the case study.* In the case study we apply our method to a car quality inspection task. The task is a real procedure performed at the end of the production line at a Škoda factory in the Czech Republic, see also [SRO+08]. The experiments were performed on a real Škoda car according to videos and initial test recordings with identical sensor setup done during a test run at the factory.

8.2 Approach

Our approach is based on the following ideas:

- ✗ *Activity separation.* We treat each activity as a separate event stream. Thus for N activities we have N independent spotting processes. Each process is responsible for a single activity and determines where events corresponding to this activity occur in the data stream. The final output of the spotting system is the union of the outputs of the individual processes. By contrast most previously published work (including work of our groups) combines spotting with multi-class recognition from the start. Our approach allows a class-dependent selection of sensors, features and algorithms. It enables easy addition and removal of classes.
- ✗ *High recall initial spotting stage.* We start each process with a fast spotting stage that provides an initial guess about possible locations of the relevant class while removing obviously non-relevant signal segments. It is optimized for high recall and low precision to ensure low deletion rates.
- ✗ *Incremental masking passes.* We incrementally improve the precision of the system through the application of a sequence of what we call *masking passes*. A masking pass works on the signal segments that have been identified as possible occurrences of a given class by the initial spotting stage. It makes a binary decision of either retaining or rejecting the segment. Each masking pass uses different algorithms and/or sensors. The sequence of passes can be determined class-dependently. This approach allows easy removal/addition of sensors or algorithms without changing the rest of the system.

8.2.1 Recognition process overview

Figures 8.1 on the following page and 8.2 on page 117 give an overview of the recognition process. The detailed overview also includes the hardware architecture; there are three main parts:

- ✗ the distributed sensor systems, their synchronization and annotation,
- ✗ the parallel sensor-dependent spotting and recognition modules which implement the masking passes and
- ✗ the fusion of the parallel results.

An essential part of a distributed sensor system is the synchronization of the data streams and their annotation for training. For recognition systems trained in a

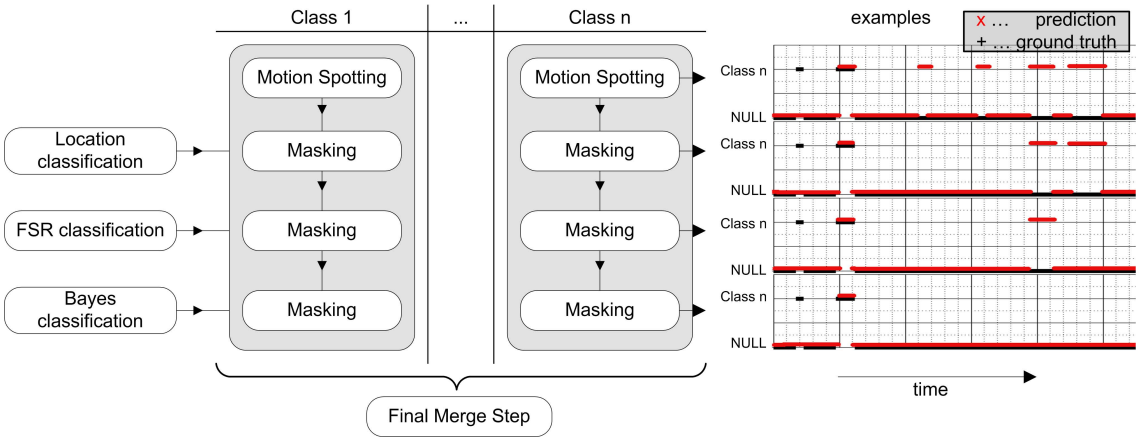

Figure 8.1
Overview of the recognition process. The examples on the right depict how the different masking steps may influence the recognition result.

supervised manner also the possibility to annotate events of interest is crucial. We use the framework proposed by [BAL08, BKL06] to cope with these issues. The individual spotting and classification modules are described in the following sections.

8.2.2 Position preprocessing

The worker's relative position to the car body is estimated using an ultra-wide-band (UWB) system from Ubisense (see Section 4.3.2.3 on page 53). Tags on the shoulders of the worker enable the system to calculate the worker's position with respect to four reference base stations placed in the environment.

To avoid extensive data loss because of the huge amount of metal materials at the assembly line the worker is equipped with four tags on his/her shoulders. Before the resulting positions are fused they are processed with a Kalman smoother.

There was a significant loss of Ubisense data of 28.7% on average per tag, which was decreased by this fourfold redundant approach to an average of 6.5% per subject.

8.2.2.1 Location classification

The worker's location is defined according to seven location classes, see Table 8.1 on page 118. The classes are modeled by k-means clustering with $k = 7$. The Euclidean distance between the estimated position vector and each location class center is calculated and the closest class center is assigned to the worker's current location class. A sliding window median filter is applied to the classifier result to omit outliers.

The location classes are statically defined in relation to the car. In the final application the location processing has to be adopted to the fact that the car is moving with the constant velocity of the conveyor belt.

8.2.3 Motion based spotting

The motion trajectory based spotting scheme was already briefly explained in Section 6.4 on page 83. For a more detailed description refer to [Sti08, SRO+08,

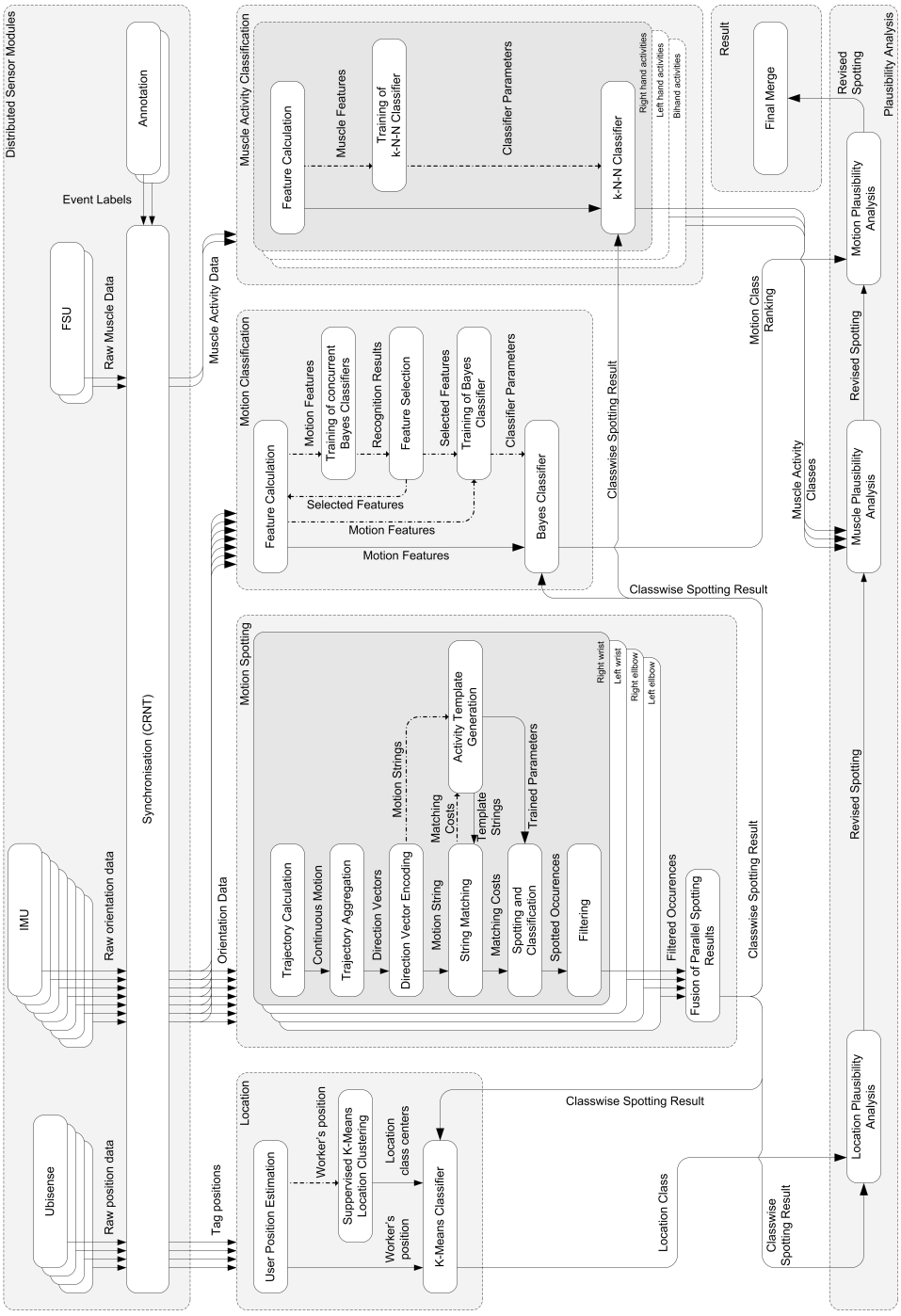

Figure 8.2

Detailed overview of the recognition process. The boxes display the individual processing modules; arrows demonstrate the data flow whereas dashed arrows show additional paths during different training steps. Note that the plausibility analysis can also be applied to any previous spotting result, e.g. per sensor spotting result right before the fusion step.

Table 8.1
Location classes

class ID	class name	class ID	class name
1	hood	5	trunk
2	front left	6	rear right
3	inside	7	front right
4	rear left		

SRT07b, SRT07a]. This section will summarize the method as it was applied in the car assembly scenario.

In order to transform the continuous-valued trajectory data into a discrete symbol space appropriate for string matching, a quantization step is performed. This quantization is achieved by fragmenting the trajectory in motion direction vectors of equal spatial length. Based on its direction each motion vector is mapped to a symbol of a finite alphabet. The sequence of resulting symbols is referred to as motion string. Approximate string matching [NR02] is used to spot activity occurrences in the motion string. The weighted edit distance or weighted Levenshtein distance provides a measure for the similarity of two strings.

During the training stage one template string for each activity class is found by computing the minimum edit distance among all training instances which corresponds to the minimum linkage distance of the training instances. Matching costs between the template string and all training instances of that class are calculated by aligning the template string with the motion string of all training instances. The mean μ_i and the standard deviation σ_i of these matching costs are computed for each activity class i and a class-related threshold ϑ_i is derived: $\vartheta_i = \mu_i + \nu \cdot \sigma_i$ where ν is a parameter which needs to be optimized for an individual data set.

In the spotting stage the matching cost for each template string with the current motion string is computed resulting in a stream of matching costs for each class. Within these cost streams, local minima are detected. When local minima are below threshold ϑ_i for class i, a spotted occurrence of the particular class i in the current cost stream is reported. The start point of the potential occurrence can be estimated by identifying the previous local maximum in the cost stream. Due to the class-dependency of the templates an implicit classification is performed.

The afore described activity spotting process is performed in parallel on four trajectory streams. The detected potential activity occurrences found in individual streams are combined to produce the fused results. An overlap detection is applied which detects segments of high consistence among the individual streams. The individual streams are weighted and a class-dependent threshold is applied to decide whether a fused segment is produced.

8.2.4 Muscular activity

The muscle activity of the lower arms is measured using two custom-built sleeves [OKL07]. In order to incorporate the muscle data into the activity recognition process, three k-Nearest-Neighbor (k-N-N) classifiers are trained using

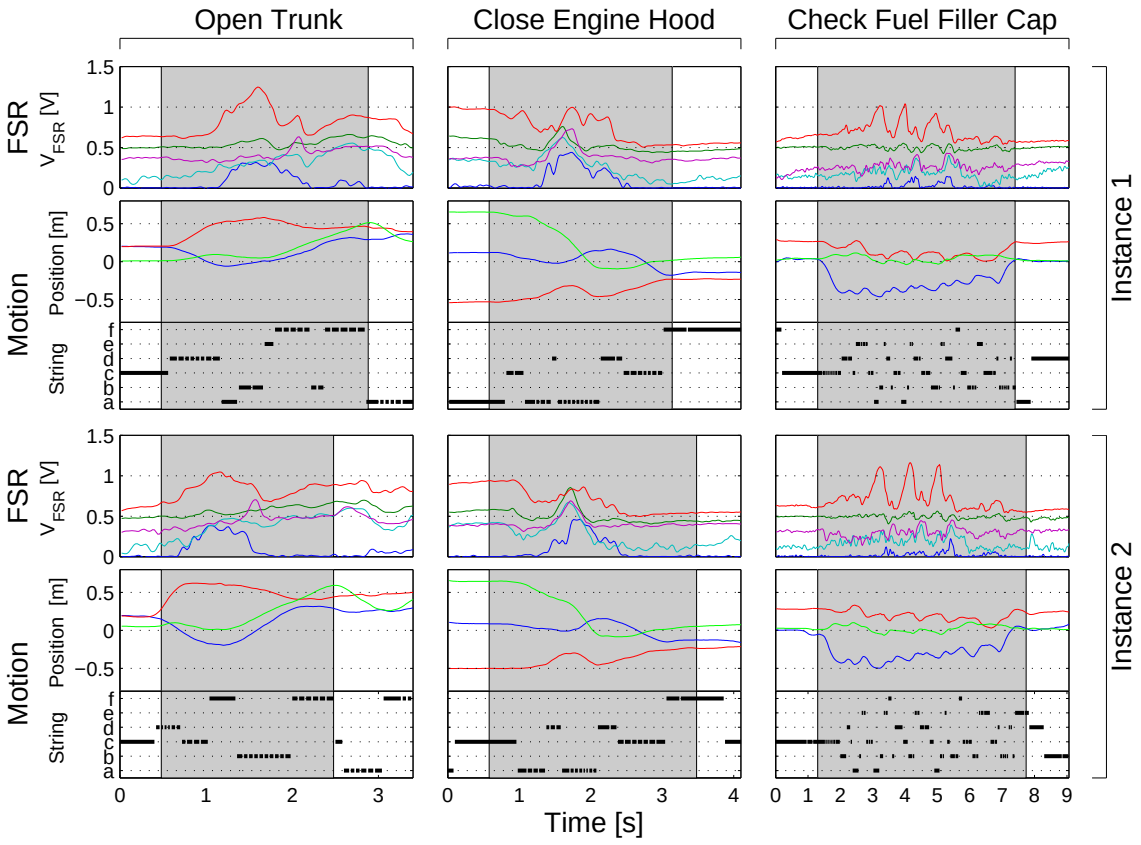

Figure 8.3
Visualization of sensor signals for individual gestures from the checkpoint scenario.

the *WEKA*[1] k-N-N implementation with $k = 5$. Each of the three sub-classifiers is responsible for a subset of classes; classifier 1 for right hand activities, classifier 2 for left hand activities and classifier 3 for bi-manual activities. This results in a threefold concurrent prediction.

8.2.5 Bayes motion classification

One of the processing steps to remove potential insertion errors from the retrieved activity events is a naive Bayes classification applied to motion data.

8.2.5.1 Features

The classification is based on features that are derived from the inertial measurement units (IMUs) that are also used for the activity spotting operation. The sensors deliver calibrated readings for acceleration, rate of turn and magnetic field. Each sensor domain provides its readings in three dimensions. In addition, the orientation of each of the sensor modules is provided in Euler angles format (roll, pitch, yaw). All used features are computed on temporal segments which can be seen as a time series of a multivariate sensor signal. We derive six features from the left and right hand respectively: mean and variance of acceleration over three axes, mean and variance of the Euler component *roll* and the start and stop angle value of this

[1] http://www.cs.waikato.ac.nz/ml/weka/

component for a given time segment. For both upper and both lower arms we compute the following four features: mean and variance of the Euler component *pitch* and the start and stop angle value of this component.

We use seven features based on the IMU that is aligned with the torso: mean and variance of the Euler component *pitch*, minimum and maximum of the Euler component *pitch*, variance of the Euler component *roll*, and variance and maximum of the gyroscope signal from the vertical axis.

8.2.5.2 Feature selection

For each of the activity classes in our experiment, we select an individual subset of the 35 available features. We developed the following procedure to achieve a class-dependent feature selection. First, we compute all 35 features for all training instances of all activity classes. From these features we create 35 univariate naive Bayes classifiers which are used to classify all training instances sequentially. The resulting probability ranking of a certain classifier (considering one feature) for a given training instance is then inspected to check whether the probability of the correct class of the segment is above a threshold ϑ_i for class i. If this is true for fraction f of the class' training instances, we select the feature under examination for the classification of the given class. That way we identify those features that carry relevant information to increase the chance for a correct classification of a certain activity class.

8.2.5.3 Training

In the case of a naive Bayes classifier with continuous multivariate variables training the classifier means to find the likelihood in Eq. (8.1).

$$posterior = \frac{likelihood \times prior}{normalizing\ \ constant} \tag{8.1}$$

The *posterior* stands for the probability of an activity class given a certain observation which is the multivariate sensor reading in our case. For our problem the likelihood is the probability of seeing the measured observation given that it is produced by a certain activity class. The *prior* is equally distributed among all activity classes. We can neglect the *normalizing constant* because it is equal for all activity classes. The likelihood is trained by finding a probabilistic model based on the observations of the training instances. We use Gaussian distributions for our activity classes models.

8.2.5.4 Classification

During classification we compute the *posterior* of all activity classes for a signal segment that has been retrieved by the spotting and masking operations before. That way we collect a probability ranking of the segment at hand. During the spotting operation that retrieved the segment an activity class is implicitly assigned. The probability ranking of the Bayes classification allows to check this implicit assignment for plausibility. Such a check requires one parameter, which determines the rank up to which we want to find the initial class label in order not to discard the segment.

8.2.6 Merging the parallel spotting streams

All the steps described so far are performed in parallel by the spotting processes devoted to individual activity classes. Thus instead of a single system output we have n separate outputs with n being the number of classes. If all the spotting processes were 100% correct, at any given time at most one stream could contain an activity while all the others would have to indicate the NULL class since the user performs only one activity at a time. We thus apply the heuristic proposed in Section 2.3.1 on page 24 for the resolution of conflicts.

8.3 Results and discussion

This section summarizes the results achieved when applying the afore-mentioned methods to the data recorded in the car assembly scenario (see Section 3.4 on page 38). Most results are given in terms of *precision and recall* merits. All error count methods – including precision and recall – are defined in an analogous manner to the definitions given in Section 6.7.3 on page 95 and in Appendix B on page 139.

8.3.1 Individual activities

Figure 8.4 on page 123 shows precision and recall for three different class sets.

- ✗ *open/close hood (1, 2), check/close trunk (4, 5), mirror (13), trunk gaps (14), open spare-wheel-box (18):*
  The first plot contains one third of the activities (seven out of 20). For those activities the system achieves very good performance with an average precision of 84.5% and an average recall of 87.1%.
- ✗ *open trunk (3), fuel lid (6), open/close right door (9, 10), lock check (15, 16), close spare-wheel-box (19):*
  The second plot depicts the results of another seven activities that go down to 60% to 70% on one or both measures (average precision 58.5%, average recall 85.2%).
- ✗ *open/close left door (7, 8), open/close two doors (11, 12), hood gaps (17), writing (20):*
  Finally on the remaining six activities the system can be said to fail as one or both of the measures drop below 50% (average precision 28.5%, average recall 49.5%). Those extremely poorly performing classes pull down the average performance of all activities, which is precision 47.8% and recall 70.6%.

8.3.2 Merged results

The results of merging the individual event streams into a single output are shown in Table 8.2 on the following page. It can be seen that even for the entire set of 20 classes the merging process does not introduce a significant amount of deletions. For the final step the recall remains on the same level.

Table 8.2

Results after the merge step of exemplary intermediate steps averaged over all gesture classes given as a percentage of the overall amount of ground truth events. Fragmentations refer to events that were split into several parts (see Appendix B on page 139 and references [WLT06, War06] for an exact definition).

	motion spotting	masking step (FSR)	masking step (location)	final
correct	80.4	76.2	80.7	70.3
deletions	1.3	8.1	8.8	18.7
insertions	427.3	175.1	140.1	52.4
substitutions	22.7	19.6	15.6	11.2
fragmentations	4.5	3.8	5.0	2.0

8.3.3 Discussion of individual activity errors

8.3.3.1 Door related activities

The bulk, i.e. eight out of 13, of the activities with mediocre and poor spotting results are related to doors (activities 7-12, 15, 16). There are two main reasons for this:

- ✗ Opening and closing the door allows a huge degree of variability in the way it can be performed. There is only a short characteristic part: pulling the handle to unlock the door. After that one can continue to pull on the handle, or use one or the other hand placed on an arbitrary part of the frame to finish opening the door. In the case of opening both doors at the same time much of the pull comes from body motion as it is difficult to pull both doors at the same time without moving the body. Such slow body motions are much more difficult to recognize than distinct arm activities and occur more often during random activities.
- ✗ Opening and closing the doors also occurred as part of other checking activities without being annotated as an activity for itself. This is particularly grave for the left door where the muscle activity classifier fails (recall below 83.4%), effecting the muscle plausibility analysis to fail as well.

8.3.3.2 Checking the gaps

Checking the gaps on the trunk (activity 14) is among the best recognized activities. Checking the hood gaps (activity 17), on the other hand, belongs to the poorly performing activities. At a first glance this may be surprising as these two activities seem related. However, a close look reveals considerable differences. To check the trunk gaps the user stands in the middle behind the car and moves both hands from the top sidewards to the bottom. In the process the user has to bend down. This is a very characteristic sequence of motion that is unlikely to occur in other activities. Checking the hood, on the other hand, consists of running the fingers along the more or less horizontal line between the hood and the fender. This is a subtle motion combined with a posture that can occur in other unrelated activities.

8.3.3.3 Writing

Writing (activity 20) is among the poorly performing classes. This is due to the fact that it is not bound to a certain location, has no large characteristic gestures and

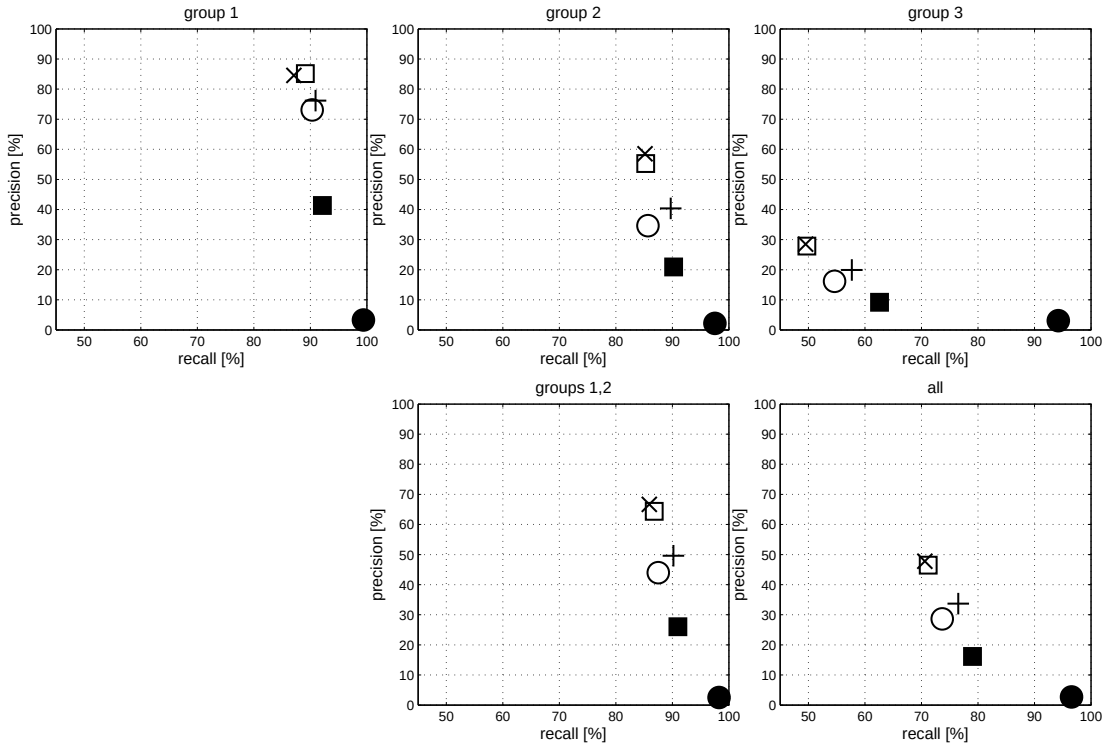

Figure 8.4
Precision and recall plots for different class sets. The markers depict particular results: motion spotting (black disk), muscle activity classification (black square), masking step (FSR) (white disk), masking step (location) (+), masking step (FSR+location) (white square), and final result (x). Precision is given as a percentage of the overall amount of predicted events and recall as a percentage of the overall amount of ground truth events.

little distinguishable muscle activity. The only specific aspect is the arm posture, i.e. lower arms in horizontal position, which is just not enough for reliable spotting. More information on finger motion and subtle hand motions would be needed here.

8.3.3.4 Checking the fuel lid

Checking the fuel lid (activity 6) is in the middle group. With an excellent recall of 91.4% but just 54.6% precision it is just out of the *good* group. The activity has a characteristic turning motion associated with opening and closing the lid and a reasonably well-defined location. On the other hand it is fairly short and subtle; and similar activities can occur in unrelated activities. This accounts for the poor precision.

8.3.3.5 Hood, trunk and mirror activities

The remaining activities, i.e. hood, trunk and mirror activities (activities 1-5, 13, 18, 19), can be said to be well defined from both the location and the upper body motion point of view. Due to that six out of these activities result in good and the remaining two in mediocre recognition results.

8.3.4 Discussion of sensing modalities

8.3.4.1 Summary results

Figure 8.4 on the preceding page shows the precision and recall curve after the application of different masking approaches to the original spotting result. It can be seen that initial spotting achieves a very good recall (79.0% for all activities, 92.1% for the seven *good* ones) at the cost of an excessive insertion rate (precision 16.2% and 41.3% respectively). First we have a significant jump in the precision (28.6% for all and 73.1% for the seven good classes) associated with a significant drop in the recall (73.7% for all and 90.3% for the good classes) as a first masking technique is applied. From there comes a steady, significant increase of precision (47.8% for all and 84.5% for the good classes) with only a relatively small drop in the recall (70.6% for all and 87.1% for the good classes).

8.3.4.2 Results on selected classes

Figure 8.5 on the next page shows the precision and recall curves for the individual activities. Three main observations result from an analysis of the precision and recall plots of individual activities.

- ✗ While the performance of the individual masking approaches varies greatly from activity to activity, the combination of all approaches consistently remains best or very close to best for all classes. The only exception are the left door classes, on which, however, results are very poor for all approaches.
- ✗ With the exception of activity 8 (close left door) the initial fast spotting sweep nearly perfectly achieves the goal of high sensitivity and has a recall in the nineties.
- ✗ The precision and recall plots of the activities can be put into four groups.
  - ✗ In the first group the application of the different masking techniques produce steep gain in precision (up to around 90%) with no or little loss of recall. *Open hood (1), check/close trunk (4, 5), mirror (13), trunk gaps (14), open spare-wheel-box (18)* belong to this group. In short this result means that our sensor system is perfectly matched to capture the unique characteristics of these activities.
  - ✗ In the second group – *fuel lid (6), open right door (9), lock check (15, 16)* – the filtering also retains high recall but the gains in precision levels are on average around 60%.
  - ✗ In the third group – *open trunk (3), close right door (10), close spare-wheel-box (19), writing (20)* – we see both, the leveling off of the precision gains and a significant drop in the recall.
  - ✗ Finally we have the group of activities *open/close left door (7, 8), open/close two doors (11, 12), hood gaps (17)* – where the system can be said to fail and filtering leads to a large drop in recall with no adequate gain in precision. For these activities we can conclude that additional or different sensing modalities are unavoidable.

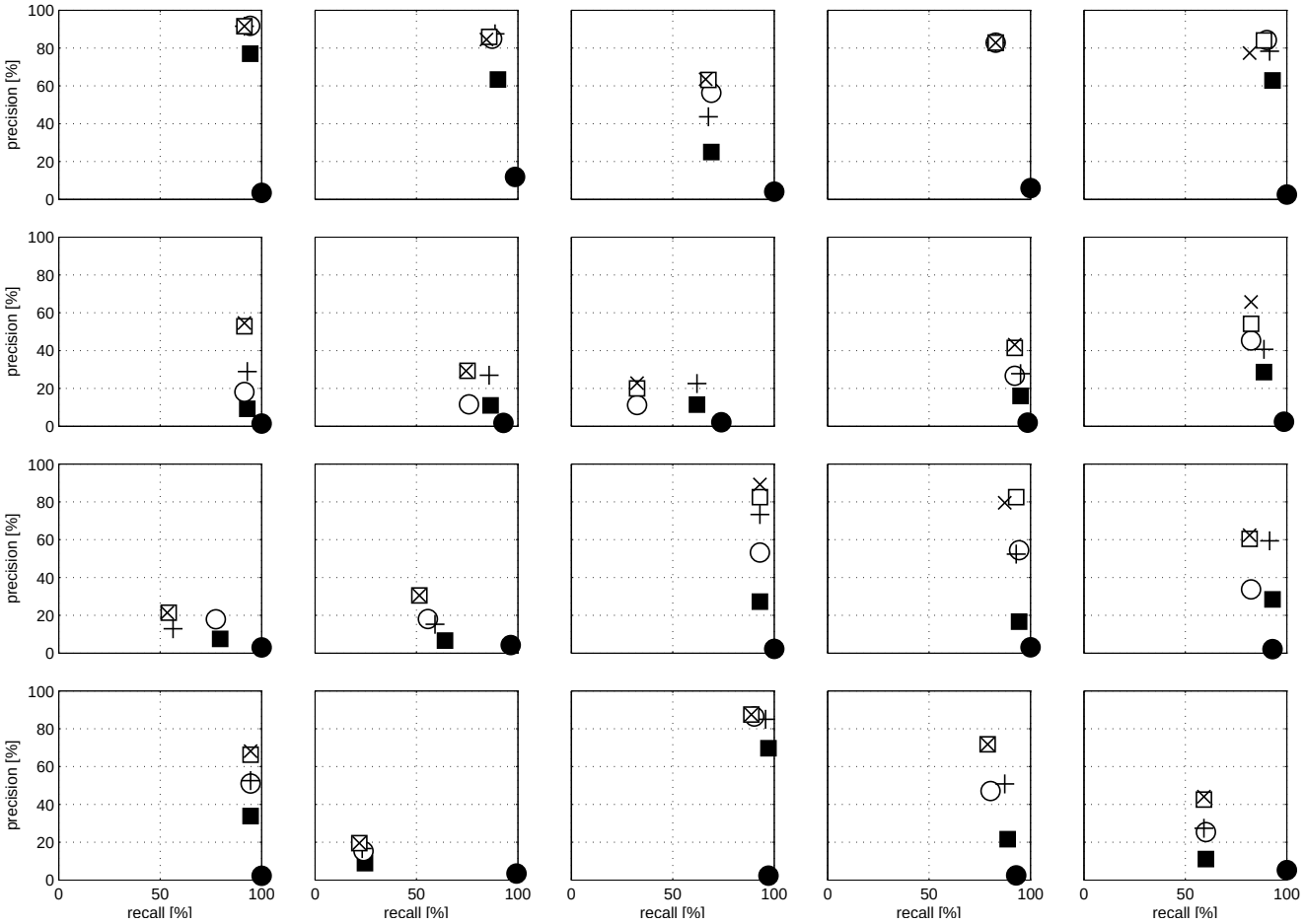

Figure 8.5

Precision and recall plots for the individual classes, see also Figure 8.4 on page 123. Plots are sorted according to class numbers, read from left to right and from top to bottom:

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 spare-wheel-box*	*close spare-wheel-box*	*writing*

8.3.5 Lessons learned

From the results given and discussed above a number of conclusions can be drawn that we believe to be significant beyond the specific car inspection example.

- ✗ *Importance of consistent class definition.* The poor performance of the door related activities underscore the importance and the difficulty of a consistent definition of an activity. As described in the previous section the problem can be traced to the similarity of activities belonging to different door closing classes and an overlap with parts of other activities (which also require opening the door).
- ✗ *Slight variations in the task setup can be crucial.* Seemingly similar activities performed under slightly different circumstances can be dramatically different in their recognition complexity. This is illustrated by the example of checking the hood and the trunk gaps. From a high level task description both sound similar and aim to achieve the same thing. However, in terms of recognition

difficulty they are very different.

- ✗ *Inadequate sensing for subtle gestures.* The FSRs have been included in the setup as a way to get information about subtle palm and finger motion related to activities such as for example writing. It turns out that this did not work as expected. As discussed in the previous section the FSRs clearly improve the overall performance of the system. However, they do not provide sufficient information to achieve the original goal. There are two conclusions from this observation. One is the need to improve the FSR system. The other is to add further sensing modalities such as sound, proximity sensing, or even wrist-mounted cameras.
- ✗ *Merit of separate recognition chains for each activity.* Our system has made extensive use of the possibility to tune the parameters for each of the investigated activities independently. This tuning has significantly improved the results. At the same time we have seen that the final fusion step does not introduce many deletions.
- ✗ *Merit of the incremental masking approach.* While it is difficult to find sensing modalities that fully capture the unique characteristics of certain activities, it is often easy to find sensors that contain certain necessary (but not sufficient) conditions for these activities to occur. The FSRs are a prime example of such sensors. While the signals from our system are often not unique to a certain activity, we can say that some activities cannot have occurred.

  The same can be said about user location with respect to the car. The incremental masking approach is perfectly suited to exploit this. The improvements achieved through the individual masking passes and the variations in the effect of the passes on different activities confirm that. As the main performance issue is insertion errors, it can be assumed that further masking passes with new sensing modalities (e.g. sound) should produce considerable improvements.

8.4 Conclusion

As described in Section 8.2 on page 115 the core ideas of our approach were: (1) to start with a fast spotting stage that has very poor precision but very good recall, (2) improve the precision through incremental application of masking passes based on different sensor modalities and recognition algorithms that can be flexibly added and removed, and (3) to treat each activity as a separate event stream for which spotting can proceed with different methods and sensors.

Overall the results of the experiment have confirmed the usefulness of this approach. The initial spotting produces few deletions. For more than a third of the investigated activities the masking passes increase the precision without a significant impact on the recall. Merging the individual event streams introduces few additional errors. At the same time masking passes can be added, removed and their sequence can be varied flexibly. The performance for individual activities can be tuned and analyzed independently.

Our approach has failed for about a third of the activities. The reasons for this failure are predominantly inadequate characterization of those activities by the

sensors used in the experiment and issues with consistent activity class definition. Interestingly, those failures actually underscore the merit of our approach. Adding additional sensors and masking passes to improve the recognition of the problematic activities will not require any changes in the processing chain of the well-recognized classes, neither will an adjustment of the definition of the activity classes (e.g. putting the door classes together). In the latter case only the final merge stage will have to be adapted.

On the negative side we have seen that sensing modalities, and in some cases possibly the specific algorithms we have used, were not sufficient to characterize some activities.

Chapter

9

Summary

This chapter concludes the thesis, gives a short summary of the achievements and a short outlook on future research questions.

9.1 Summary of achievements

With a rather specific wearable application area in mind – maintenance and production in industrial environments – this thesis has investigated a novel on-body sensing combination to its limitations according to its applicability to the recognition of manipulative hand gestures.

FSRs

The thesis has evaluated the combination of location and motion sensing techniques with force sensing based muscular activity recognition. To this end a wearable sensing platform for muscular activity monitoring has been designed and implemented. This sensing technique has been evaluated according to its aptitude to recognize basic manipulative hand gestures. The evaluations show that activities with strong muscular activities (e.g. pulling the screen up and down, erasing text on the whiteboard) are recognized at a rate of up to 100% and thus FSRs are outperforming or at least equal to motion sensing. On the other side subtle gestures (e.g. open/close a pen, open/close a notebook) are hardly recognized by means of FSR muscular activity sensing. Nevertheless by means of adding FSRs as an additional sensing modality the performance of motion based activity recognition can be significantly increased. In the car assembly scenario the achieved increase in the precision rate goes even up to 15%.

Ultrasonic hand tracking

Moreover, ultrasonic based hand tracking has been used together with on-body motion and orientation sensing to suggest a novel approach to on-body sensing based gesture spotting and recognition. The thesis has outlined an approach to fuse the motion and orientation readings with the position estimations, resulting in a hand trajectory with a far better dynamic response than the mere slow sampling positioning system provides. In addition, the thesis has described and evaluated different methods for location modeling, location based gesture spotting, motion based gesture spotting, and methods for fusing intermediate, class-wise results.

Location modeling

We have shown that both supervised and semi-supervised location modeling techniques can be employed for the envisioned activity recognition scenarios. The location information introduces user-independence to the recognition approach. Moreover mere *intra* location modeling seems to provide too little diverse location training data. Whereas for motion based gesture recognition *intra* training schemes usually outperform *inter-subject* trained models, this does not seem to be valid for location features. Thus hand locations can and should be modeled by means of a set of recordings providing a decent diversity.

Location based spotting

We have shown that location trajectory based spotting is a promising approach (above 90% correct spotted gestures). We have suggested a modification (LDA classification of polynomial matching cost features) for the motion trajectory based spotting approach presented by [Sti08, SRO+08, SRT07b, SRT07a].

Location based recognition

Finally, the suggested recognition approach achieves recall rates above 90% at precision rates above 95% no matter whether user-dependent or user-independent models are applied. As expected do/undo gesture pairs (e.g. tightening/loosening a specific screw) are easily substituted.

Multi-modality

The thesis has described an approach to real-life gesture spotting and recognition by using a multi-modal sensor system. To recognize gestures in a continuous data stream the work has suggested to use a wearable system composed of seven motion sensors, 16 force sensing resistors for lower arm muscle monitoring and four ultra-

wide-band tags for tracking the user's position. Overall the results of the experiment have confirmed the usefulness of the suggested approach with precision and recall above 80% for some classes. Other activities still demand additional optimization, e.g. additional sensing modalities.

The continuous recognition experiments apply an recognition approach that deals separately with each gesture class. This allows the system to be fine-tuned separately for each activity of interest. It also allows to extend the recognition system by means of accommodating further gesture classes in a simplified way. The approach applies independent, consecutive masking passes or plausibility checks to consecutively increase the high recall spotting stage. For each sensing modality an independent masking pass can be trained. Thus sensing modalities can be exchanged, added, removed with only little impact on the performance of the recognition system. *Modularity*

9.2 Conclusion

The results of the location based spotting and recognition experiments confirm the user-independent nature of location features. The results indicate that the use of user-independent sensing modalities and features outperforms an artificial user-independent training by means of multi-user training sets. *User-independence*

A multi-modal approach addresses the challenge of large heterogeneity inherent in activity spotting and recognition. Thus a multi-modal sensing approach increases the range of gestures that can be recognized. On the other the side multi-modality demands higher integration of the hardware components and increases the demands for the processing unit and the power supply. These are essential questions when designing wearable devices. One will have to balance these issues when setting up wearable context recognition systems. *Multi-modality*

The modular class-wise approach seems to be a promising approach for activity recognition. As shown by the comparison of current results with results achieved by a preliminary recognition approach confirms the usefulness of this approach. *Modularity*

Moreover we believe that modular setups are a key feature when designing commercial activity recognition systems. Another key feature are unsupervised learning schemes, which was just touched within this thesis by means of the semi-supervised location modeling approaches. We presume that unsupervised modeling of human gestures and activities will play a major role in future research, and modularity can be one of the enabling techniques.

9.3 Outlook

- ✗ We have shown that FSR based muscle monitoring is indeed useful for the recognition of activities involving hand actions. However, they do not provide sufficient information to achieve persuasive recognition results when used for subtle gestures. An important next step will be to investigate in more detail and possibly to automate the calibration of the system to different users. An auto-calibration procedure could also compensate changes in the force signal baseline caused by slipping sensors.

  Future work will also investigate new and improved sensing modalities.

This includes improvements in the FSR setup and addition of sensing modalities such as microphones and RFID readers which are known to be useful from previous research.

- ✗ Future work will look at further improving the segmentation methods for the location trajectories in order to facilitate better precision of the initial spotting stage. Moreover the suggested approach will be tested on other sensor modalities as well.
- ✗ We will further investigate the use of higher level activity models in particular in conjunction with the final merge step. This will also include looking at combinations of errors made in parallel on different event streams as means of improving spotting performance.
- ✗ We intend to record data sets from different applications to be able to verify our method not just on one case study but over a broader range of conditions.
- ✗ The modular approach might be used to achieve a self-configuring recognition system. This aims at automatically selecting the best suited sensing modalities in case new activities are added or a specific sensing modality is malfunctioning. Future experiments will consider this demand and investigate concepts to accomplish this goal.

Appendix

Kalman filtering

Chapter A

This chapter gives a quick introduction to the Kalman filter algorithm.

A.1 Introduction

The Kalman filter (see also [BH97]) can be applied for linear systems defined by a state transition model and a measurement model. In case either one is non-linear, the Kalman filter in its original definition cannot be applied. The *extended Kalman filter* is one possible way to linearize such models.

The state transition model must describe the transition concerning two consecutive states, thus the system must be of the form

$$\mathbf{x}_k = \mathbf{F}_{k-1} \cdot \mathbf{x}_{k-1} + \mathbf{w}_{k-1} \tag{A.1}$$

with $\mathbf{x}_k$ being the state vector at time t_k and $\mathbf{F}$ being the state transition matrix. $\mathbf{w}_k$ represents the system noise at time t_k. The noise is assumed to be white and independent. The Kalman filter furthermore expects a linear measurement model of the form

$$\mathbf{z}_k = \mathbf{H}_k \cdot \mathbf{x}_k + \mathbf{v}_k \tag{A.2}$$

with $\mathbf{z}$ being the measurement vector, $\mathbf{H}$ being a matrix giving the ideal connection between the measurement and the state vector. Vector $\mathbf{v}_k$ represents the measurement error; again it is supposed to be a white sequence with zero mean and known covariance structure.

The extended Kalman filter allows a measurement model of the form

$$\mathbf{z}_k = h(\mathbf{x}_k) + \mathbf{v}_k \tag{A.3}$$

with h being a known, differentiable function, defining the relation between measurements and the state vector analogous to Eq. (A.2). To apply the extended Kalman filter state transition and observation model must be of the form

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{u}_k) + \mathbf{w}_k \tag{A.4}$$

$$\mathbf{z}_k = h(\mathbf{x}_k) + \mathbf{v}_k \tag{A.5}$$

with $\mathbf{u}_k$ being a known, deterministic force function.

The Kalman filter *prediction* step is done according to

$$\hat{\mathbf{x}}_{k|k-1} = f(\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_k) \tag{A.6}$$

$$\mathbf{P}_{k|k-1} = \mathbf{F}_k \mathbf{P}_{k-1|k-1} \mathbf{F}_k^T + \mathbf{Q}_k \tag{A.7}$$

and the *update* step according to

$$\tilde{\mathbf{y}}_k = \mathbf{z}_k - h(\hat{\mathbf{x}}_{k|k-1}) \tag{A.8}$$

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^T \left(\mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{R}_k \right)^{-1} \tag{A.9}$$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \tilde{\mathbf{y}}_k \tag{A.10}$$

$$\mathbf{P}_{k|k} = (I - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k|k-1} \tag{A.11}$$

In case of the extended Kalman filter the state transition matrix $\mathbf{F}_k$ and observation matrix $\mathbf{H}_k$ are defined to be the following Jacobeans

$$\mathbf{F}_k = \left. \frac{\partial f}{\partial \mathbf{x}} \right|_{\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_k} \tag{A.12}$$

$$\mathbf{H}_k = \left. \frac{\partial h}{\partial \mathbf{x}} \right|_{\hat{\mathbf{x}}_{k|k-1}} \tag{A.13}$$

$\mathbf{Q}_k$ and $\mathbf{R}_k$ represent the covariance matrices of system noise $\mathbf{w}_k$ and measurement error $\mathbf{v}_k$, respectively. The following notation is used in the previous:

$\hat{\mathbf{x}}$	...	state estimation for state vector $\mathbf{x}$
$\mathbf{P}$	...	error covariance matrix of the current state estimation
$\mathbf{K}$	...	Kalman filter gain
$\tilde{\mathbf{y}}$	...	measurement residual
$k\|k-1$	...	state prediction at time t_k given measurements and state estimations until time t_{k-1}
$k\|k$	...	state estimation at time t_k given measurements and state predictions until time t_k
I	...	Identity matrix

Spotting evaluation

Chapter B

This chapter gives a quick overview of the evaluation metrics used throughout this thesis.

B.1 Introduction

This chapter describes the performance measures applied throughout this thesis. As shown by Ward *et al.* [WLT06, War06], state-of-the-art evaluation measures of classification results such as *precision and recall curves (PR)* or *receiver-operator characteristics (ROC)* and others are too little significant merits when applied in the field of activity and gesture spotting. These measures often cause ambiguous and delusive results.

B.2 The SET measure

[WLT06, War06] present a performance evaluation method called *segment error table (SET)* considering the following aspects:

- The method evaluates frame-based and event-based errors in an analogous manner.
- It considers substitutions.
- It considers event merges and event fragmentations.
- Activity or gesture recognition case studies rely on manually annotated data streams, thus the ground truth event have fuzzily defined event boundaries and so do the recognized events. The SET metric tries to account for this timing problem.
- It scales for n number of classes not only for two as e.g. ROC does.
- It results in an unambiguous evaluation table.

This section will give a short summary of the SET metric because the results presented in this thesis rely on that evaluation measure; moreover, we will suggest an extension to the method.

The SET method affords that the resulting *prediction/ground-truth stream* is fragmented into segments that may not contain either a change in the ground-truth stream or a change in the prediction stream; and each segment border must cover a change in at least one of these streams. We will refer to these segments as *bi-unique segments (BUS)*.

Each BUS is then assigned either the label *match* or one of the following *error pair labels* (for details refer to [WLT06, War06]):

- *event error pairs*
  - *Insertion-Deletion (ID)*
  - *Insertion-Fragmentation (IF)*
  - *Merge-Deletion (MD)*
- *timing error pairs*
  - *Insertion-Underfill (IU)*
  - *Overfill-Deletion (OD)*
  - *Overfill-Underfill (OU)*

These errors pairs can further be interpreted from the ground-truth point of view:

Table B.1
Segment error table (SET) with NULL (N) as special case

	D	U	F	D_N	U_N	F_N
I	ID	IU	IF	ID_N	IU_N	IF_N
O	OD	OU		OD_N	OU_N	
M	MD			MD_N		
I_N	$I_N D$	$I_N U$	$I_N F$			
O_N	$O_N D$	$O_N U$				
M_N	$M_N D$					

- $Deletions = ID + OD + MD$
- $Underfill = IU + OU$
- $Fragmentations = IF$

or from the prediction point of view:

- $Insertions = ID + IU + IF$
- $Overfill = OD + OU$
- $Merges = MD$

These are referred to as *error labels* in contrast to the above listed *error pair labels*. The error counts in the SET can be given both in number of frames or in number of BUS. A SET can be generated in different ways, e.g.:

- for the overall result,
- for pairs of classes; this way the SET can be interpreted as an extended version of the *confusion matrix* where each non-diagonal entry contains one single SET, and each diagonal entry contains the number of matches of a specific class; or
- for the overall result accounting for two special cases: (1) the prediction is the NULL class, (2) the ground-truth is the NULL class.

The latter results in a threefold table, see Table B.1 taken from [War06]. Within this thesis we use additional overfill and underfill labels to be more precise. Thus

- a *pre-overfill* error accounts for an overfilled ground-truth event with the prediction starting too early (*preemption*),
- a *post-overfill* error accounts for an overfilled ground-truth event with the prediction stopping too late (*prolongation*),
- a *pre-underfill* error accounts for an underfilled ground-truth event with the prediction starting too late (*delay*), whereas
- a *post-underfill* error accounts for an underfilled ground-truth event with the prediction stopping too early (*shortening*).

B.3 SET extension

The SET approach can be used as an evaluation metric both when summarizing results and when evaluating intermediate steps, i.e. while parameters are optimized or concurrent algorithms are compared with each other. Due to the fact that a varying number of intermediate spotting and recognition steps within this thesis contain what we call *concurrent prediction streams*, the SET is not directly applicable because it does not scale for more than one prediction stream. We suggest an extension to the SET that preserves its unambiguousness to some extent but is able to handle concurrent prediction streams. To describe this problem in more detail first we need to define two special sub-sequences within the *prediction/ground-truth* stream.

A *NULL BUS* specifies a segment containing the NULL class in all prediction streams and the ground-truth stream. A special NULL BUS with $length = 0$ is defined by a border that does not intersect any prediction event nor a ground-truth event. For each BUS that is not a NULL BUS we can then define its *adjacencies* according to the sub-sequence that (1) contains that BUS, (2) is bordered by a NULL BUS on each side, but (3) does not contain any NULL BUS itself. Adjacencies can either contain predictions of only one single class (*non-concurrent adjacencies*) or predictions of different classes (*concurrent adjacencies*).

[War06] defines an error pair label assignment directive that finds the SET for any given prediction/ground-truth stream. The algorithm loops over the whole BUS sequence and assigns each one of them either the label *match* or one of the *error pair labels*. Two inner loops are necessary to assign the IF and the MD label. In this manner SET is able to handle concurrent prediction streams in case of a NULL BUS and in case of non-concurrent adjacencies, but it cannot handle concurrent adjacencies.

Our extension simply suggests another inner loop for the concurrent adjacencies that assigns a match label or one of the error pair labels to each prediction/ground-truth pair that does not contain a NULL class prediction element just in the same manner as the SET directive suggests. NULL BUS and non-concurrent adjacencies are processed as suggested by the original assignment directive. Thus we end up with an evaluation table that sums up to a value bigger than the overall number of evaluated frames or the overall number of BUS.

To cope with that issue a *label assignment sequence* has to be defined with the label *match* being the first to be assigned. The first label out of the sequence that suites is assigned normally whereas any subsequent label assigned obtains the annex *extra*. In this manner the final table sums up to the overall number of evaluated frames (or BUS) for the original error pair labels whereas the sum of the derived error pair labels defines an excess of predicted events and gives also a clue on the degree of concurrency.

As mentioned before, this SET extension preserves the unambiguousness only to some extent: the extended SET is ambiguous according to the label assignment sequence. Changes in this sequence will shift labels from the SET to the *extra* part and vice versa. One would have to define a sequence that best fits the contemplated application.

B.4 Event-based error measure

In addition to the SET assignment directive [War06] suggests an unambiguous event error assignment directive, assigning each prediction event and each ground-truth event an error label defined in analogous manner to the BUS error labels listed in the previous section, resulting in an *Event Error Table (EET)*.

Two additional labels are defined: *no ground-truth label (nG)* is assigned to any correctly recognized ground-truth event in case the respective prediction event accounts for a merge error and analogous to that *no prediction label (nP)* is assigned to any correct prediction in case the respective ground-truth event accounts for a fragmentation error.

In addition, one can differentiate between substitution and insertion errors, with insertions being prediction events that account for a NULL deletion error and substitutions being prediction events that account for a non-NULL deletion error. Unfortunately the SET extension is not reasonable applicable to the EET assignment directive.

As a straightforward solution, we suggest to apply the EET assignment directive in a twofold manner. In the first stage the event errors are assigned on each *class-dependent prediction/ground-truth* stream separately, i.e. for each class it is applied on a prediction/ground-truth stream that has been cleaned from any event (prediction event or ground-truth event) that is not of this specific class type; whereas cleaning means *replaced by a NULL event of the same length.* In such a way we end up with a n-fold event error table, with n being the number of classes. Accumulating these class-wise EETs results in the *accumulated, class-wise EET (acEET)*.

Evidently, acEET over-estimates the insertion error rate because any substitution error will be counted as an insertion error. Thus this acEET has got two disadvantages:

- ✗ It is unable to assign substitution errors.
- ✗ Insertion errors are over-estimated.

In the second stage the n-fold prediction/ground-truth stream will be merged into an 1-fold prediction/ground-truth stream, in order to count the number of substitution errors. In order to achieve that first the n-fold prediction/ground-truth stream is *fragmented* into a BUS stream. The following mapping directive is then applied aiming to map each n-fold BUS into an 1-fold BUS (no changes are made to the ground-truth element of the BUS)

- ✗ Any NULL BUS is mapped to a NULL BUS.
- ✗ A non-NULL BUS containing no correct match is mapped to a BUS containing the prediction element with the lowest class identification number.
- ✗ A non-NULL BUS containing a match is mapped to a BUS containing the matching prediction element.

Applying then the EET assignment directive on this (*de-fragmented*) prediction/ground-truth stream results in substitution and insertion error rates that give just a rough estimation of how many percent of the insertion error rate achieved by the acEET are actually substitution errors.

Evidently, the first stage will result in an unambiguous EET whereas the second stage results in an EET that is difficult to be interpreted because of the artificial prediction event separations, e.g. a false prediction event accounting for a substitution error may be separated into two prediction events by a matching event. Mapping an n-fold BUS into a 1-fold BUS causes of course information loss and thus under-estimated error rates. On the other hand, the event separating effect may cause over-estimation, e.g. considering the example given before, in case the matching event causing the artificial prediction event separation is underfilling the appropriate ground-truth event on both sides the false prediction event will then account for two substitution errors.

Thus the results of this second stage should always be contrasted with the result of the acEET achieved in the first stage. Nevertheless, in case the concurrencies are not too complex the second stage gives a clue on the actual proportion of insertion to substitution errors.

B.5 Precision and recall

In order to evaluate the final recognition performance the *precision and recall* metric is still a viable evaluation measure. These values can be directly calculated from the EET according to:

$$recall = \frac{\#C + \#F + \#nG}{\#G} \tag{B.1}$$

$$precision = \frac{\#C + \#M + \#nP}{\#P} \tag{B.2}$$

with $\#X$ denoting *overall amount of events being labeled with* X.

Note that all continuous precision and recall results given within this thesis are based on these definitions. In case of a processing stage with a n-fold intermediate spotting or recognition stage, these event errors are assigned in a class-wise manner by means of applying the acEET metric (as described in the previous section).

References

[ABK+08] Kurt Adamer, David Bannach, Tobias Klug, Paul Lukowicz, Marco Luca Sbodio, Mimi Tresman, Andreas Zinnen, and Thomas Ziegert. Developing a wearable assistant for hospital ward rounds: An experience report. In *Proceedings of the International Conference for Industry and Academia on Internet of Things (IOT 2008)*, pages 289–307, Zürich, Switzerland, March 26-28 2008.

[ACH+01] Mike Addlesee, Rupert Curwen, Steve Hodges, Joe Newman, Pete Steggles, Andy Ward, and Andy Hopper. Implementing a sentient computing system. *IEEE Computer Magazine*, 34(8):50–56, August 2001.

[AJL+06] Oliver Amft, Holger Junker, Paul Lukowicz, Gerhard Tröster, and Corina Schuster. Sensing muscle activities with body-worn sensors. In *Proceedings of the International Workshop on Wearable and Implantable Body Sensor Networks (BSN 2006)*, pages 138–141, Washington, DC, USA, April 3-5 2006.

[AJT05] Oliver Amft, Holger Junker, and Gerhard Tröster. Detection of eating and drinking arm gestures using inertial body-worn sensors. In *Proceedings of the 9th IEEE International Symposium on Wearable Computers (ISWC'05)*, pages 160–163, Osaka, Japan, October 18-21 2005.

[AM06] Ian Anderson and Henk Muller. Qualitative positioning for pervasive environments. In *Proceedings of the 3rd International Conference on Mobile Computing and Ubiquitous Networking (ICMU 2006)*, pages 10–18, London, UK, October 11-13 2006.

[Amf] Oliver Amft. Marker - A Matlab time-series labeling toolbox. http://www2.ife.ee.ethz.ch/~oam/projects/marker/.

[Amf08] Oliver Amft. *Automatic dietary monitoring using on-body sensors: Detection of eating and drinking behaviour in healthy individuals.* PhD thesis, ETH Zürich, Zürich, Switzerland, 2008.

[AMGC01] Sanjeev Arulampalam, Simon Maskell, Neil Gordon, and Tim Clapp. A tutorial on particle filters for on-line non-linear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 50:174–188, 2001.

[AMS02] Stavros Antifakos, Florian Michahelles, and Bernt Schiele. Proactive instructions for furniture assembly. In *Proceedings of the 4th International Conference on Ubiquitous Computing (UbiComp 2002)*, pages 351–360, Göteborg, Sweden, September 29 - October 1 2002.

[AS02] Daniel Ashbrook and Thad Starner. Learning significant locations and predicting user movement with GPS. In *Proceedings of the 6th IEEE International Symposium on Wearable Computers (ISWC'02)*, pages 101–108, Seattle, WA, USA, October 7-10 2002.

[AS03] Daniel Ashbrook and Thad Starner. Using GPS to learn significant locations and predict movement across multiple users. *Personal and Ubiquitous Computing*, 7(5):275–286, October 2003.

[BAL08] David Bannach, Oliver Amft, and Paul Lukowicz. Rapid prototyping of activity recognition applications. *IEEE Pervasive Computing*, 7(2):22–31, April-June 2008.

[BAS96] Michael S. Brandstein, John E. Adcock, and Harvey F. Silverman. Microphone-array localization error estimation with application to sensor placement. *Journal of the Acoustical Society of America*, 99(6):3807–3816, June 1996.

[BAS97] Michael S. Brandstein, John E. Adcock, and Harvey F. Silverman. A closed-form location estimator for use with room environment microphone arrays. *IEEE Transactions on Speech and Audio Processing*, 5(1):45–50, January 1997.

[BBC+03] Roberto Battiti, Mauro Brunato, Renato Lo Cigno, Alessandro Villani, Roberto Flor, and Gianni Lazzari. WILMA: An open lab for 802.11 hotspots. In *Proceedings of Personal Wireless Communications (PWC 2003)*, pages 163–168, Venice, Italy, September 23-25 2003.

[BBS+08] Stacy J. Morris Bamberg, Ari Y. Benbasat, Donna Moxley Scarborough, David E. Krebs, and Joseph A. Paradiso. Gait analysis using a shoe-integrated wireless sensor system. *IEEE Transactions on Information Technology in Biomedicine*, 12(4):413–423, July 2008.

[BH97] Robert Grover Brown and Patrick Y.C. Hwang. *Introduction to random signals and applied Kalman filtering*. Wiley, third edition, 1997.

[BHP+06] Helene Brashear, Valerie Henderson, Kwang-Hyun Park, Harley Hamilton, Seungyon Lee, and Thad Starner. American sign language recognition in game development for deaf children. In *Proceedings of the 8th International ACM SIGACCESS Conference on Computers and Accessibility (Assets 2006)*, pages 79–86, Portland, OR, USA, October 23-25 2006.

[BI04] Ling Bao and Stephen S. Intille. Activity recognition from user-annotated acceleration data. In *Proceedings of the 2nd International Conference on Pervasive Computing (Pervasive 2004)*, pages 1–17, Linz / Vienna, Austria, April 21-23 2004.

[Bil98] Jeff A. Bilmes. A gentle tutorial of the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. Technical Report TR-97-021, Computer Science Division Department of Electrical Engineering and Computer Science U.C. Berkeley, April 1998.

[BK00] Matthew Brand and Vera Kettnaker. Discovery and segmentation of activities in video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):844–851, August 2000.

[BKL06] David Bannach, Kai Kunze, and Paul Lukowicz. Distributed modular toolbox for multi-modal context recognition. In *Proceedings of 19th International Conference on Architecture of Computing Systems (ARCS 2006)*, pages 99–113, Frankfurt/Main, Germany, March 13-16 2006.

[BL08] Gerald Bauer and Paul Lukowicz. Developing a sub-room level indoor location system for wide scale deployment in assisted living systems. In *Proceedings of the 11th International Conference on Computers Helping People with Special Needs (ICCHP 2008)*, pages 1057–1064, Linz, Austria, July 9-11 2008.

[BNSS01] Michael Boronowsky, Tom Nicolai, Christoph Schlieder, and Ansgar Schmidt. Winspect - a case study for wearable computing supported inspection tasks. In *Proceedings of the 5th IEEE International Symposium on Wearable Computers (ISWC'01)*, pages 163–164, Zürich, Switzerland, October 7-9 2001.

[BP00] Paramvir Bahl and Venkata N. Padmanabhan. RADAR: An in-building RF-based user location and tracking system. In *Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2000)*, pages 775–784, Tel Aviv, Israel, March 26-30 2000.

[BP03] Hari Balakrishnan and Nissanka Bodhi Priyantha. The Cricket indoor location system: Experience and status. In *Proceedings of the Workshop on Location-Aware Computing (UbiComp 2003)*, pages 7–9, Seattle, WA, USA, October 12 2003.

[Che08] Hao Chen. *Non-Intrusive Computing*. PhD thesis, University of Waterloo, School of Computer Science, Kitchener-Waterloo, Canada, December 2008.

[CSMC06] K. W. Cheung, H. C. So, W.-K. Ma, and Y. T. Chan. A constrained least squares approach to mobile positioning: algorithms and optimality. *EURASIP Journal on Applied Signal Processing*, 2006:1–23, 2006.

[dlHT08] Patrick de la Hamette and Gerhard Tröster. Architecture and applications of the FingerMouse: A smart stereo camera for wearable computing HCI. *Personal and Ubiquitous Computing*, 12(2):97–110, January 2008.

[DMC+05] Paul Duff, Michael McCarthy, Angus Clark, Henk Muller, Cliff Randell, Shahram Izadi, Andy Boucher, Andy Law, Sarah Pennington, and Richard Swinford. A new method for auto-calibrated object tracking. In *Proceedings of the 7th International Conference on Ubiquitous Computing (UbiComp 2005)*, pages 123–140, Tokyo, Japan, September 11-14 2005.

[EGK91] Sylvan Elhay, Gene H. Golub, and Jaroslav Kautsky. Updating and downdating of orthogonal polynomials with data fitting applications. *SIAM Journal on Matrix Analysis and Applications*, 12(2):327–353, April 1991.

[FHK+03] Dieter Fox, Jeffrey Hightower, Henry Kauz, Lin Liao, and Don Patterson. Bayesian techniques for location estimation. In *Proceedings of the Workshop on Location-Aware Computing (UbiComp 2003)*, Seattle, WA, USA, October 12 2003.

[FHL+03] Dieter Fox, Jeffrey Hightower, Lin Liao, Dirk Schulz, and Gaetano Borriello. Bayesian filtering for location estimation. *IEEE Pervasive Computing*, 2(3):24–33, July-September 2003 2003.

[FMPS06] G. Ferraiuolo, Gianluca Massei, L. Paura, and Amedeo Scarpiello. *Distributed Cooperative Laboratories: Networking, Instrumentation, and Measurements*, chapter NeBULa (Network-Based User-Location Sensing System): A Novel and Open Location Sensing Framework operating on a WLAN Environment, pages 511–526. Springer US, January 2006. editors: Franco Davoli and Sergio Palazzo and Sandro Zappatore.

[Fox05] Eric Foxlin. Pedestrian tracking with shoe-mounted inertial sensors. *IEEE Computer Graphics and Applications*, pages 38–46, November - December 2005.

[GF03] Richard J. Glassey and Robert I. Ferguson. Space Semantics: An architecture for modeling environments. In *Proceedings of the Workshop on Location-Aware Computing (UbiComp 2003)*, pages 19–21, Seattle, WA, USA, October 12 2003.

[GH07] Nachi Gupta and Raphael Hauser. Kalman filtering with equality and inequality state constraints. Technical report, Oxford University Computing Laboratory, Oxford, United Kingdom, August 2007.

[Gha98] Zoubin Ghahramani. *Adaptive Processing of Sequences and Data Structures. Lecture Notes in Artificial Intelligence*, volume 1387, chapter Learning Dynamic Bayesian Networks, pages 168–197. Springer Verlag, Berlin, 1998.

[Gha01] Zoubin Ghahramani. An introduction to Hidden Markov Models and Bayesian networks. *International Journal of Pattern Recognition and Artificial Intelligence*, 15(1):9–42, 2001.

[HAZ+04] M. Hasanuzzaman, V. Ampornaramveth, Tao Zhang, M.A. Bhuiyan, Y. Shirai, and H. Ueno. Real-time vision-based gesture recognition for human robot interaction. In *Proceedings of the IEEE International Conference on Robotics and Biomimetics (ROBIO 2004)*, pages 413–418, Shenyang, China, August 22-26 2004.

[HB01] Jeffrey Hightower and Gaetano Borriello. Location systems for ubiquitous computing. *IEEE Computer*, 34(8):57–66, August 2001.

[HH06] Mike Hazas and Andy Hopper. Broadband ultrasonic location systems for improved indoor positioning. *IEEE Transactions on Mobile Computing*, 5(5):536–547, May 2006.

[HHS+02] Andy Harter, Andy Hopper, Pete Steggles, Andy Ward, and Paul Webster. The anatomy of a context-aware application. *Wireless Networks*, 8(2-3):187–197, March 2002.

[Hig03] Jeffrey Hightower. From position to place. In *Proceedings of the Workshop on Location-Aware Computing (UbiComp 2003)*, Seattle, WA, USA, October 12 2003.

[HKG+05] Mike Hazas, Christian Kray, Hans Gellersen, Henoc Agbota, Gerd Kortuem, and Albert Krohn. A relative positioning system for co-located mobile devices. In *Proceedings of the 3rd International Conference on Mobile systems, applications, and services (MobiSys 2005)*, pages 177–190, Seattle, Washington, USA, June 6-8 2005.

[HS95] Eric Horvitz and Michael Shwe. Handsfree decision support: Toward a non-invasive human-computer interface. In *Proceedings of the 19th Annual Symposium on Computer Applications in Medical Care: Toward Cost-Effective Clinical Computing (SCAMC 1995)*, New Orleans, LA, USA, October 28 - November 1 1995.

[HVBW01] Jeffrey Hightower, Chris Vakili, Gaetano Borriello, and Roy Want. Design and calibration of the spotON ad-hoc location sensing system. Technical report, University of Washington, Department of Computer Science and Engineering, Seattle, WA, USA, August 2001.

[HW02] Mike Hazas and Andy Ward. A novel broadband ultrasonic location system. In *Proceedings of the 4th International Conference on Ubiquitous Computing (UbiComp 2002)*, pages 299–305, Göteborg, Sweden, September 29 - October 1 2002.

[HWB00] Jeffrey Hightower, Roy Want, and Gaetano Borriello. SpotON: An indoor 3D location sensing technology based on RF signal strength. Technical report, University of Washington, Department of Computer Science and Engineering, Seattle, WA, USA, February 2000.

[HWL+03] Sumi Helal, Bryon Winkler, Choonhwa Lee, Youssef Kaddoura, Lisa Ran, Carlos Giraldo, Sree Kuchibhotla, and William Mann. Enabling location-aware pervasive computing applications for the edlerly. In *Proceedings of the 1st IEEE International Conference on Pervasive Computing and Communications (PerCom 2003)*, pages 531–536, Dallas-Fort Worth, TX, USA, March 23-26 2003.

[Int] Interlink Electronics. Force sensing resistor specifications. http://www.interlinkelectronics.com/force_sensors/products/forcesensingresistors/standardsensors.html.

[JCH+04] Xiaodong Jiang, Nicholas Y. Chen, Jason I. Hong, Kevin Wang, Leila Takayama, and James A. Landay. Siren: Context-aware computing for firefighting. In *Proceedings of the 2nd International Conference on Pervasive Computing (Pervasive 2004)*, pages 87–105, Linz / Vienna, Austria, April 21-23 2004.

[Jim08] Borja Jiménez Salmerón. Modeling of mobile end-user context. Master's thesis, Helsinki University of Technology, Department of Electrical and Communications Engineering, Madrid, Spain, May 2008.

[JKS98] Kang-Hyun Jo, Yoshinori Kuno, and Yoshiaki Shirai. Manipulative hand gesture recognition using task knowledge for human computer interaction. In *Proceedings of the 3rd IEEE International Conference on Automatic Face and Gesture Recognition (FG 2008)*, pages 468–473, Nara, Japan, April 14-16 1998.

[JLP06] Guang-yao Jin, Xiao-yi Lu, and Myong-Soon Park. An indoor localization mechanism using active RFID tag. In *Proceedings of the IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC 2006)*, pages 40–43, Taichung, Taiwan, June 5-7 2006.

[JLT04a] Holger Junker, Paul Lukowicz, and Gerhard Tröster. Continuous recognition of arm activities with wearable motion sensors. In *Proceedings of the 8th IEEE International Symposium on Wearable Computers (ISWC'04)*, Arlington, VA, USA, October 31 - November 3 2004.

[JLT04b] Holger Junker, Paul Lukowicz, and Gerhard Tröster. Locomotion analysis using a simple feature derived from force sensing resistors. In *Proceedings of the 2nd International Conference on Biomedical Engineering*, Innsbruck, Austria, February 16-18 2004.

[JS04] Isaacs Jason and Foo Simon. Optimized wavelet hand pose estimation for american sign language recognition. In *Proceedings of the Congress on Evolutionary Computation (CEC 2004)*, pages 797– 802, Portland, OR, USA, June 19-23 2004.

[Jun05] Holger Junker. *Human activity recognition and gesture spotting with body-worn sensors.* PhD thesis, ETH Zürich, Zürich, Switzerland, 2005.

[KBD01] Ludmila I. Kuncheva, James C. Bezdek, and Robert P.W. Duin. Decision templates for multiple classifier fusion: An experimental comparison. *Pattern Recognition*, 34(2):299–314, February 2001.

[Kla07] Markus Klann. Playing with fire: User-centered design of wearable computing for emergency response. In *Proceedings of the 1st International Workshop on Mobile Information Technology for Emergency Response (Mobile Response 2007)*, pages 116–125, Sankt Augustin, Germany, February 22-23 2007.

[KOL08] Matthias Kreil, Georg Ogris, and Paul Lukowicz. Muscle activity evaluation using force sensitive sensors. In *Proceedings of the 5th International Workshop on Wearable and Implantable Body Sensor Networks (BSN 2008)*, Hong Kong, China, June 1-3 2008.

[KRG+07] Markus Klann, Till Riedel, Hans Gellersen, Carl Fischer, Matt Oppenheim, Paul Lukowicz, Gerald Pirkl, Kai Kunze, Monty Beuster, Michael Beigl, Otto Visser, and Mirco Gerling. Lifenet: An ad-hoc sensor network and wearable system to provide firefighters with navigation support. In *Proceedings of the 9th International Conference on Ubiquitous Computing (UbiComp 2007)*, pages 116–125, Innsbruck, Austria, September 16-19 2007.

[Kru03] John Krumm. Probabilistic inferencing for location. In *Proceedings of the Workshop on Location-Aware Computing (UbiComp 2003)*, pages 25–27, Seattle, WA, USA, October 12 2003.

[KSS03] Nicky Kern, Bernt Schiele, and Albrecht Schmidt. Multi-sensor activity context detection for wearable computing. In *Proceedings of the 1st European Symposium on Ambient Intelligence (EUSAI 2003)*, pages 220–232, Veldhoven, The Netherlands, November 3-4 2003.

[LABT04] Paul Lukowicz, Oliver Amft, David Bannach, and Gerhard Tröster. Heterogeneous, distributed on body computing in the WearIT@Work project. In *Proceedings of the IEEE International Conference in Mechatronics and Robotics (MechRob 2004)*, Aachen, Germany, September 13-15 2004.

[LC00] Kristof Van Laerhoven and Ozan Cakmakci. What shall we teach our pants? In *Proceedings of the 4th IEEE International Symposium on Wearable Computers (ISWC'00)*, pages 77–83, Atlanta, GA, USA, October 16-17 2000.

[LCB06] Jonathan Lester, Tanzeem Choudhury, and Gaetano Borriello. A practical approach to recognizing physical activities. In *Proceedings of the 4th International Conference on Pervasive Computing (Pervasive 2006)*, pages 1–16, Dublin, Ireland, May 7-10 2006.

[Leo98] Ulf Leonhardt. *Supporting Location-Awareness in Open Distributed Systems*. PhD thesis, Imperial College of Science, Technology and Medicine, University of London, London, UK, May 1998.

[LFK05] Lin Liao, Dieter Fox, and Henry Kautz. Location-based activity recognition using relational Markov networks. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI 2005)*, Edinburgh, Scotland, 30 July - 5 August 2005.

[LHSS06] Paul Lukowicz, Friedrich Hanser, Christoph Szubski, and Wolfgang Schobersberger. Detecting and interpreting muscle activity with wearable force sensors. In *Proceedings of the 4th International Conference on Pervasive Computing (Pervasive 2006)*, pages 101–116, Dublin, Ireland, May 7-10 2006.

[LHW07] Michael Lawo, Otthein Herzog, and Hendrik Witt. An industrial case study on wearable computing applications. In *Proceedings of the 9th ACM International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI 2007)*, pages 448–451, Singapore, September 9-12 2007.

[LK98] Hyeon-Kyu Lee and Jin-Hyung Kim. Gesture spotting from continuous hand motion. *Pattern Recognition Letters*, 19(5-6):513–520, April 1998.

[LKLC03] Jessica Lin, Eamonn Keogh, Stefano Lonardi, and Bill Chiu. A symbolic representation of time series, with implications for streaming algorithms. In *Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery (DMKD 2003)*, San Diego, CA, USA, June 13 2003.

[LKT04] Paul Lukowicz, Tünde Kirstein, and Gerhard Tröster. Wearable systems for health care applications. *Methods of Information in Medicine*, 43(3):232–230, 2004.

[LM02] Seon-Woo Lee and Kenji Mase. Activity and location recognition using wearable sensors. *IEEE Pervasive Computing*, 1(3):24–32, July 2002.

[LTGLH07] Paul Lukowicz, Andreas Timm-Giel, Michael Lawo, and Otthein Herzog. WearIT@work: Toward real-world industrial wearable computing. *IEEE Pervasive Computing*, 6(4):8–13, October - December 2007.

[LWJ+04] 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 *Proceedings of the 2nd International Conference on Pervasive Computing (Pervasive 2004)*, pages 18–32, Linz/Vienna, Austria, April 18-23 2004.

[LWL07] Jessica Lin, Eamonn Keogh Li Wei, and Stefano Lonardi. Experiencing SAX: A novel symbolic representation of time series. *Data Mining and Knowledge Discovery Journal*, 15(2):107–144, October 2007.

[LX96] Christopher Lee and Yangsheng Xu. Online, interactive learning of gestures for human/robot interfaces. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 1996)*, pages 2982–2987, Minneapolis, MN, USA, April 22-28 1996.

[Man97] Steve Mann. Wearable computing: A first step toward personal imaging. *Computer*, 30(2):25–32, February 1997.

[Mau07] Iñaki Maurtua. Wearable technology as key enabling technology for user empowerment. In *Proceedings of the 9th European Conference for the Advancement of Assistive Technology (AAATE 2007)*, San Sebastian, Spain, October 3-5 2007.

[Men03] Diego R. Mendoza. Using ontologies in context-aware services platforms. Master's thesis, University of Twente, Enschede, The Netherlands, November 2003.

[MHS01] Jani Mäntyjärvi, Johan Himberg, and Tapio Seppänen. Recognizing human motion with multiple acceleration sensors. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC 2001)*, pages 747–752, Tucson, AZ, USA, October 07-10 2001.

[MIES07] David Minnen, Charles Isbell, Irfan Essa, and Thad Starner. Detecting subdimensional motifs: An efficient algorithm for generalized multivariate pattern discovery. In *Proceedings of the 7th IEEE International Conference on Data Mining (ICDM 2007)*, pages 601–606, Omaha, NE, USA, October 28-31 2007.

[MLT06] Jan Meyer, Paul Lukowicz, and Gerhard Tröster. Textile pressure sensor for muscle activity and motion detection. In *Proceedings of the 10th IEEE International Symposium on Wearable Computers (ISWC'06)*, pages 69–72, Montreux, Switzerland, October 11-14 2006.

[MRC05] Meinard Müller, Tido Röder, and Michael Clausen. Efficient content-based retrieval of motion capture data. In *Proceedings of the 35th International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH 2005)*, pages 677–685, Los Angeles, CA, USA, July 31 - August 4 2005.

[NDA04] Chahe Nerguizian, Charles Despins, and Sofiene Affes. Indoor geolocation with received signal strength fingerprinting technique and neural networks. In *Proceedings of the 11th International Conference on Telecommunications (ICT 2004)*, pages 866–875, Fortaleza, Brazil, August 1-6 2004.

[NDA06] Chahe Nerguizian, Charles Despins, and Sofiene Affes. Geolocation in mines with an impulse response fingerprinting technique and neural networks. *IEEE Transactions on Wireless Communications*, 5(3):603–611, March 2006.

[NLLP03] Lionel M. Ni, Yunhao Liu, Yiu Cho Lau, and Abhishek P. Patil. Landmarc: Indoor location sensing using active RFID. In *Proceedings of the 1st IEEE International Conference on Pervasive Computing and Communications (PerCom 2003)*, pages 407–415, Dallas-Fort Worth, TX, USA, March 23-26 2003.

[NR02] Gonzalo Navarro and Mathieu Raffinot. *Flexible pattern matching in strings: Practical on-line search algorithms for texts and biological sequences.* Cambridge University Press, 2002.

[NSKW05] Tom Nicolai, Thomas Sindt, Holger Kenn, and Hendrik Witt. Case study of wearable computing for aircraft maintenance. In *Proceedings of the 2nd International Forum on Applied Wearable Computing (IFAWC 2005)*, pages 97–110, Zürich, Switzerland, March 17-18 2005.

[NSP02] James Nord, Kare Synnes, and Peter Parnes. An architecture for location aware applications. In *Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS 2002)*, page 293, Waikoloa Village, HI, USA, January 07 - 10 2002.

[NSSKAA06] Joe Naoum-Sawaya, Mazen Slim, Sami Khawam, and Mohamad Adnan Al-Alaoui. Dynamic system design for american sign language recognition. In *Proceedings of the 2nd International Symposium on Communications, Control, and Signal Processing (ISCCSP 2006)*, Marakesh, Morocco, March 13-15 2006.

[OKL07] Georg Ogris, Matthias Kreil, and Paul Lukowicz. Using FSR based muscle activity monitoring to recognize manipulative arm gestures. In *Proceedings of the 11th IEEE International Symposium on Wearable Computers (ISWC'07)*, pages 45–48, Boston, MA, USA, October 11-13 2007.

[OLST] Georg Ogris, Paul Lukowicz, Thomas Stiefmeier, and Gerhard Tröster. Continuous recognition of manipulative hand gestures. *Pattern Analysis and Applications.* prepared for submission.

[OSJ+05] Georg Ogris, Thomas Stiefmeier, Holger Junker, Paul Lukowicz, and Gerhard Tröster. Using ultrasonic hand tracking to augment motion analysis based recognition of manipulative gestures. In *Proceedings of the 9th IEEE International Symposium on Wearable Computers (ISWC'05)*, pages 152–159, Osaka, Japan, October 18-21 2005.

[OSLT08] Georg Ogris, Thomas Stiefmeier, Paul Lukowicz, and Gerhard Tröster. Using a complex multi-modal on-body sensor system for activity spotting. In *Proceedings of the 12th IEEE International Symposium on Wearable Computers (ISWC'08)*, pages 55–62, Pittsburgh, PA, USA, September 28 - October 1 2008.

[Par03] Rita Paradiso. Wearable health care system for vital signs monitoring. In *Proceedings of the 4th International IEEE EMBS Special Topic Conference*

on Information Technology Applications in Biomedicine (ITAB 2003), pages 283–286, Birmingham, UK, 24-26 April 2003.

[PCB00] Nissanka B. Priyantha, Anit Chakraborty, and Hari Balakrishnan. The Cricket location-support system. In *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MobiCom 2000)*, pages 32–43, Boston, MA, USA, August 6-11 2000.

[PFKP05] Donald J. Patterson, Dieter Fox, Henry Kautz, and Matthai Philipose. Fine-grained activity recognition by aggregating abstract object usage. In *Proceedings of the 9th IEEE International Symposium on Wearable Computers (ISWC'05)*, pages 44–51, Osaka, Japan, October 18-21 2005.

[PMBT01] Nissanka B. Priyantha, Allen K. L. Miu, Hari Balakrishnan, and Seth J. Teller. The Cricket compass for context-aware mobile applications. In *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking (MobiCom 2001)*, pages 1–14, Rome, Italy, July 16-21 2001.

[Pri05] Nissanka Bodhi Priyantha. *The Cricket Indoor Location System*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, June 2005.

[PTA06] Shwetak N. Patel, Khai N. Truong, and Gregory D. Abowd. Powerline positioning: A practical sub-room-level indoor location system for domestic use. In *Proceedings of the 8th International Conference on Ubiquitous Computing (UbiComp 2006)*, pages 441–458, Orange County, CA, USA, September 17-21 2006.

[PWP+07] Lucas Paletta, Roland Wack, Gerhard Paar, Georg Ogris, and Christophe le Gal. Appearance based positioning in urban environments using Kalman filtering. *Advances in Mobile Mapping Technology, ISPRS Book Series, Taylor & Francis, Leiden, Netherlands*, pages 79–88, 2007.

[Rab89] Lawrence R. Rabiner. A tutorial in Hidden Markov Models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, February 1989.

[Ran05] Cliff Randell. Wearable computing: A review. Technical report, University of Bristol, Bristol, UK, 2005.

[RAT04] Julian Randall, Oliver Amft, and Gerhard Tröster. Location by solar cells: An experiment plan. In *Proceedings of the 8th Student IEEE International Symposium on Wearable Computers (ISWC'04)*, pages 50–55, Arlington, VA, USA, October 31 - November 3 2004.

[RDM03] Cliff Randell, Chris Djiallis, and Henk Muller. Personal position measurement using dead reckoning. In *Proceedings of the 7th IEEE International Symposium on Wearable Computers (ISWC'03)*, pages 166–173, White Plains, NY, USA, October 21-23 2003.

[Rek01] Jun Rekimoto. GestureWrist and GesturePad: Unobtrusive wearable interaction devices. In *Proceedings of the 5th IEEE International Symposium on Wearable Computers (ISWC'01)*, pages 21–27, Zürich, Switzerland, October 7-9 2001.

[RM00] Cliff Randell and Henk Muller. Context awareness by analysing accelerometer data. In *Proceedings of the 4th IEEE International Symposium on Wearable Computers (ISWC'00)*, pages 175–176, Atlanta, GA, USA, October 16-17 2000.

[Roe06] Daniel Roetenberg. *Inertial and Magnetic Sensing of Human Motion.* PhD thesis, University of Twente, Enschede, The Netherlands, January 2006.

[RV05] Daniel Roetenberg and Peter H. Veltink. Camera-marker and inertial sensor fusion for improved motion tracking. In *Proceedings of the European Society of Movement Analysis for Adults and Children (ESMAC 2005)*, pages 51–52, Barcelona, Spain, September 22-24 2005.

[SAW94] Bill N. Schilit, Norman Adams, and Roy Want. Context-aware computing applications. In *Proceedings of the IEEE Workshop on Mobile Computing Systems and Applications (WMCSA 1994)*, pages 89–101, Santa Cruz, CA, USA, December 8-9 1994.

[SBGP04] Adam Smith, Hari Balakrishnan, Michel Goraczko, and Nissanka Priyantha. Tracking moving devices with the Cricket location system. In *Proceedings of the 2nd International Conference on Mobile systems, applications, and services (MobiSys 2004)*, pages 190–202, Boston, MA, USA, June 6-9 2004.

[Sch02] Albrecht Schmidt. *Ubiquitous Computing - Computing in Context.* PhD thesis, Lancaster University, Computing Department, Lancaster, UK, November 2002.

[SHvLS08] Maja Stikic, Tam Huynh, Kristoph van Laerhoven, and Bernt Schiele. ADL recognition based on the combination of RFID and accelerometer sensing. In *Proceedings of the 2nd International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth 2008)*, pages 258–263, Tampere, Finland, January 30 - February 1 2008.

[Sie01] Dan Siewiorek. Wearable computer architecture and applications. Presentation at Coordinated Science Laboratory (CSL), University of Illinois, March 20 2001.

[SLP+03] Mathias Stäger, Paul Lukowicz, Niroshan Perera, Thomas von Büren, Gerhard Tröster, and Thad Starner. SoundButton: Design of a low power wearable audio classification system. In *Proceedings of the 7th IEEE International Symposium on Wearable Computers (ISWC'03)*, pages 12–17, White Plains, NY, USA, October 21-23 2003.

[SLR+06] Thomas Stiefmeier, Clemens Lombriser, David Roggen, Holger Junker, Georg Ogris, and Gerhard Tröster. Event-based activity tracking in work environments. In *Proceedings of the 3rd International Forum on Applied Wearable Computing (IFAWC 2006)*, pages 91–100, Bremen, Germany, March 15-16 2006.

[SMLG04] Moritz Störring, Thomas B. Moeslund, Yong Liu, and Erik Granum. Computer vision-based gesture recognition for an augmented reality interface. In *Proceedings of the 4th IASTED International Conference on Visualization, Imaging, and Image Processing (VIIP 2004)*, pages 766–771, Marbella, Spain, September 6-8 2004.

[SOJ+06] 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. In *Proceedings of the 10th IEEE International Symposium on Wearable Computers (ISWC'06)*, pages 97–104, Montreux, Switzerland, October 11-14 2006.

[SRO+08] Thomas Stiefmeier, Daniel Roggen, Georg Ogris, Paul Lukowicz, and Gerhard Tröster. Wearable activity tracking in car manufacturing. *IEEE Pervasive Computing: Special issue on Activity-Based Computing*, 7(2):42–50, April-June 2008.

[SRT07a] Thomas Stiefmeier, Daniel Roggen, and Gerhard Tröster. Fusion of string-matched templates for continuous activity recognition. In *Proceedings of the 11th IEEE International Symposium on Wearable Computers (ISWC'07)*, pages 41–44, Boston, MA, USA, October 11-13 2007.

[SRT07b] Thomas Stiefmeier, Daniel Roggen, and Gerhard Tröster. Gestures are strings: Efficient online gesture spotting and classification using string matching. In *Proceedings of the 2nd International Conference on Body Area Networks (BodyNets 2007)*, Florence, Italy, June 11-13 2007.

[SSP98] Thad Starner, Bernt Schiele, and Alex Pentland. Visual contextual awareness in wearable computing. In *Proceedings of the 2nd International Symposium on Wearable Computers (ISWC'98)*, pages 50–57, Pittsburgh, PA, USA, October 19-20 1998.

[SSS08] Dan Siewiorek, Asim Smailagic, and Thad Starner. *Application Design for Wearable Computing*. Synthesis Lectures on Mobile and Pervasive Computing. Morgan and Claypool Publishers, 2008.

[ST94] Bill N. Schilit and Marvin M. Theimer. Disseminating active map information to mobile hosts. *IEEE Network*, 8(5):22–32, September - October 1994.

[Sti08] Thomas Stiefmeier. *Real-Time spotting of human activities in industrial environments*. PhD thesis, ETH Zürich, Zürich, Switzerland, July 2008.

[SWP98] Thad Starner, Joshua Weaver, and Alex Pentland. Real-time American sign language recognition using desk and wearable computer based video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(12):1371–1375, December 1998.

[Tau02] Joshua A. Tauber. Indoor location systems for pervasive computing. Technical report, MIT Laboratory for Computer Science, Cambridge, MA, USA, August 2002.

[Ten00] David Tennenhouse. Proactive computing. *Communications of the ACM*, 43(5):43–50, May 2000.

[TMK04] Georgios Theocharous, Kevin Murphy, and Leslie Kaelbling. Representing hierarchical POMDPs as DBNs for multi-scale robot localization. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2004)*, pages 1045–1051, New Orleans, LA, USA, April 26 - May 1 2004.

[UDL07] Sridhar Ungarala, Eric Dolence, and Keyu Li. Constrained extended Kalman filter for nonlinear state estimation. In *Proceedings of the 8th International IFAC Symposium on Dynamics and Control of Process Systems (DYCOPS 2007)*, pages 63–68, Cancún, Mexico, June 6-8 2007.

[VM98] Christian Vogler and Dimitris N. Metaxas. ASL recognition based on a coupling between HMMs and 3D motion analysis. In *Proceedings of the 6th International Conference on Computer Vision (ICCV 1998)*, pages 363–369, Bombay, India, January 4-7 1998.

[War98] Andy Ward. *Sensor-driven Computing*. PhD thesis, University of Cambridge, Cambridge, UK, August 1998.

[War06] Jamie A. Ward. *Activity Monitoring: Continuous Recognition and Performance Evaluation*. PhD thesis, ETH Zürich, Zürich, Switzerland, February 2006.

[Wei99] Mark Weiser. The computer for the 21st century. *ACM SIGMOBILE Mobile Computing and Communications Review archive: Special issue dedicated to Mark Weiser*, 3(3):3–11, July 1999.

[WH08] Oliver Woodman and Robert Harle. Pedestrian localisation for indoor environments. In *Proceedings of the 10th International Conference on Ubiquitous Computing (UbiComp 2008)*, pages 114–123, Seoul, Korea, September 21-24 2008.

[Wit07] Hendrik Witt. Evaluation of five interruption methods for speech interaction in wearable computing dual-task environments. In *Proceedings of the 11th IEEE International Symposium on Wearable Computers (ISWC'07)*, pages 63–66, Boston, MA, USA, October 11-13 2007.

[WJH97] Andy Ward, Alan Jones, and Andy Hopper. A new location technique for the active office. *IEEE Personal Communications*, 4(5):42–47, October 1997.

[WLKK06] Hendrik Witt, Rüdiger Leibrandt, Andreas Kemnade, and Holger Kenn. SCIPIO: A miniaturized building block for wearable interaction devices. In *Proceedings of the 3rd International Forum on Applied Wearable Computing (IFAWC 2006)*, pages 103–108, Bremen, Germany, March 15-16 2006.

[WLT06] Jamie A. Ward, Paul Lukowicz, and Gerhard Tröster. Evaluating performance in continuous context recognition using event-driven error characterisation. In *Proceedings of the 2nd International Workshop on Location- and Context-Awareness (LoCA 2006)*, pages 239–255, Dublin, Ireland, May 10-11 2006.

[WLTS06] Jamie A. Ward, Paul Lukowicz, Gerhard Tröster, and Thad E. Starner. Activity recognition of assembly tasks using body-worn microphones and accelerometers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(10):1553–1567, October 2006.

[WNK06] Hendrik Witt, Tom Nicolai, and Holger Kenn. Designing a wearable user interface for hands-free interaction in maintenance applications. In *Proceedings of the 4th annual IEEE international conference on Pervasive Computing and Communications Workshops (Percomw 2006)*, pages 4–7, Pisa, Italy, March 13-17 2006.

[WPVS05] Gertjan Wijnalda, Steffen Pauws, Fabio Vignoli, and Heiner Stuckenschmidt. A personalized music system for motivation in sport performance. *IEEE Pervasive Computing*, 4(3):26–32, July 2005.

[YJ07] Jane Yau and Mike Joy. A context-aware and adaptive learning schedule framework for supporting learners' daily routines. In *Proceedings of the Mobile Communications and Learning Workshop (MCL 2007), as part of the IEEE ICONS Conference (ICONS 2007)*, pages 31–37, Sainte-Luce, France, April 22-28 2007.

[YOI92] Junji Yamato, Jun Ohya, and Kenichiro Ishii. Recognizing human action in time-sequential images using Hidden Markov Models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 1992)*, pages 379–385, Champaign, IL, USA, June 15-18 1992.

[YSBY01] Ho-Sub Yoon, Jung Soh, Younglae J. Bae, and Hyun Seung Yang. Hand gesture recognition using combined features of location, angle and velocity. *Pattern Recognition*, 34(7):1491–1501, March 2001.

[ZLS07] Andreas Zinnen, Kristof Van Laerhoven, and Bernt Schiele. Toward recognition of short and non-repetitive activities from wearable sensors. In *Proceedings of the European Conference on Ambient Intelligence (AMI 2007)*, pages 142–158, Darmstadt, Germany, November 7-10 2007.

[ZWS09] Andreas Zinnen, Christian Wojek, and Bernt Schiele. Multi activity recognition based on bodymodel-derived primitives. In *4th International Symposium on Location and Context Awareness (LoCA 2009)*, Tokyo, Japan, May 7-8 2009.

Abbreviations

1-m-s	.....	1-component Gaussian mixture distribution supervised location modeling approach
-A-		
acEET	.....	accumulated class-wise event error table
AD converter	.....	analog-to-digital converter
almighty	.....	artificial classifier fusion method: either one classifier result is correct
s-m-m	.....	semi-supervised Gaussian mixture location modeling
aob	.....	classifier fusion method based on average ranking of winner classes
avg	.....	classifier fusion method based on average class ranking
-B-		
binary	.....	strict classifier fusion method: both classifiers must agree
BUS	.....	bi-unique segment in the prediction/ground-truth stream
-C-		
C	.....	correctly recognized or spottet event
C4.5	.....	a decision tree classifier
CM	.....	confusion matrix
-D-		
D	.....	deletion error
DC	.....	direct current
-E-		
EET	.....	event error table
EM	.....	expectation maximization
-F-		
F	.....	fragmentation error
FSR	.....	force sensing resistor
-G-		
GNSS	.....	global navigation satellite system
GPS	.....	global positioning system

GSM	.....	global system for mobile communications
-H-		
HCI	.....	human-computer interaction
HMM	.....	Hidden Markov Model
-I-		
I	.....	insertion error
I^2C	.....	inter-integrated circuit serial bus
ICA	.....	independent component analysis
ID	.....	unique identifier
IMU	.....	inertial measurement unit
IR	.....	infrared
-K-		
k-N-N	.....	k-nearest-neighbor (decision tree) classifier
-L-		
LDA	.....	least discriminant analysis
LSQ	.....	least squares optimization
-M-		
M	.....	merge error
-N-		
nG	.....	ground truth event was not assigned an event error label
n-m-s	.....	n-component Gaussian mixture distribution supervised location modeling approach
nP	.....	prediction event was not assigned an event error label
NULL	.....	undefined class or event, i.e. a gesture or activity class or instance of this class (event) of unknown or undefined type.
-O-		
O	.....	timing error (overfill)
op-amp	.....	operational amplifier
-P-		
pa	.....	classifier fusion method based on location plausibility analysis
PCA	.....	principle components analysis

p-m	.....	pseudo-manual location modeling
PR	.....	precision and recall

-R-

red	.....	reduced gesture class resolution
RF	.....	radio-frequency
RFID	.....	radio-frequency identification
s-m-r	.....	semi-supervised Gaussian mixture location modeling using the reduced gesture class resolution
RMS	.....	root mean square
ROC	.....	receiver-operator characteristics
RSSI	.....	received signal strength indication

-S-

SAX	.....	symbolic aggregate approximation
SET	.....	segment error table
SPI	.....	serial peripheral interface bus
SPP	.....	serial port profile
SWAB	.....	sliding window and bottom-up algorithm

-U-

U	.....	timing error (underfill)
UART	.....	universal asynchronous receiver/transmitter
UPS	.....	ultrasonic positioning system
USB	.....	universal serial bus
UWB	.....	ultra-wide-band

List of figures

List of tables