
Early Abstraction of Inertial Sensor Data for Long-Term Deployments

Frühe Abstraktion von Inertialsensordaten für Langzeit-Anwendungen

Vom Fachbereich Informatik der Technischen Universität Darmstadt
zur Erlangung des akademischen Grades Doktor-Ingenieur (Dr.-Ing.)
genehmigte Dissertation von M.Sc. Eugen Berlin aus Sankt Petersburg
Darmstadt 2014 – D 17

1. Gutachten: Dr. Kristof Van Laerhoven

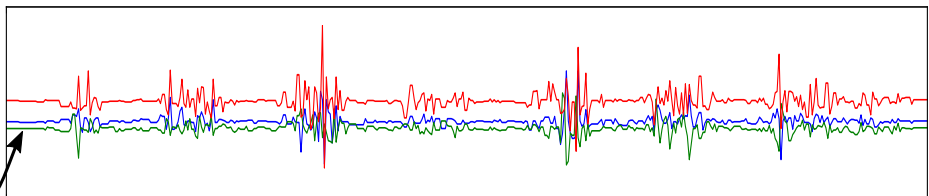
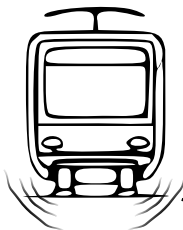
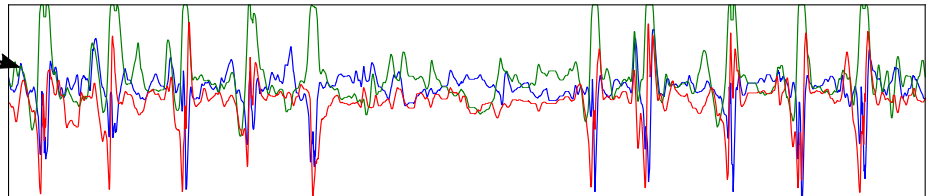
2. Gutachten: Prof. Dr. Oliver Amft

Tag der Einreichung: 27.08.2014

Tag der mündlichen Prüfung: 30.10.2014



TECHNISCHE
UNIVERSITÄT
DARMSTADT



Bitte zitieren Sie dieses Dokument als:

URN: urn:nbn:de:tuda-tuprints-42250

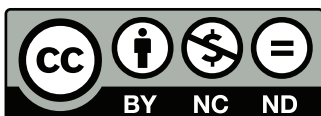
URL: <http://tuprints.ulb.tu-darmstadt.de/4225>

Dieses Dokument wird bereitgestellt von tuprints,

E-Publishing-Service der TU Darmstadt

<http://tuprints.ulb.tu-darmstadt.de>

tuprints@ulb.tu-darmstadt.de



Die Veröffentlichung steht unter folgender Creative Commons Lizenz:

Namensnennung – Keine kommerzielle Nutzung – Keine Bearbeitung 3.0 Deutschland

<http://creativecommons.org/licenses/by-nc-nd/3.0/de/>

Abstract

Advances in microelectronics over the last decades have led to miniaturization of computing devices and sensors. A driving force to use these in various application scenarios is the desire to grasp physical phenomena from the environment, objects and living entities. We investigate sensing in two particularly challenging applications: one where small sensor modules are worn by people to detect their activities, and one where wirelessly networked sensors observe events over an area.

This thesis takes a data-driven approach, focusing on human motion and vibrations caused by trains that are captured by accelerometer sensors as time series and shall be analyzed for characteristic patterns. For both, the acceleration sensor must be sampled at relatively high rates in order to capture the essence of the phenomena, and remain active for long stretches of time. The large amounts of gathered sensor data demand novel approaches that are able to swiftly process the data while guaranteeing accurate classification results. The following contributions are made in particular:

- A data logger that would suit the requirements of long-term deployments is designed and evaluated. In a power profiling study both hardware components and firmware parameters are thoroughly tested, revealing that the sensor is able to log acceleration data at a sampling rate of 100 Hertz for up to 14 full days on a single battery charge.
- A technique is proposed that swiftly and accurately abstracts an original signal with a set of linear segments, thus preserving its shape, while being twice as fast as a similar method. This allows for more efficient pattern matching, since for each pattern only a fraction of data points must be considered. A second study shows that this algorithm can perform data abstraction directly on a data logger with limited resources.
- The railway monitoring scenario requires streaming vibration data to be analyzed for particular sparse and complex events directly on the sensor node, extracting relevant information such as train type or length from the shape of the vibration footprint. In a study conducted on real-world data, a set of efficient shape features is identified that facilitates train type prediction and length estimation with very high accuracies.
- To achieve fast and accurate activity recognition for long-term bipolar patients monitoring scenarios, we present an approach that relies on the salience of motion patterns (motifs) that are characteristic for the target activity. These motifs are accumulated by using a symbolic abstraction that encodes the shape of the original signal. A large-scale study shows that a simple bag-of-words classifier trained with extracted motifs is on par with traditional approaches regarding the accuracy, while being much faster.
- Some activities are hard to predict from acceleration data alone with the aforementioned approach. We argue that human-object interactions, captured as human motion and grasped objects through RFID, are an ideal supplement. A custom bracelet-like antenna to detect objects from up to 14 cm is proposed, along with a novel benchmark to evaluate such wearable setups.

By aiming for wearable and wirelessly networked sensor systems, these contributions apply for particularly challenging applications that require long-term deployments of miniature sensors in general. They form the basis of a framework towards efficient event detection that relies heavily on early data abstraction and shape-based features for time series, while focusing less on the classification techniques.



Zusammenfassung

Die Fortschritte in der Mikroelektronik der vergangenen Jahrzehnte führten zur Miniatürisierung von Rechnern und Sensoren. Eine treibende Kraft, diese in verschiedenen Anwendungsszenarien zu verwenden ist der Wunsch nach der Erfassung physikalischer Phänomene in der Umgebung, an Objekten und Lebewesen. Wir untersuchen den Einsatz von Sensoren in zwei besonders anspruchsvolle Anwendungsszenarien: zum Einen, wo kleine Sensormodule von Menschen getragen werden, um ihre Aktivitäten zu erkennen, und zum Anderen, wo drahtlos vernetzte Sensorknoten eine Umgebung nach relevanten Ereignissen überwachen.

Diese Arbeit nimmt einen datenorientierten Ansatz, wobei der Schwerpunkt beim Erfassen menschlicher Bewegungen und von durch Züge verursachten Vibrationen liegt. Diese werden durch Beschleunigungssensoren als Zeitreihen aufgezeichnet und sollen auf charakteristische Muster und Ereignisse untersucht werden. In beiden Szenarien müssen die Beschleunigungssensoren mit relativ hohen Raten abgetastet werden, um die Essenz der Phänomene zu erfassen, was zudem über lange Zeiträume erfolgen muss. Die dabei anfallenden sehr großen Mengen an Sensordaten verlangen nach neuen Ansätze, mit denen gewährleistet sein soll, dass die Daten schnell verarbeitet und gleichzeitig gute Klassifikationsergebnisse bei der Analyse erzielt werden. Die wissenschaftlichen Beiträge dieser Arbeit lassen sich wie folgt zusammenfassen:

- Es wird ein Gerät vorgestellt, welches Beschleunigung und Vibrationen aufzeichnet und den Anforderungen der Langzeiteinsätze entsprechend konstruiert und evaluiert wurde. Eine Untersuchung des Stromverbrauchs bezüglich verschiedener Hardware-Komponenten und Firmware-Parameter hat ergeben, dass der Sensor in der Lage ist Beschleunigungsdaten mit einer Abtastrate von 100 Hertz bis zu 14 Tage lang mit einer einzigen Akkuladung aufzuzeichnen.
- Wir schlagen eine Abstraktionsmethode vor, die das aufgezeichnete Signal effizient und genau durch lineare Segmente abstrahiert, dabei das ursprüngliche Aussehen bewahrt und zudem doppelt so schnell wie ein ähnliches Verfahren ist. Damit wird auch effiziente Mustererkennung ermöglicht, da für jedes Muster nur ein Bruchteil der Datenpunkte zu berücksichtigen ist. Eine zweite Studie zeigt, dass die Methode auf einem Datenlogger mit begrenzten Ressourcen implementiert werden kann.
- Das Szenario zur Schienenverkehrsüberwachung führt die Idee nach früher Datenabstraktion weiter aus, indem es gestreamte Vibrationsdaten auf spärlich vorkommende und komplexe Ereignisse direkt auf dem Sensorknoten untersucht. Damit lassen sich für die Überwachung relevante Informationen, wie Zugtyp oder dessen Länge, aus dem Vibrationsmuster extrahieren. Eine Studie an realen Daten hat eine Reihe von effizienten Features ergeben, auf deren Grundlage der Zugtyp und die Länge mit sehr hohen Genauigkeiten vorhergesagt werden können.
- Um eine schnelle und akkurate Aktivitätserkennung für langfristige Überwachung von bipolaren Patienten zu ermöglichen, stellen wir einen Ansatz vor, der auf der Dichte des Auftretens von Bewegungsmustern (Motif), die charakteristisch für eine Zielaktivität sind, basiert. Die relevanten Motifs werden durch die Verwendung einer

symbolischen Darstellung der Sensordaten, die die Form des ursprünglichen Signals berücksichtigen, akkumuliert. Unsere umfangreiche Studie zeigt, dass ein mit den extrahierten Motifs trainierter einfacher Bag-of-Words Klassifikator ähnliche Erkennungsraten erreicht, wie traditionelle Ansätze auch, und dabei viel schneller ist.

- Einige Aktivitäten sind mit dem oben genannten Ansatz auf der alleinigen Grundlage von Beschleunigungsdaten nur sehr schwer vorherzusagen. Wir argumentieren an dieser Stelle, dass die Interaktion mit Objekten, bestehend aus der Erkennung menschlicher Bewegungen durch Beschleunigungssensoren und der Identifikation von ergriffen Objekten durch RFID, sich ideal ergänzen und die Erkennungsgüte verbessern. Wir stellen eine neu hergestellte RFID Armband-Antenne vor, mit der Objekte bis zu einer Reichweite von 14 cm erkannt werden können. Zudem stellen wir einen neuartigen Vergleichstest vor, mit dem solche tragbare Sensoren ausgewertet und verglichen werden können.

Aufgrund des in dieser Arbeit gewählten Fokus auf tragbare und drahtlos vernetzte Sensor-Systeme, sind die vorgestellten wissenschaftliche Beiträge für eine Vielzahl von besonders anspruchsvollen Anwendungen, die auf langfristigen Einsatz von Sensorknoten abzielen, relevant. Sie bilden eine Grundlage für effiziente Ereigniserkennungssysteme, die insbesondere auf früher Datenabstraktion und der Verwendung von Signalform-Features basieren, und dabei auf weniger komplexe Klassifizierungstechniken zurückgreifen.

Acknowledgments

A dissertation is life in a nutshell. In the beginning there is a spark that triggers the creation of such a life. For me, it were the words: “You’re done with your bachelor thesis. Now, how about doing a PhD?” Well, it required additional two years of Masters studies for the idea to breed inside my head until it finally became live in January 2010. This life now is near its end, but with a prospect of being succeeded by great ‘future work’. During this time, many things have happened and I have learned a lot. The most important thing for me about this life is that I never walked alone. For that I am greatly thankful.

First, I would like to sincerely thank my supervisor, Dr. Kristof Van Laerhoven, for giving me the opportunity to join his research team. His huge knowledge and the inspiring hands-on approach have motivated me to try different things during my research, which influenced the final shape of the thesis. Regular constructive discussions helped a great deal when I was in doubt or got stuck on my journey. I would also like to thank Prof. Dr. Oliver Amft for agreeing to be the co-examiner of my thesis.

Many thanks to my colleagues, Marko Borazio, Philipp Scholl, Pablo Guerrero, Manuel Dietrich, Agha Muhammad, Sofia Nikitaki, Christian Seeger, Iliya Gurov and François Philipp. I highly appreciate the great research environment that enabled lots of productive discussions and collaborations, as well as the fun we had together on many occasions. A special mention for Marko, with whom I shared the office during these years, for the countless discussions regarding research and life in general, as well as his balanced and relaxed way. I am particularly grateful to Ursula Paeckel. Without her taking care of the group’s office business and her generous advice in any aspect this life would have been agonizingly painful.

I would like to acknowledge the numerous study participants for wearing the data logger thus providing the huge data base for this thesis. I highly appreciate the help and expertise of Dr. Oliver Soffke for tuning the new RFID antenna prototype; Prof. Dr. Jan Peters for granting access to the BioRob robot and Hany Abdulsamad for programming it for our power-profiling study; Hans Gabler, Hubert Schmühl and Harald Behr from the Deutsche Bahn Netz AG for cooperation leading to the deployment of sensors at the railway tracks near Cologne.

Hereby I would like to extend my gratitude to my friend Jan Rehker for proof-reading the thesis and providing precise, tartly, and very valuable feedback during hour-long discussions.

On a more personal note, my biggest debt of gratitude goes to my family for their help and support. Words cannot describe how eternally grateful I am to my parents and my sister Vera for their unconditional love, advice and patience. My beloved wife Julia, for sharing our lives and supporting me, whatever will come. My daughter Nadja, who was born during my PhD, for motivating and keeping me awake during the writing process, and always putting a smile on my face.



Contents

1	Introduction	1
1.1	Problem statement	3
1.2	Challenges for Activity Recognition from Long-term Inertial Data	5
1.3	Challenges for on-line vibration event classification in WSNs	6
1.4	Challenges in hardware design for long-term deployments	6
1.5	Contributions	7
1.5.1	Applications' demand for a custom hardware platform	7
1.5.2	Efficient on-line data abstraction technique	8
1.5.3	Efficient shape features for WSNs in railway monitoring scenarios	9
1.5.4	Long-term activity recognition with Dense Motifs	9
1.5.5	Detecting interactions with objects	10
1.5.6	Contributions summary	11
1.6	Outline	11
2	Related Work	15
2.1	Abstracting Sensor Data While Preserving Its Shape	15
2.1.1	Binary Encoding	16
2.1.2	Mean and Variance	16
2.1.3	Discrete Fourier Transform	18
2.1.4	Piecewise Defined Functions	19
2.1.5	Symbolic Representation	21
2.2	Embedding Abstraction Algorithms Directly on a Sensor Node	24
2.3	Human Activity Recognition Applications	25
2.3.1	Actigraphy	25
2.3.2	Human Motion Analysis in the Medical Domain	27
2.3.3	Activity Recognition for Elderly Care	28
2.3.4	Fitness Applications	29
2.4	Wireless Sensor Networks Applications	30
2.5	Summary	31
3	Designing an Efficient Activity Logger	33
3.1	Motivation	33
3.2	Wrist-worn Unit Design	34
3.3	Experimental Setup	35
3.3.1	Measuring Current Consumption	35
3.3.2	The Robot Arm Benchmark	36
3.3.3	Long-term Current Consumption Measurement	37
3.4	Evaluation	38
3.4.1	Which Component Should Control the Sampling?	39

3.4.2	What a Difference an SD Card Makes	40
3.4.3	The (Battery) Cost of an OLED Display	42
3.4.4	The (Obvious) Impact of Reduced Sampling Frequency	43
3.4.5	The 10-day Deployment Test	44
3.5	Conclusions	45
4	Efficient Data Abstraction with Linear Segments	47
4.1	Motivation	47
4.2	Application Scenario: Interactive Analysis of Human Motion Data	48
4.3	Approximation of Accelerometer Data	50
4.3.1	SWAB	50
4.3.2	mSWAB	52
4.3.3	Evaluating the Approximations	52
4.4	Case Study – Subsequence Matching for Activity Recognition	57
4.4.1	Dynamic Time Warping	58
4.4.2	K Longest Segments	59
4.4.3	Evaluation of Matching Methods	59
4.4.4	Study Conclusions	61
4.5	Case Study – Embedded mSWAB for WSN Applications	62
4.5.1	emSWAB	62
4.5.2	Experiments Methodology	64
4.5.3	Results	67
4.5.4	Study Conclusions	70
4.6	Conclusions	71
5	Complex Event Classification in WSN	73
5.1	Motivation	73
5.2	Encoding Shape of the Signal into Features	76
5.3	Evaluation	78
5.3.1	Trainspotting Data Sets	79
5.3.2	Evaluation Methodology	82
5.3.3	Train Type Classification	84
5.3.4	Train Length Estimation	87
5.4	Discussion and Outlook	89
5.5	Conclusions	90
6	Activity Recognition Through Dense Motif Discovery	93
6.1	Motivation	93
6.2	Case Study – Bipolar Monitoring Scenario	95
6.3	Dense Motif Discovery	96
6.3.1	Method Overview	96
6.3.2	Approximation from Raw Data	97
6.3.3	Mapping to Discrete Symbols	97
6.3.4	Extracting Motifs by Means of Suffix Trees	98
6.3.5	Bag-of-Words Classification	100

6.4	Evaluation	100
6.4.1	Participants and Target Activities	100
6.4.2	Benchmarking the Performance	102
6.4.3	Experiment Results and Discussion	103
6.4.4	Applicability for Other Activities	110
6.5	Conclusions	112
7	Detecting Interactions with Efficient Wrist-worn Sensors	113
7.1	Motivation	113
7.2	Combining RFID reading and Inertial Sensing	114
7.3	Optimizing RFID Reading Range	115
7.4	Optimizing RFID Reading Frequency	118
7.5	Evaluation of RFID Reading: The Box Test	118
7.6	Application Scenarios for Combining RFID and Inertial Sensing	122
7.7	Conclusions	125
8	Conclusion	127
8.1	Summary of Contributions	127
8.2	Conclusions	129
8.3	Outlook	131
	List of Figures	134
	List of Tables	137
	Bibliography	139
	Curriculum Vitæ	149



1 Introduction

Advances in microelectronics and computer science over the past decades have significantly changed our world and how we perceive it. The ongoing computerization has impacted almost all aspects of our lives, including communications, businesses, industry, health, science, or leisure. We are surrounded with a multitude of microelectronic devices that compute, sense and communicate with each other directly or via the Internet, offering their users various types of services. The ongoing miniaturization and diversification of hardware components and their integration in everyday things aids the concept of ubiquitous computing, a paradigm coined by Mark Weiser in late 80ies.

Ubiquitous computing, in all its manifolds, has actively been pursued by thousands of researches, practitioners and businesses from over the world. Pushing beyond general purpose computers as central computing units, the development of new technologies, computing and communication devices and sensors, aimed at obtaining a much more detailed insight on how “things” work. A major focus was put on grasping physical phenomena from the environment as well as objects and living entities within, with application possibilities ranging from the health-care domain over dedicated industry and logistics scenarios up into the everyday’s life of every human being.

For environmental monitoring, sensors can be used that capture relevant conditions, such as temperature, humidity, light intensity, atmospheric pressure, direction and speed of wind, gas concentrations, air or water pollution. For monitoring large spaces or human built infrastructure for relevant physical phenomena, wireless sensor networks have been proposed, consisting of huge numbers of small and relatively inexpensive sensor nodes that are capable of sensing the environment, processing the data and communicating with each other in order to exchange relevant information. Besides already mentioned simple sensor modalities, inertial and vibration data is of particular interest, as it can capture ocean wave fluctuations, land slides and earth-quakes. Sensors that are capable to sense acceleration and vibrations are used to monitor human-built infrastructure, such as bridges, tunnels or buildings, for critical vibration or oscillating motions caused by vehicles or natural phenomena, such as wind.

Monitoring animals or humans is a much more challenging task and requires more and more complex sensors that can capture their location and motion, interactions with objects or other individuals. Various approaches have been proposed over the years that rely on visual data obtained through cameras installed in the environment, location information from different outdoor and indoor localization systems, context information from environmental sensors, or inertial data from body-worn sensors. The diversity of applications was specifically aided by the miniaturization of hardware components, where especially wearable technology profited a lot. A significant impact can be seen in the health-care domain, where the availability of inertial sensors allowed to obtain fine-grained information about the patients’ disease symptoms or conditions. Many researchers focus on elderly care and health applications, where detecting falls, seizures, movement disabilities, regular medication intake, ability of self-contained living are

relevant to the medical staff. The latter has spawned a specific sub-area inside human activity recognition focusing on activities of daily living, including personal care, cooking, eating and drinking, cleaning dishes, or household work.

While many challenges in these research fields have been addressed over the years, and novel hardware and data processing algorithms were proposed, many of the targeted scenarios have specific restrictions that make the use of invasive sensors (such as cameras or microphones) not possible, or can not be tackled even with computing power and resources in reasonable amount of time. Hereby, domain knowledge is of a particular importance, since domain-specific limitations and restrictions, regarding the hardware, energy efficiency, costs, legal aspects and many others, create a rigid corset that the scientist and practitioners need to consider. This is often specifically the case for application scenarios where humans or large-area environments are to be monitored with limited hardware resources. Therefore, there is a high demand for power-efficient hardware as well as computationally efficient approaches and algorithms that handle data collection, perform data abstraction or compression and further data analysis.

Motion and vibration play an essential and ubiquitous component of the physical world. Capturing them yields huge potential for various applications and thus has already been addressed with various types of hardware, preferably with inertial sensors. Among others, applications focusing on human motion are to be found in the health-care domain and fitness monitoring applications, which nowadays become more and more popular through inertial-equipped smartphones and other wearable devices. Acceleration sensors are used in commercial products and research projects, for remote control device implementations that are based on gesture recognition for data entry, control functions and gaming. Moreover, from acceleration data, motion or orientation information can be extracted, which then can be intelligently used in portable or wearable devices. For example, the device would be able to adjust the interaction interface, or adapt the system's behavior for more power efficiency.

Typical applications for vibration sensing applications lie in detecting erroneous vibrations for predictive and preventative maintenance of industrial equipment and machinery, identifying anomalous vibrations for safety or security event trigger, capturing acoustic vibration signatures for event detection, classification and tracking. Highly sensitive and specialized inertial sensors are heavily used for seismological equipment. Moreover, vibration sensors are used for structural health monitoring in bridges, tunnels and buildings.

Thus, due to the versatility of inertial sensors that are able to capture these physical phenomena both in its forms as acceleration and vibration, this work is specifically taking a data driven approach. We hereby rely on different application scenarios with human motion and infrastructure vibration as motivation for this work. The specific challenge in our case lies with building a coherent system, beginning with the actual hardware platform and its corresponding firmware for logging the data, backed with efficient data processing algorithms that are used to obtain desired information, such as motion or vibration patterns, from recorded data. The system is made complete by techniques that estimate from the gathered shape-based features either human motion patterns and activities or the cause of vibration.

With new application scenarios and corresponding constraints, this research field becomes more complex and more focused on systems that are custom designed for specific

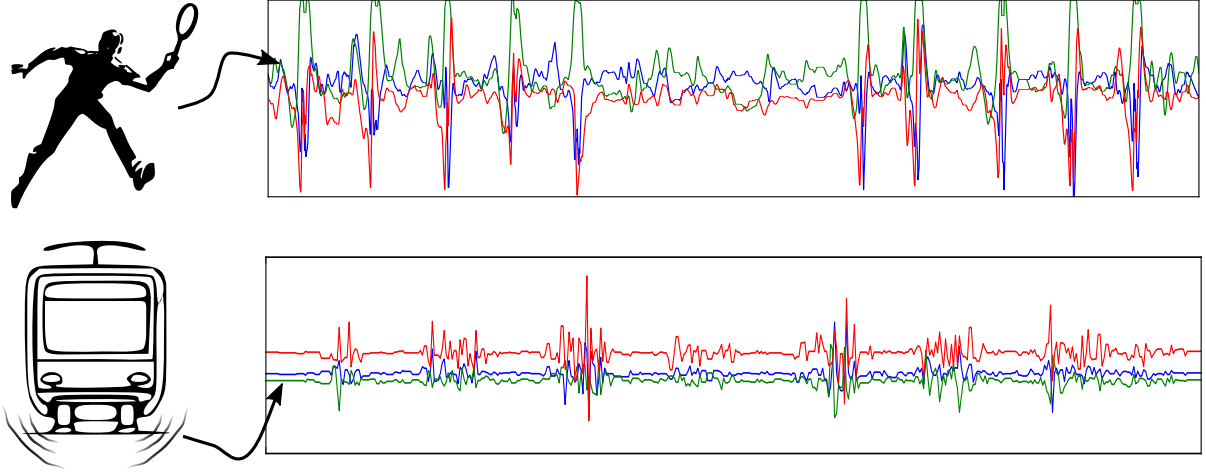


Figure 1.1: Motion and vibration can be picked up by MEMS accelerometers, resulting in acceleration data, which can be viewed as time series and analyzed for patterns. This thesis aims at using the shape of the signal as a feature to recognize characteristic recurring motion patterns and thus perform activity recognition, and discern moving objects by their vibration footprint.

purposes. The hardware-related components, including the hardware itself, as well as the firmware or embedded operating software, are often created for the target applications, relying on the domain knowledge and restrictions. The software components for data analysis are more and more tailored towards the specific application domain, considering the type of sensors used, their sampling frequencies, utilizing novel approaches to reduce computational cost as well as increasing the accuracy of predictions.

The next section presents the research problem addressed in this thesis, specifically focusing on the similarities of the chosen, quite different application scenarios.

1.1 Problem statement

This thesis addresses a class of problems that is created through a specific combination of the application scenario itself, the utilized sensors to achieve the task, and the amount of data that is created during sampling and to be processed afterward. Hereby, we are interested in the physical phenomena “motion” of humans and “vibrations” caused by moving objects, such as vehicles, that can be captured by micro-electromechanical systems (MEMS) accelerometers and visualized as a time series.

Due to various applications focusing on inertial data and aiming at identifying activities performed by a human, or events to be detected and classified within wireless sensor networks, this work proposes to use the shape of the signal in time specifically for these tasks: identifying and classifying different activities and complex vibration events from shape-based features.

Figure 1.1 shows such particularly interesting phenomena: a human performing activities with characteristic motion patterns (e.g. playing badminton) and vibrations patterns caused to the railway tracks by different types of trains.

To capture the essential details of human motion (which includes translation, tilt, rotation) as well as vibrations caused by moving vehicles, the accelerometer sensor's sampling frequency needs to be sufficiently high, leading to large amounts of raw sensor data. This fact very much impacts the sensor hardware design choices – when choosing among existing devices or designing a new device – and the algorithms that are used to handle the data during the logging process and afterward, demanding efficient algorithms to extract relevant information and to enable swift and at the same time accurate data analysis.

Considering these two application domains, the similarities in restrictions with regard to hardware and the need for efficient shape-based features become obvious:

- First, the hardware platform needs to be small, lightweight, energy efficient, robust, and inexpensive. The sensor needs to be built and packaged such that it will survive a long-lasting deployment at the wearer's body or in the open world environment, where it will be impacted by nature forces, including temperature fluctuations, humidity, dust, shock, splash water, etc. In the wearable application scenario, the usability aspect also plays a significant role, often requiring the addition of more components for a more in-suspicious and user-friendly experience, such as, for example, a display that can display time, date, and other information.
- Second, the sampling frequency is set relatively high in order to capture the physical phenomena of interest, resulting in large amounts of data, which is very critical, both during the logging process and in the following data analysis. Writing raw sensor data to a persistent local storage or wireless forwarding to a base station is very expensive with regard to energy consumption.

In the activity recognition scenario, human motion data has to be stored on the sensor until it is downloaded for further analysis to a more powerful machine. This makes efficient early data abstraction and compression mandatory, whereby the goal is to perform early data abstraction steps in an on-line fashion directly on the sensor device. For our WSN application, the raw vibration data should not be stored at all. Instead, it needs to be processed on-the-fly, such that only its much smaller abstraction or preferably the result (e.g. the identified event type) is reported to a base station.

- Third, we are focusing on the shape of the signal as a feature in both selected application scenarios. For human motion, these are recurring motion patterns that are characteristic for specific activities and thus can be used as evidence for activity spotting. In the second domain, vibrations caused by trains can be analyzed for their shapes, based on which the train types can be discerned.

To obtain required shape-based features, there is a high demand for abstraction algorithms that can significantly reduce the huge amount of raw sensor data while preserving the essential shape of the signal. With such a proper abstraction, further data analysis (i.e. pattern matching) is feasible in a tolerable time frame.

The following sections will present this thesis' challenges and contributions in more detail.

1.2 Challenges for Activity Recognition from Long-term Inertial Data

The challenges for long-term activity recognition from inertial data lie, first and foremost, in the sizeable data sets recorded by the body-worn MEMS accelerometer sensors. To be able to grasp human motion in whole detail, accelerometer sensors are to be sampled at relatively high frequencies (e.g. 100 Hz), resulting in more than 25 million 3D samples for a day. Focusing on specific health-care domain, the data sets obtained tend to consist of weeks worth of continuous raw sensor data. For activity recognition applications, the combination of high sampling rates and the long-lasting deployments poses a severe computational obstacle in further processing steps.

A lot of work in activity recognition from inertial data proposed extraction of features that abstract the sensor data and preserve interesting characteristics, such that these can be used as a basis for activity discrimination in common classifier approaches (e.g. SVM or HMM). Many of these approaches face a computational problem, due to the huge amount of samples and their in-distinctiveness with regard to class vs. non-class problem.

This work considers the shape of the signal as a very important feature that describes the essence of the human locomotion, thus requiring it to be preserved during the data reduction step. Depending on the actual application, the shape abstraction step thus consider how the abstraction will be used in the further data analysis. Here, I consider two challenges common in activity recognition domain: First, pattern matching from user queries or from a previously created data base against new data. Second, automated collection of patterns from a time series that are characteristic for an activity, aiming at building efficient classifiers for large-sized data sets.

In the first study, the challenge is to match few user-selected or pre-defined motion patterns against a continuous time series with weeks worth of sensor data in a fast and accurate way.

A second study aims at extracting recurring motion patterns that are characteristic for specific activities, such that these can then be used in efficient shape-based classifiers.

While there exist different techniques to reduce the amount of sensor data preserving the shape of the signal, few works have looked into optimizing the algorithms such that these can be directly implemented on a wearable sensor node.

To summarize:

- Human motion is captured continuously at 100 Hz and results in huge data sets, demanding early data abstraction to facilitate further analysis.
- Focusing on signal shape as a feature, the abstraction algorithm has to preserve the shape while reducing the amount of data.
- The abstraction algorithm should be efficient enough to be implemented directly on the sensor node.
- The impact of early abstraction should be low with regard to pattern matching and requires evaluation.
- To conduct activity recognition on long-term data, we need to efficiently extract unknown patterns that are characteristic to a target activity.

1.3 Challenges for on-line vibration event classification in WSNs

The challenges for on-line vibration event classification in wireless sensor network lie in our case specifically in the relatively high sampling rate, which results in amounts of data exceeding on-board operating memory. Since communication in sensor networks is the most expensive operation with respect to energy consumption, local on-sensor processing of data is required, such that only the classification result is transmitted to a base station or the backbone system. The demand for a long-term deployment and power-efficient hardware hereby constrains the processing capabilities and memory resource of the embedded system.

To avoid expensive wireless communication, the ultimate goal is to process the streaming data in an on-line fashion directly on the sensor node. The challenge hereby is to extract from high-frequent streaming data shape-related information that allows to perform event classification in real-time. With limited resources on the hardware side, such as only few Kilobytes of RAM to hold the streaming data or the lack of a floating point unit on the main processor, requires a study for finding suitable and easy-to-compute shape-preserving features.

To summarize:

- Vibrations are captured at 100 Hz; the amount of data exceeds on-board RAM.
- Forwarding sensor data is not possible, it must be analyzed on the sensor node.
- Events can be discerned based on their vibration footprints, such that appropriate shape-preserving features need to be found and evaluated.
- The classifier implementation must be simple enough to fit on the sensor node.

1.4 Challenges in hardware design for long-term deployments

Considering the development of hardware and software systems targeting application domains in fundamental research as well as health- or industry-related areas, the focus of this work lies with applications in which motion and vibration are the most important physical phenomena. Both, motion and vibration, physically defined by occurring accelerations, can be captured efficiently with off-the-shelf available MEMS accelerometer sensors.

The main requirement of the targeted application scenarios is what makes the development of efficient data analysis systems very challenging: the sensors are deployed as wearable data loggers for human motion logging or as main sensing components in distributed sensor networks, whereby the continuous operation time frames range from multiple weeks up to several months, respectively. With this, due to limited power supply in form of a battery, power efficiency of the sensor node becomes critical.

In contrast to the power-efficiency demand stands the amount of data generated by the accelerometer sensor. For capturing human motion or vibration data the sampling rates are set relatively high (100 Hz), resulting in lots of raw sensor data. Human motion data obtained through a wearable data logger has to be locally stored to a persistent memory, such that it can be downloaded and analyzed afterwards on a more powerful computer. In the wireless sensor network application, where vibration data is being captured, the streaming data has to be processed directly on the sensor node.

Many research prototypes as well as commercially available devices have some limitations in this regard. The former tend to record raw sensor data in short-term trials lasting few hours or days on a single battery charge; the latter tend to last much longer, but do not provide access raw sensor data, mostly storing features computed from the signal. Aiming at obtaining raw sensor data at high sampling frequency in a long-term deployment, novel designing the hardware and the corresponding firmware (logging procedure) of the sensor is critical.

To summarize:

- The data logger captures human motion or vibrations from trains at a 100 Hz.
- Sensor data needs to be efficiently compressed and stored for off-line analysis
- Sensor data needs to be directly processed and analyzed on the sensor node.
- Accurate time-stamps are required in both scenarios.
- Robustness and usability are mandatory to consider.

1.5 Contributions

With the challenges named above, this thesis' contributions in the two application scenarios are manifold in many aspects.

As a first contribution, we discuss our prototype system that includes a custom-designed hardware platform with a MEMS accelerometer unit to obtain raw sensor data, designed specifically to fit the harsh application demands and constraints of both applications. Secondly, we propose efficient data abstraction algorithms that reduce the amount of raw sensor data to its fraction, while preserving the essential shape information of the signal, thus facilitating further data analysis. Thirdly, we evaluate our proposed shape-based features by conducting multiple studies that show their efficiency and suitability for the chosen scenarios. The studies' results are hereby discussed in the context of the target applications.

1.5.1 Applications' demand for a custom hardware platform

The two application scenarios require a sensor platform that is able to capture acceleration and vibration (Figure 1.2) and that at the same time is small, lightweight, robust, power-efficient, inexpensive, and can be deployed for longer time spans. To this end, a custom sensor node has been developed. We will highlight some of the key design choices and present evaluation of the platform.

Considering the long-term human activity recognition scenario, we will also highlight additional work in making the sensor node as user-friendly as possible, which includes

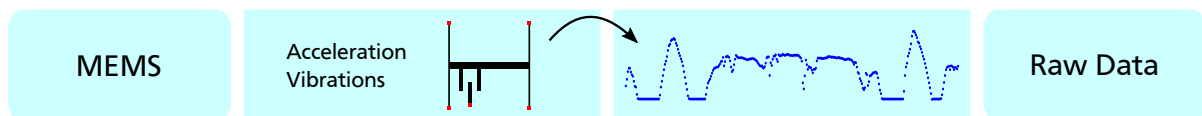


Figure 1.2: Capturing accelerations and vibrations with a MEMS accelerometer sensor at a predefined rate will produce what we call raw sensor data. For long-term deployments, an efficient design of the sensor device is crucial.

a protective encasing and the wrist-mounting strap. Additionally, we consider an extended sensor node that is equipped with a power-efficient display for in-suspicious deployment and a more richer user experience, e.g. for displaying time and date, recent events, providing interface for annotations.

Aiming at improved activity recognition performance, specifically with activities that are quite hard to distinguish solely with acceleration data, such as activities of daily living, we propose to include additional information. Hereby, it is widely acknowledged that specifically user-object interaction information yields huge potential. To this end, we attach an off-the-shelf available miniature RFID reader to our wrist-worn sensor, and design a bracelet-like antenna that is able to detect interactions with tagged object.

1.5.2 Efficient on-line data abstraction technique

Detecting recurring activities, routines and trends can be very helpful in medical monitoring applications. The activity recognition scenario thus focuses on long-term deployment of the wrist-worn sensor in order to obtain such detailed information about the wearer's routines. Using wearable inertial sensors in medical monitoring applications with relatively high sampling frequencies will result in huge data sets. The size of the recorded data is a computational obstacle on the way towards data inspection, annotation and post-analysis, i.e. activity detection and classification.

Following this activity recognition scenario, Chapter 4 proposes a method for sensor data abstraction in order to reduce the amount of data points but still preserve the essence of the signal. The approximation algorithm specifically targets human motion data and produces a piecewise linear approximation (Figure 1.3), which can also be visualized as a time series. We evaluate how our algorithm performs against existing work, by varying the approximation parameters and comparing the approximation error as well as the time required for processing a data set. Furthermore, we evaluate the implementation of the algorithm directly on a sensor device, thus moving the early abstraction step to the logging platform.

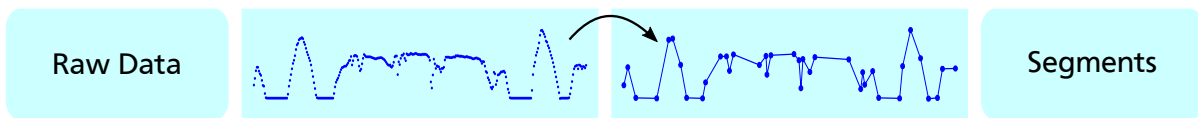


Figure 1.3: Approximating raw sensor data by linear segments significantly reduces the amount of data while preserving the shape of the signal, facilitating faster pattern matching.

Besides these basic evaluations, we are specifically interested in how such an abstraction impacts the finding of visually similar patterns. Our human activity recognition application scenario specifically requires query-by-example pattern matching for data annotation purposes. To this end, we evaluate pattern matching of linear segments with a traditional approach, namely dynamic time warping, and also propose a novel algorithm that considers only the most significant segments of the pattern.

1.5.3 Efficient shape features for WSNs in railway monitoring scenarios

Aiming at infrastructure monitoring with inertial sensors, we propose a railway monitoring scenario with a wireless sensor network. The sensor nodes deployed on the rails are able to sense the vibrations caused by the running trains, and from this data extract relevant information that has to be provided to the monitoring system backbone.

For this, in Chapter 5 we present a wireless sensor network deployment for railway operation monitoring. By using a set of small and power-efficient, yet computationally sufficient sensor nodes, a data set with vibrations from running trains was obtained from two geographically different deployment locations. A study then investigates various efficient features that can be extracted from streaming data in an on-line fashion (Figure 1.4), which then can be used to predict the type of train, as well as estimate further important details such as the number of wagons in the train, train speed, traffic density, or even detect potentially defective wheels that need replacement.

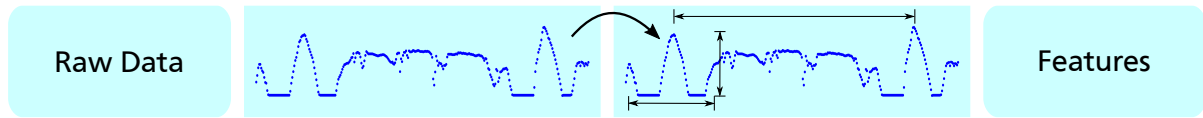


Figure 1.4: In wireless sensor networks, the shape of the vibration footprint can be represented by simple time-domain features, such that on-line event classification can be performed.

1.5.4 Long-term activity recognition with Dense Motifs

Performing manual data annotation is a very tedious task, especially when the user is confronted with long-term data, and the desire for more autonomous system is prevalent. In this part of the work we address an activity recognition scenario, where selected activities have to be recognized as autonomously as possible. We argue that many activities can be discerned based on characteristic repetitive motions, such as playing badminton with its forehand and backhand motions.

Chapter 6 presents our approach that aims at finding repetitive motions (motifs) that are specifically characteristic for an activity and can be used to discern the target activity from other activities. The main problem hereby is to efficiently find such patterns, since an excessive query-by-example search, with randomly or systematically chosen patterns of linear segments, is not desirable due to runtime complexity.

Our contribution in finding recurring patterns is the utilization of a search-oriented data structure called suffix trees, which is heavily used in bio-informatics and text processing for matching symbolic sequences. In order to be able to use this data structure, we introduce an additional discretization step that converts the linear segments time series into its symbolic representation, namely a string of characters. For that, we evaluate different discretization approaches based on the properties of the linear segments (e.g. length, slope, angle between segments, etc). With the symbolic representation of the data and the search-optimized data structure available, the motion patterns (motifs) can be easily extracted (Figure 1.5).

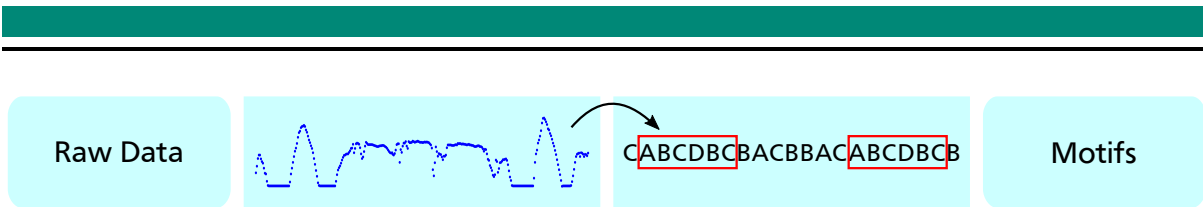


Figure 1.5: A symbolic representation of the original data facilitates efficient extraction of recurring motion patterns (motifs). These motifs can then be used as weak detectors in a simple bag-of-words classifier for activity recognition.

With these motifs, we are able to build simple bag-of-words classifiers, which can be applied to abstracted new sensor data, resulting in detections. Based on the density of these detections, our simple classifier then can decide whether a specific region should be labeled as the target activity, or considered background data.

To evaluate our approach, we conduct a detailed study with 33 participants with a total of more than 3800 hours worth of continuous data. Our study evaluates the performance in detecting various activities against generic approaches with signal features and common SVM classifiers.

1.5.5 Detecting interactions with objects

Chapter 7 investigates the usability of a wrist-worn bracelet that combines an acceleration sensor with an RFID reader for long-term deployments. We aim at improving the activity recognition for activities that are hard to classify solely from acceleration data obtained from a human’s wrist. This is achieved on the hardware level, by introducing RFID sensing to detect tags attached to various objects (Figure 1.6).

With only limited amount of freely available information on the design and manufacturing process of a bracelet-like RFID antenna with a sufficient reading range, sharing the hardware design for reproducibility is considered an important contribution.

After manufacturing the sensor device, we present a novel antenna benchmark, called “box test” that can be used to identify how well RFID tags attached to various objects of different material can be detected. After that, the “gardening” study is carried out, where a participant performed gardening activities and had to use different tools to complete his tasks. Interaction with objects is detected by the RFID tags, while the inertial sensors pick up the motions on “how” the tools were used.

An extended long-term study evaluates the applicability of such a device in household settings aiming at detecting activities of daily living, such as vacuum cleaning, watering flowers, cooking, ironing, etc. While the experiment showed that deployment is possible and most objects can be successfully detected, the usability aspect of the bracelet should be improved.

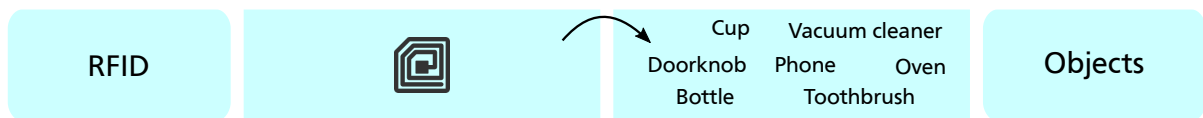


Figure 1.6: Detecting object interaction with wearable wrist-worn RFID readers yields significant information gain. With regard to long-term activity recognition, RFID tags detections can be considered as motifs.

1.5.6 Contributions summary

This thesis' contributions are summarized as follows:

- We motivate our work by two challenging application scenarios and argue for shape-based features.
- We present a custom data logger that is able to efficiently capture human motion and vibrations caused by trains.
- We propose various efficient data abstraction algorithms that preserve the shape of the signal.
- In multiple studies we evaluate the abstraction algorithms with regard to quality and efficiency of the abstraction or its impact on pattern matching.
- In a multi-user large-scale study we evaluate the use of dense motifs for activity spotting.
- A study compares various efficient features for vibration footprint abstraction through classification performance evaluation.
- For detecting human-object interaction and use, we build and evaluate a custom RFID bracelet.

1.6 Outline

This section gives an overview of the organization of this thesis, providing a short summary of each chapter, along with information on the originating publications. The thesis is structured as follows:

Chapter 2 – Related Work. This chapter provides an overview on some important related work and positions the thesis within. Aiming at long-term deployment of wearable sensors for human activity recognition or wireless sensor networks applications, we focus on efficient data representation approaches. Hereby, our main target are efficient data abstraction techniques and fast feature extraction that should preferably run directly on the sensor node and still be usable for pattern matching and for classification purposes.

Chapter 3 – Designing an Efficient Activity Logger. This chapter presents the custom acceleration data logger that has been designed, built and used specifically for obtaining human motion data in long-term deployments as well as vibration data from the railway tracks. By considering various application challenges, the system design choices are discussed and thoroughly evaluated. The main focus hereby was put on energy efficient implementation and an evaluation of how different hardware components as well as multiple configuration parameters impact the current consumption and how this affects the life time of the sensor node.

The work of this chapter currently undergoes peer review at SAS 2015 conference as a full paper with the title “Low-power Lessons from Designing a Wearable Logger for Long-term Deployments”.

Chapter 4 – Efficient Data Abstraction with Linear Segments. This chapter presents an efficient on-line approximation algorithm that works well on human inertial data,

abstracting raw sensor data to its piecewise linear representation and by that significantly reducing the amount of data, while preserving the essence (shape) of the signal. After presenting the main characteristics of the algorithm, two studies show how the proposed algorithm can be used productively: The first study focuses on applications targeting at post-analysis and annotation of recorded sensor data as well as activity recognition. The study shows that the abstracted time series allows more efficient query-by-example pattern matching and activity classification, whereby we also present a novel pattern matching technique that speeds up the subsequence matching process. The second study presents the implementation of the abstraction algorithm on a wireless sensor node, focusing hereby on the implementation challenges as well as the impact of on-line data approximation on the power consumption of the sensor node.

The work of this chapter was published in three papers a) at ISWC 2009 in “When Else Did This Happen? Efficient Subsequence Representation for Matching in Wearable Activity Data” (Van Laerhoven and Berlin, 2009); b) at ICMLA 2009 in “Enabling Efficient Time Series Analysis for Wearable Activity Data” (Van Laerhoven *et al.*, 2009); and c) at SenseApp 2010 in “An On-Line Piecewise Linear Approximation Technique for Wireless Sensor Networks” (Berlin and Van Laerhoven, 2010).

Chapter 5 – Complex Event Classification in WSN. This chapter is dedicated to efficient data abstraction in wireless sensor network applications, where the shape of the signal turns out to be not useful. In our study we evaluate the deployability of a wireless sensor network for railway monitoring, where detection and classification of sparse complex events (passing trains) in or close to real-time is required. Hereby, we investigate various features that can be efficiently extracted on-line facilitating event classification directly on the sensor node, thus reducing the amount of communication in the network, extending its lifetime.

The work of this chapter is covered by two publications: a) INSS 2012 in “Trainspotting: Combining Fast Features to Enable Detection on Resource-constrained Sensing Devices” (Berlin and Van Laerhoven, 2012b) and b) DCOSS 2013 in “Sensor Networks for Railway Monitoring: Detecting Trains from their Distributed Vibration Footprints” (Berlin and Van Laerhoven, 2013).

Chapter 6 – Activity Recognition Through Dense Motif Discovery. In this chapter we focus on long-term deployment applications with wearable sensors, where huge amounts of data need to be processed and analyzed afterward as fast as possible. In our study, motivated by a long-term medical monitoring scenario, we propose a system consisting of an unobtrusive wearable sensor and an efficient activity inference system that relies on early data abstraction (abstraction algorithm from Chapter 4 followed by a symbolic representation of segments) and that uses the salience of recurring motion patterns (motifs) for activity spotting.

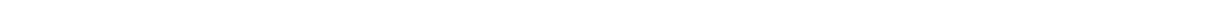
The work of this chapter was published at Ubicomp 2012 in “Detecting Leisure Activities with Dense Motif Discovery” (Berlin and Van Laerhoven, 2012a). A significantly extended study with a much larger data set and discussion currently undergoes the review process of IEEE Transactions on Pattern Analysis and Ma-

chine Intelligence (TPAMI) as article with the title “Using Saliency of Motifs as Evidence for Activities in Long-term Wearable Data”.

Chapter 7 – Detecting Interactions with Efficient Wrist-worn Sensors. In this chapter we report about the design details of a wrist-worn bracelet-like sensor platform consisting of an acceleration sensor and an RFID reader that is able to sense nearby RFID tags attached to objects. This combination allows to detect “what” objects the user was interacting with and “how”. Hereby, we evaluate the detection range of the new bracelet-like antenna on different objects with our novel “box test” benchmark. Enriching inertial sensor data with RFID tag detections yields the benefit of a much more detailed insight and improved classification performance, especially in cases where the classification is hard to do solely for acceleration data.

The work of this chapter was published at TEI 2010 in “Coming to Grips with the Objects We Grasp: Detecting Interactions with Efficient Wrist-Worn Sensors” (Berlin *et al.*, 2010).

Chapter 8 – Conclusion and outlook. In this chapter, we summarize the contributions and the results of this thesis and give an outlook on possible directions of future work.



2 Related Work

The shape of a time series can be considered as a feature in various applications. Our main focus is acceleration sensor data and efficient data abstraction techniques that facilitate detection of motion patterns for human activity recognition purposes or event classification from vibration data in wireless sensor network monitoring deployments. This chapter gives an overview of relevant related work and positions the thesis within.

This chapter is organized as follows. First we present important work on time series representation. We then present activity recognition and WSN monitoring applications with a prominent view on the used data representation, to additionally motivate and frame our own work. We hereby consider a variety of important points, including the challenges of the whole processing chain: the hardware platforms and the lengths of the deployment, the different data abstraction methods and utilization of various features to achieve the desired task, namely the dimensionality reduction of sensor data while preserving its essence, the detection of patterns, and finally the classification of activities or complex events.

2.1 Abstracting Sensor Data While Preserving Its Shape

This section presents various time series representation techniques and discusses their characteristics with regard to the chosen application scenarios constraints.

Time series representations or abstraction techniques can be, according to Lin *et al.* (2003, 2007), generally divided into two categories: data adaptive and non data adaptive (Figure 2.1). Hereby, the first type of techniques will produce an approximation that minimizes the overall approximation error, while the latter considers local properties of the signal and computes the parameters of the approximation at the potentially higher approximation error.

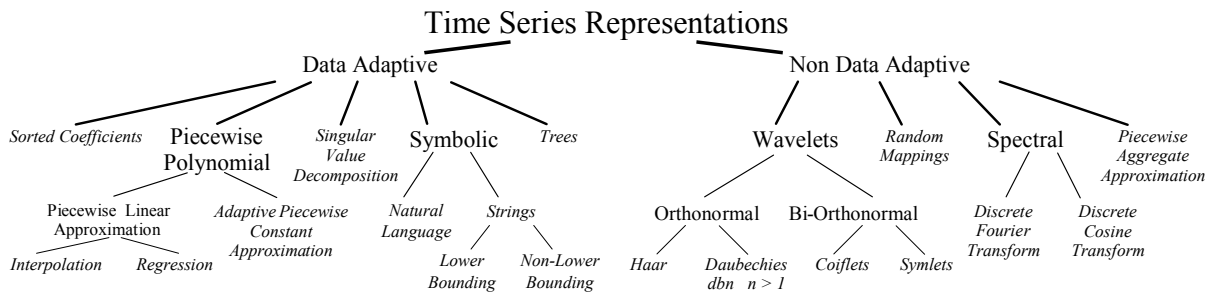


Figure 2.1: Various time series representations available in the literature. [Image used with kind permission of Lin *et al.* (2007).]

We take an important characteristic into consideration, namely the “visualizability” of the time series representation. While many approaches can be visualized straight forward as a time series, there are also many transforms that produce parameters only,

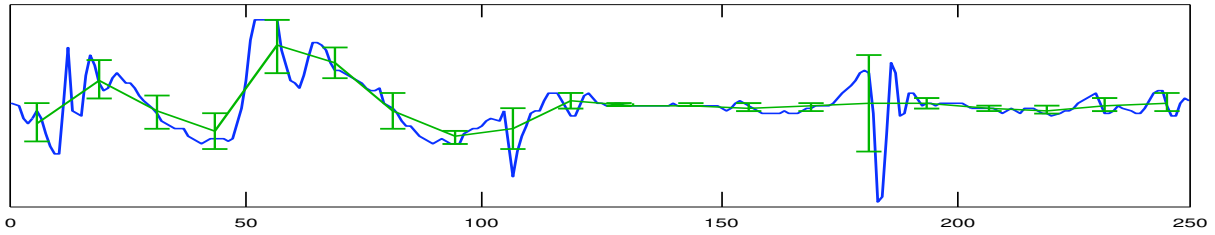


Figure 2.3: Abstracting a time series with segmented mean and variance. Here, the mean values can be connected by linear segments for better readability, and the variance is shown as error bars. This representation reduces the amount of data and facilitates faster pattern matching.

Yi and Faloutsos (2000) and Keogh *et al.* (2001a) have independently proposed to use the mean value over a fixed window size for dimensionality reduction of a time series, thus facilitating fast similarity search, naming the techniques piecewise constant approximation (Faloutsos *et al.*, 1997) and piecewise aggregate approximation (PAA), respectively. According to the time series representation overview given in Figure 2.1, this technique is considered a non-data adaptive technique.

Chakrabarti *et al.* (2002) have relaxed the fixed window size constraint, which leads to a data adaptive technique that was called Adaptive Piecewise Constant Approximation. The advantage of this approach lies in a better overall approximation (when following the above definition), but at a cost of storing more data, since not only the mean value but also the length of the segment have to be stored. This approximation technique can also be implemented in such a way that it considers local peaks in sensor data and approximates these more accurately, by, for example, growing the constant segment until a specified variance threshold in the window is reached.

To preserve the potentially important information about the amount of sensor value fluctuation, combining the mean value of the window with the variance results in a mean-variance representation. This representation can also be easily visualized: a visual example of such representation can be found in Figure 2.3. The importance of mean and variance in activity recognition domain as well as in wireless sensor networks is tremendous, since it can be used for event detections and classification, and more efficient pattern search than on raw sensor data.

Mean and variance have been widely used in the human activity recognition scenario, since the mean value tends to capture the local posture of the body, while variance describes how much motion is present in the signal (Foerster *et al.*, 1999). The combination of mean and variance has also been used with much success in detecting high-level activities by calculating them over large sliding windows (Huynh *et al.*, 2007). These features have been used effectively when combining multiple body-worn sensors (Ogris *et al.*, 2008) or in short sliding windows with an HMM-based approach (Ward *et al.*, 2006).

In various wireless sensor network scenarios, computing mean and variance on a small buffer of streaming sensor data and applying a threshold have been commonly used to, for example, detect and monitor volcano eruptions (Werner-Allen *et al.*, 2005, 2006) or to monitor variability of soil water content (Bogena *et al.*, 2010) over long stretches of time. Furthermore, variance describes the total amount of vibrations caused

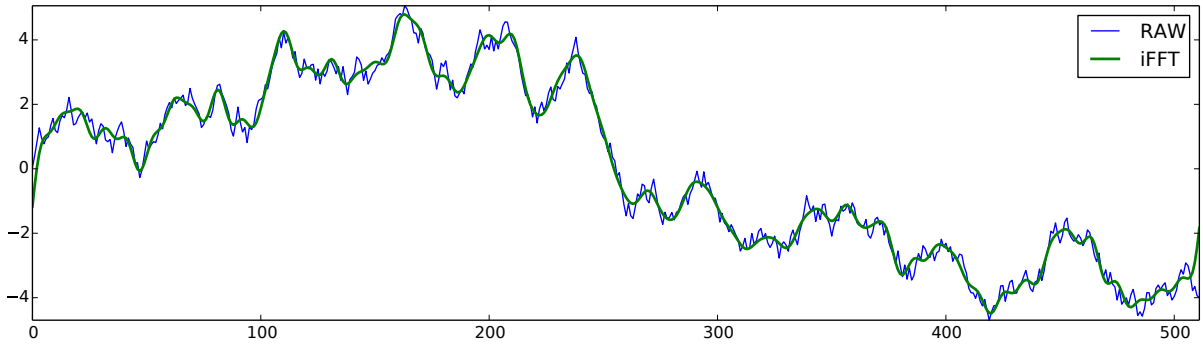


Figure 2.4: Discrete Fourier Transform of a random walk time series. Zeroing small FFT coefficients will remove the noise and less important peaks, resulting in an approximation that retains the main shape of the signal. To visualize the representation in the time domain, inverse FFT reconstruction is necessary.

by an event, such as a passing vehicle to the ground or a train to the rails, such that in combination with other features it can be used for event classification. More interesting is the variance computed over a short sliding window, which would allow to detect vibration peaks that correspond to wheel impacts.

2.1.3 Discrete Fourier Transform

One of the very prominent methods to analyze time series in the frequency domain is the Fourier Analysis, whereby for time-discrete applications the Discrete Fourier Transform (DFT) is widely used in a multitude of disciplines. The original time series is represented by an infinite sum of sine and cosine terms, where the coefficients of the terms are the actual resulting abstraction. Based on these coefficients, most prominent frequencies can be extracted and used for similarity searches, as suggested by Agrawal *et al.* (1993).

From the computed coefficients, the whole time series can be reconstructed. Zeroing less important coefficients leads to an approximation of the time series, where noise or less important peaks are removed, preserving the main frequencies and thus the shape of the signal. Figure 2.4 shows a random walk time series along with a DFT representation (computed through the inverses DFT), where the smallest 2% of the coefficients have been zeroed.

While this time series representation is very popular and widely used, even the Fast Fourier Transform (FFT) algorithms come at a high computational cost of $O(N^2)$ to $O(N \log(N))$, as for example was shown by Duhamel and Vetterli (1990), which is specifically problematic when considering large amounts of data. This is of particular importance when considering this thesis' long-term deployment scenarios, where compression of the sensor data or their analysis should be performed on-line on computationally very limited hardware platforms. Furthermore, visualizing the DFT representation of the time series is similarly costly, since it requires an additional re-construction step (the inverse transform) from the frequency domain coefficients into the original human-readable time domain.

Several other features in the spectral analysis are based on the Fourier Transform and are thus also costly to calculate. On the other hand, the frequency domain repre-

sentations have resulted in better pattern matching or classification performance than the previously presented mean and variance. Autocorrelation, Discrete Fourier Transform, Entropy and filterbank analysis can be expected to work especially well on signals with dominant frequencies, and have been identified as superior in several comparison studies (Bao and Intille, 2004; Huynh and Schiele, 2005).

Frequency analysis is specifically important in various vibration sensing and monitoring scenarios, such as in geophysical or infrastructure monitoring. Computing the FFT, on the other hand, usually requires more powerful hardware than a low-cost low-power WSN scenario offers. We show in Chapter 5 that even simple features extracted from vibration data are sufficient to detect and classify events and facilitate long-term deployments for railway monitoring.

In Chapter 6 we utilize the DFT based features in combination with a SVM classifier as a comparison technique for our dense motifs activity recognition approach.

2.1.4 Piecewise Defined Functions

Approximation of time series with piecewise defined functions aims at preserving the shape of the signal while reducing the amount of data has a long ranging tradition. The following paragraphs will present the more relevant piecewise polynomial representation approaches, along with the advantages as well as limitations. We start with polynomials of degree one, which are called linear approximations, and then consider higher level polynomials.

Piecewise linear approximation

Approximating a time series with linear segments ranges back decades (Stone, 1961; Cameron, 1966; Phillips, 1968). The goal hereby is to represent a given time series of values with a set of linear segments, constrained by a maximum number of segments or a maximum approximation error. The main advantages is that the amount of data can be significantly reduced, and the resulting approximation stays in the time domain.

While the above mentioned piecewise constant approximation techniques computes the mean value or the median from a sliding window of fixed or variable width, piecewise linear approximation (PLA) use interpolation or regression approaches to compute well fitting linear segments with varying slopes and lengths (Figure 2.5). In this work we are more interested in the interpolation approximation techniques, since these visually produce a continuous abstracted time series, whereas their regression pendants do not. Generally, PLA techniques produce representations that have a smaller approximation error and preserve the shape of the time series better, when compared to piecewise constant approximations, but at additional computation cost as well as the amount of information to store.

Various algorithms have been proposed aiming at improving the approximation quality or making these computationally more efficient. Among others, there are the Sliding Window, Bottom-Up or Top-Down approaches, as well as various combinations of those, such as the SWAB by (Keogh *et al.*, 2001b). These techniques differ with respect to the quality of their approximations, and also with regard to the ability to be performed in an on-line fashion on streaming data, which is specifically important in our human activity

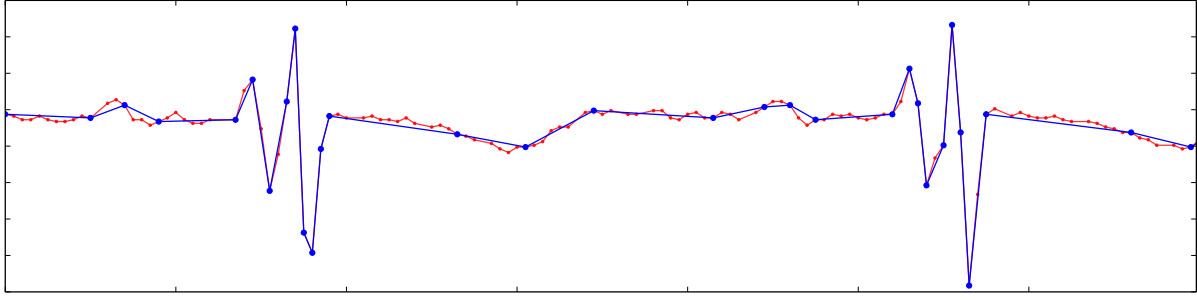


Figure 2.5: An example of a piecewise linear approximation of a time series. Various existing algorithms differ with regard to their on-line or off-line capability, computational complexity and quality of approximation.

recognition scenario. The other constraint is the quadratic computational complexity with respect to the length of the time series.

While the Sliding Window approach can be implemented in an on-line fashion, its quadratic complexity and the final approximation error have been shown inferior when compared to the two Bottom-Up and Top-Down approaches. These two approaches, on the other hand, cannot be performed on streaming data. SWAB combines the Bottom-Up approach on a small buffer window to produce segments with the Sliding Window technique that is responsible for filling the buffer with new data, resulting in an on-line algorithm with a “semi-global” view on the sensor data.

The positive impact of compressing data in such a way is obvious: Using the approximating linear segments rather than the initial raw sensor data, searching for similar patterns in the time series can be carried out more efficiently. Pattern matching in our application example is mostly carried out by taking a query pattern (a set of adjacent segments) and by matching it against the whole or a subset of the time series. This is generally implemented by computing the distance between the segments’ points of the query pattern to the time series, whereby the sum of euclidean distances is often used as a distance metric. Working with patterns with much less data points, the computation of the distance metric can be performed much faster.

With a linear approximations produced by SWAB or its modifications, matching subsequences of these linear segments can be carried out, such as shown in (Van Laerhoven *et al.*, 2009). Due to a varying number of segments in these patterns, matching can be improved by using dynamic programming or dynamic time warping techniques (Bellman, 1957).

In Chapter 4 we present a modification of the original SWAB approximation algorithm, that aims at human acceleration data characteristics and reduces the computational complexity, such that the algorithm can be implemented directly on the sensor hardware.

Polynomial approximation

Instead of approximating time series with linear segments, fitting piecewise polynomial functions to the sensor data has been shown very beneficial to preserve the shape. One of the prominent algorithms is SwiftSeg by Fuchs *et al.* (2009), based on polynomial least-squares approximation with a sliding or growing window over the raw sensor

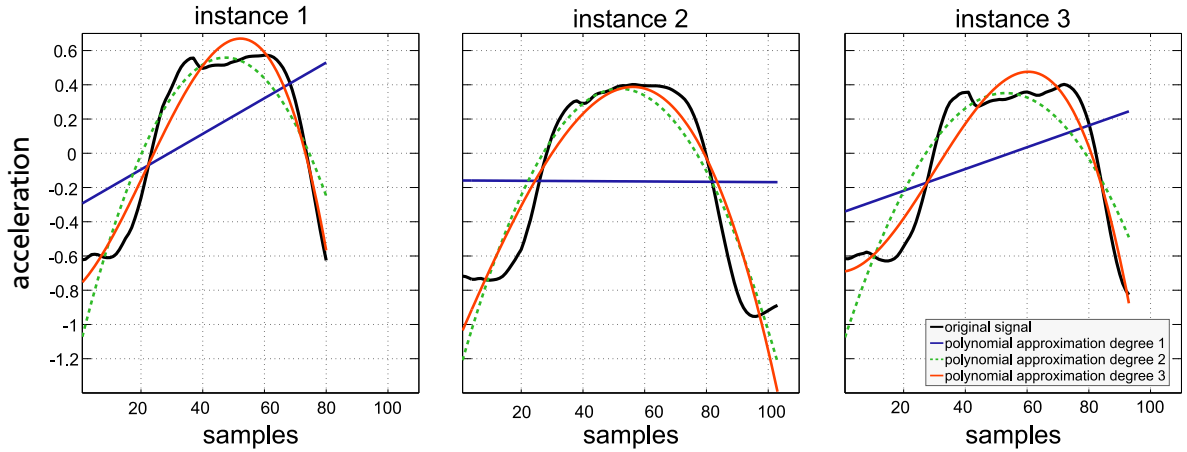


Figure 2.6: Approximating sensor data, in this example from body-worn inertial sensors featuring three instances of the same gesture, with polynomial functions of different degrees. The parameters of the resulting polynomial function can be used as features to train a classifier.

[Image IEEE ©2010, used with kind permission of Blanke *et al.* (2010).]

data. This approach offers some advantages with regard to piecewise linear techniques, such as a potentially better approximation with a lower approximation error, along with similar or even better data compression rates or computation runtime.

The resulting polynomial segments can then be considered as motifs (Fuchs *et al.*, 2010) and thus more efficient pattern matching or the search for anomalies in the time series can be performed. The authors hereby evaluate their approach on a subset of freely available datasets (Keogh *et al.*, 2011), which, unfortunately, did not contain human motion data.

Blanke *et al.* (2010) have applied the SwiftSeg approximation algorithm to human motion data obtained with inertial sensors, aiming at detecting particular interesting or relevant gestures. Hereby, after performing the polynomial approximation (Figure 2.6), the parameters of the polynomial functions were used as features to train the classifier. The approach, however, relied on using already known patterns and finding similar patterns on a relatively small data set with 45-90 minutes worth of data per person.

Due to our challenging long-term application scenario and with the prospect of implementing the data abstraction directly on the sensor node, using polynomial approximation is considered challenging, both from computational point of view, as well as due to technical restrictions of embedded micro-controllers, such as the lack of a floating point unit or limited operating memory. Because of this, the more “simplistic” approaches, i.e. piecewise linear approximation, are preferred in this work.

2.1.5 Symbolic Representation

One of the more challenging tasks in time series analysis is to find similar or recurring previously unknown patterns, specifically when the time series is very large and no a-priori information is available. While time series representation with statistical features, piecewise linear or polynomial functions reduce the amount of data and facilitate query-

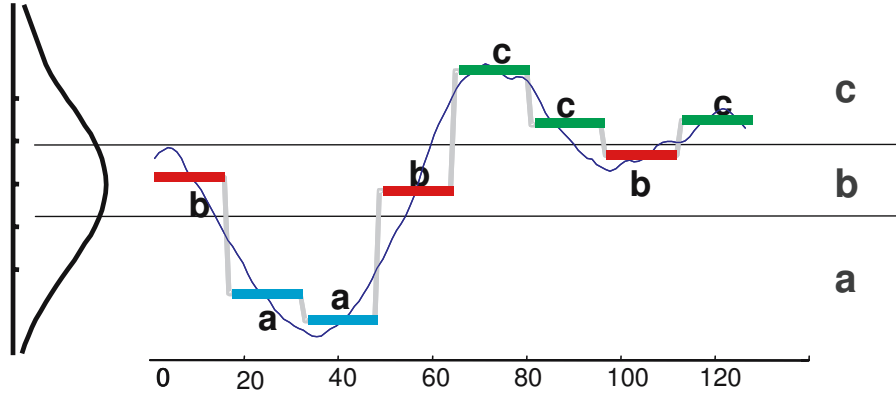


Figure 2.7: The symbolic aggregate approximation (SAX) technique first computes a piecewise aggregate approximation of the original data (colored in the plot) and then maps these constant segments to a set of characters (here of size 3), resulting in a symbolic representation of the time series: baabccbc. [Image ©2007, used with permission of Lin *et al.* (2007).]

by-example searches, the extraction of unknown recurring patterns (motifs) is often very tedious and computationally expensive.

Abstracting or encoding the time series to a symbolic representation, a string of characters, yields very promising possibilities. First and for most, a string can be used for fast sub-string searches or queering, relying upon efficient data structures, such as suffix trees (Ukkonen, 1992, 1995; Hamid *et al.*, 2007) or suffix arrays (Kim *et al.*, 2005), that are widely used in text searches (Baeza-Yates and Gonnet, 1996), genome matching (Barsky *et al.*, 2009), and other applications. On the other hand, which is even more important for our application scenario, these data structures allow efficient extraction of recurring patterns. Of course, we first need to obtain a symbolic representation of the time series.

There are various techniques to map a time series to its symbolic representation. (Lin *et al.*, 2003) proposed symbolic aggregate approximation (SAX), a technique that computes the mean of the sensor samples from a sliding window of fixed length, which is basically a piecewise constant approximation of the time series. The constant segments are then discretized to bins labeled with characters, thus producing the final symbolic representation. An example of this process is shown in Figure 2.7. In the most straightforward case, the window width is set to 1 sample, resulting in one character per sensor value, and thus in the string's length equaling the length of the time series.

Minnen *et al.* (2006) relies on the SAX technique to convert the acceleration and gyroscope data to their symbolic representation. With a set of beforehand known patterns, which were annotated during experiments, or in general should then come from the domain experts' knowledge, the authors concentrate on efficient and accurate detection of motifs in sparse multivariate time series. Hereby, generalized suffix trees are utilized to find motif occurrences, reaching overall good performance. The dataset of 27 minutes of sensor readings is rather small, compared to our long-term deployment scenario and the corresponding study.

SAX has been widely used for time series pattern matching, but there are some limiting aspects that are important to mention with regard to our work. By computing the

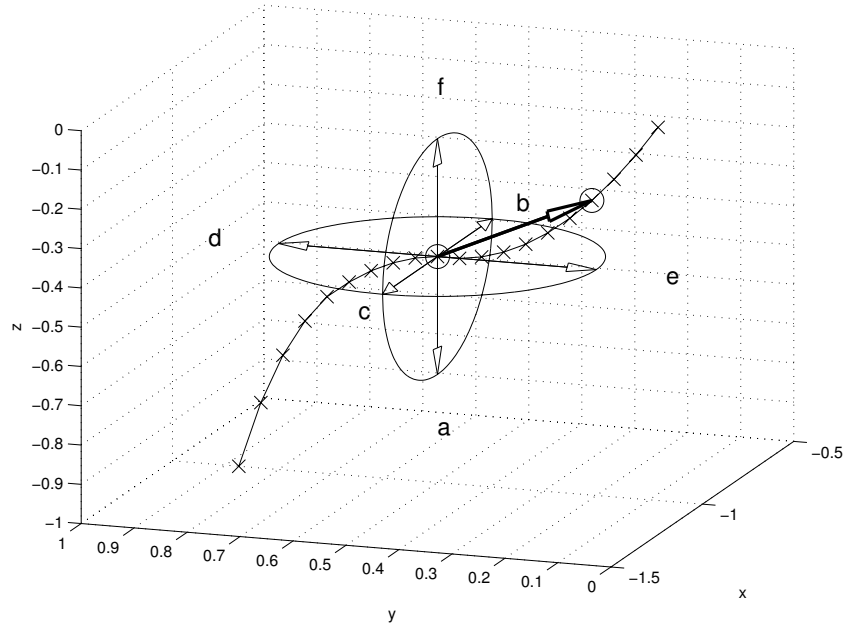


Figure 2.8: Mapping human motion traces in 3D space to their symbolic representation: First, the sensor values are aggregated into trajectory segments (here, with temporal aggregation of fixed size $w = 8$). Then, the trajectory segments direction is mapped to symbols, which form a trajectory string.
[Image ©2007, used with kind permission of Stiefmeier *et al.* (2007).]

mean value from a fixed width window results to some extent in information loss with regard to the essential shape of the signal, in our case specifically the characteristic peak patterns. Preserving the signal shape more accurately can be achieved by decreasing the window size, which results in a much longer string of characters. To this end, Hung and Anh (2007) propose to first perform SAX and find potential patterns, which are then filtered based on their shape encoded in PLA and their slope encoded to characters. The additional PLA step thus allows to use the shape as a feature and results in better overall similarity search.

Stiefmeier *et al.* (2007) considers human motion in space as trajectories, produces a 3D segmentation through temporal or spacial aggregation and then maps the direction of the vector to characters, thus producing a symbolic representation of the trajectory (Figure 2.8). Relying on fast string matching techniques, the aim is to detect recurring motion patterns (motifs) and thus to spot characteristic gestures for the bicycle maintenance activity. Specifically, this work aimed at the classification part, with efficient time series representation and detection of already obtained motifs. The accumulation or extraction of motifs (building a classifier) has not been primary target, though.

Our approach to abstract raw sensor data is based on a combination of PLA and SAX, and to some extent is comparable to Stiefmeier’s 3D segmentation approach. We first approximate the raw sensor data with linear segments, and then map these segments to symbols using the slopes of (or angle between) adjacent segments. Hereby, for the PLA step, we rely on our modified version of SWAB, which is presented in Chapter 4, and then use the discretization step in Chapter 6 for efficient extraction of motifs.

2.2 Embedding Abstraction Algorithms Directly on a Sensor Node

Performing the necessary feature extraction or data abstraction steps as early as possible has multiple, some already mentioned advantages. First and foremost, microprocessor computation is much more cheaper with regard to power consumption than writing data to a local persistent memory or forwarding it wirelessly to a base station. This thesis application scenarios specifically motivate early data abstraction, whereby additional effort is required to identify implementable representations that yield good potential in further analysis, i.e.: pattern matching, event classification and activity recognition. The following paragraphs will present some of the relevant related work that considers data compression, abstraction or feature extraction directly on the sensor node.

Early work (Barr and Asanović, 2003) already identified that it is more beneficial to process/compress data on a sensor node and only wirelessly transmit its compression over the network. Various lossless compression algorithms, such as LZW, bzip2 or GP-zip have been evaluated, comparing the algorithms' compression performance as well as the power consumption required to produce the compression. While the focus of the publication was put on sensor network applications where specifically the lossless compression of data is important, our approach targets at other applications where lossy compression of sensor data is allowed and even welcome. Still, the basic idea of reducing the amount of data by using relatively cheap CPU computation instead of power-hungry wireless transmission of uncompressed data holds for both approaches.

Run-length encoding (Golomb, 1966) is a very common and widely used method to compress data, also in the embedded systems domain including wearable sensors or wireless sensor networks. Capo-Chichi *et al.* (2009) presented the K-RLE algorithm, an adaptation of the common run-length encoding algorithm, which in essence is run-length encoding with a sensor value deviation threshold $K > 0$. The main advantage of K-RLE is that it not only compresses identical sensor values, but, with a correctly chosen threshold, can be used to filter out noise in the signal. It performs very efficiently (in terms of the approximation error) on flat data, or data with flat periods in the signal. On the other hand, if the sensor readings are constantly fluctuating, run-length encoding often results in even more data than the plain raw signal.

A good example of the use of features as data compression approaches is the Mercury wearable sensor network platform by Lorincz *et al.* (2009). The platform is designed to be used in the medical domain for on-line analysis of inertial data to monitor and detect specific events in human motion. The authors of Mercury are aiming at long-term deployments, and thus pay a lot of attention both to battery lifetime and hardware resource constraints. Their approach to handle high-fidelity data is to compute high-level features, such as mean, maximum peak-to-peak amplitude, peak velocity, and RMS of the jerk time series, from the raw signal and transmit only the features to the base station, thus preserving a considerable amount of bandwidth and energy. On the other hand, when the features are indicating a motion that is of particular interest, the raw data that was previously stored on the sensor node needs to be downloaded for more detailed analysis, which also results in additional power consumption.

The usefulness of piecewise linear approximation (PLA) in wireless sensor network applications has been shown by Le Borgne *et al.* (2007), where the aim is to reduce radio

communication to its minimum for energy preservation purposes. When a sensor node is monitoring the development of a physical variable over time, such as temperature or humidity, based on the readings it will choose a suiting model among a previously specified set of candidate models, and then only transmit the model parameters (e.g. a linear model and its parameters). Once the chosen model and the parameters have been transmitted to the base station, future sensor readings can be predicted, both at the base station or the sensor node itself, until the sensor readings exceed predefined application-dependent error bounds. Only then the model and its parameters have to be updated and transmitted to the base station again.

Our PLA method preserves the shape of the signal and avoids the storage or transmission of raw data as well, this way reducing battery power consumption. Still, depending on the application, encoding the shape of the signal with features yields potentially better results with regard to event classification, such as in our proposed WSN scenario.

One of the studies in Chapter 4 specifically benchmarks the impact of our PLA algorithm against the transmission of raw sensor data and the K-RLE compression technique.

2.3 Human Activity Recognition Applications

This section will present relevant application domains for activity recognition and motivate our approach with wrist-worn inertial data loggers and shape-based data analysis.

Since the activity recognition field has grown huge over the last decades, we specifically constrain ourselves to activity recognition from human inertial data. We are hereby interested in long-term deployment scenarios, where activity recognition is much more challenging, since the deployments are happening in the real-world, outside any constrained laboratory setting.

Monitoring the well-being of a person is one of the main driving forces for deploying wearable sensors and building systems that extract relevant information from the sensor data. Gaining more information about the patients daily routines, biorhythm, habits, medical conditions etc., allows for a more detailed insight, diagnosis or treatment as well as monitoring over longer time spans. In the health care domain various applications exist that will be addressed in the following.

2.3.1 Actigraphy

Actigraphy is a widely used non-invasive method in the medical domain for obtaining human activity/resting cycles with wearable sensors, ranging back decades (Godfrey and Knight, 1984; Royant-Parola *et al.*, 1986). It is used to assess human well being over long periods of time as well as finding characteristic sleep, performance, circadian rhythms, and trends during the monitoring period for various medical purposes (Tahmasian *et al.*, 2013). Sleep and mood disorders, such as insomnia, depressions, mania, bi-polar disorders (Burton *et al.*, 2013), attention-deficit/hyperactivity disorder (ADHD) (Teicher, 1995; Corkum *et al.*, 2001), impact the quality of life, and require professional medical attention. Therefore, it is of great interest to diagnose such diseases, to monitor their condition and progress as well as the impact of medication or treatment plans, to detect or predict mood phases and mood changes. Furthermore, Dijk *et al.* (2001)

took actigraphy to space, where it is being used for monitoring purposes of astronauts' biorhythms and provides information that allows them to adapt their schedule to ensure optimal fitness for critical tasks during space flights.

In the general medical domain, users usually wear actigraphs (data loggers with inertial and sometimes additional sensors) on their body and report on their mood and well being in detailed diaries. Various commercially available actigraph devices, such as the wrist-worn Philips Actiwatch¹, Camntech Motionwatch², ActiGraph GT3X+³, or the BodyMedia Sensewear upper arm sensor⁴, equidistantly sample the intensity of human motion and store these as the “activity counts” feature. The counts hereby range from “zero” during sleep or when lying down/relaxing, through “normal” activity levels during daily routines, up to “high” activity levels when sports and similar activities are performed.

An actigraph record, called actogram, generally looks similar to the one shown in Figure 2.9, and usually comes with general information about the patient (name, age, sex, etc.), as well as days with activity levels. Recording activity levels over long stretches of time (i.e. weeks or months), allows for detailed statistical analysis, which reveals characteristic activity level patterns and sleep/wake cycles and trends. Along with the diaries and regularly performed interviews, mood assessment of patients, diagnoses of conditions and their treatment is possible. In the long-term deployment scenarios, such as bipolar patients monitoring, psychiatrists will use statistical methods to extract information from the data, allowing them to assess their patients well being, current mood state, mood phase and phase changes, etc.

Recent work in this field has used both activity levels and segmented sleep and wake cycles of the patients. For a more advanced monitoring of the patients, detecting activities of daily living as well as hobby or physical activities is beneficial, which will also lead to reduction of detailed and time-consuming diary keeping, give a more detailed insight on patient's daily routines, and with that also increase the psychiatrists awareness on his patients current mood and state of well being. With regard to psychiatric disorders, such as bi-polar disorder, the detection of mood states and shifts or anomalies within is specifically important, since medication or activity treatment needs appropriate adaptation. For this, detecting relevant activities, preferably chosen for individual patients by the doctors in charge, is of significant importance.

Actigraph devices compute activity levels or counts from raw inertial data over larger time frames of usually 30 seconds up to 2 minutes, and raw sensor data is then discarded. To be able to perform gesture and activity recognition based on recurring patterns within the signal as proposed in our long-term monitoring scenario, obtaining raw sensor data or its abstraction that preserves the essence of the signal is mandatory. At the time when this project started, data loggers that would fit these requirements were not available off-the-shelf, and had to be designed and manufactured from scratch, as shown in Chapter 3. Recent work of Virkkala (2012) and Borazio *et al.* (2014) shows that such human motion loggers can still be used as actigraph devices, which addition-

¹ <http://www.healthcare.philips.com/main/homehealth/sleep/actiwatch/default.wpd>

² <http://www.camntech.com/products/motionwatch/motionwatch-8-overview>

³ <http://www.actigraphcorp.com/support/devices/gt3xplus/>

⁴ <http://sensewear.bodymedia.com/>

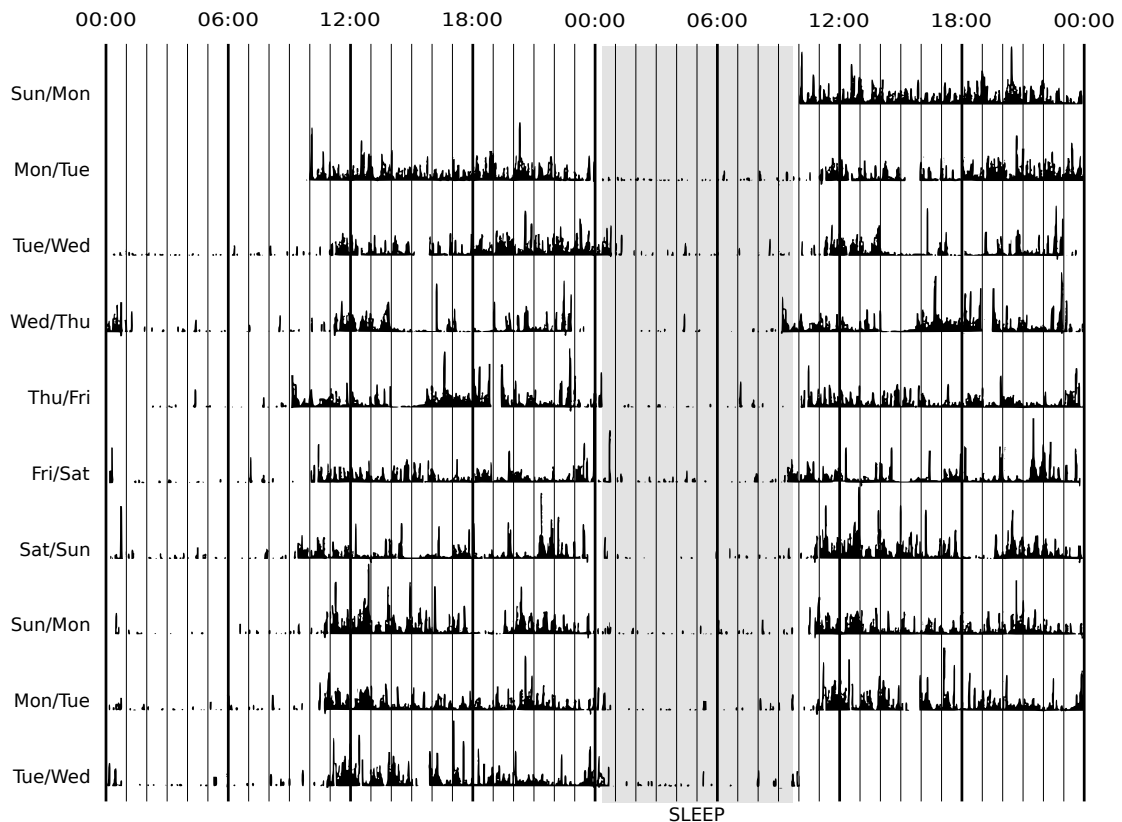


Figure 2.9: An actogram for a patient, as used in sleep and psychiatric monitoring, showing continuous long-term data with activity levels (bi-daily view). Utilizing statistical methods on such actigraph data, wake and sleep segmentation is possible. Additional mood assessment questionnaires as well as periodical personal interviews allow the psychiatrists to closely monitor their patients' phases, phase changes as well as the impact of medication.
[Image manually redrawn based on multiple real-world medical actogram records, for example as in (Jankelowitz *et al.*, 2005).]

ally motivates to preserve raw acceleration data, as both actigraphy and motion pattern detection is then feasible.

2.3.2 Human Motion Analysis in the Medical Domain

Monitoring human daily life and detecting the impact of physiological diseases has become one of the driving forces for the development of body worn sensors and body sensor networks.

Monitoring Parkinson's disease with wearable sensors Patel *et al.* (2009) aims at detecting tremors of limbs or frozen gaits. Similarly, using wearable sensors, epileptic seizures can be detected (Jallon *et al.*, 2009) and with appropriate systems a real-time alarm triggered (Gouravajhala *et al.*, 2012).

Lorincz *et al.* (2009) present a wearable, wireless sensor platform for motion analysis of patients being treated for neurological disorders, such as Parkinson's Disease, epilepsy, and stroke. While the system is designed for longer-term data collection in

hospital and home settings, the setup of up to 8 sensor is considered for short-term deployments in the doctors office or a hospital. For that, the system computes high level features, which are then used to train a linear classifier.

Most of the work in this domain, such as mentioned above, focuses on using time and frequency domain features to detect specific relevant events that are related to conditions of interest. On the other hand, aiming at preventive care, long-term monitoring of users' well being and way of life is required. For example, detecting and estimating intensity of alcohol (Leffingwell *et al.*, 2013) or tobacco (Scholl *et al.*, 2013) consumption, can be used for alarming the user about potentially harmful effects.

Wearable sensors have been also proposed for the psychiatric domain in form of actigraphy, as presented above, but also in form of human motion analysis (Teicher, 1995). Valenza *et al.* (2013) propose to use heart rate variability in the obtained ECG signal to recognize the four mood states (depression, mania, hypomania and mixed state) of bipolar disorder patients.

2.3.3 Activity Recognition for Elderly Care

Activity recognition as means for monitoring the ability of the elderly for self-contained living is a very popular research field. It aims at detecting human actions and activities with wearable (Bao and Intille, 2004; Huynh, 2008) or environmental sensors (Tapia *et al.*, 2004), and use these to assess whether an elderly person is able to perform the activities of daily living or requires additional substantial help. This is achieved through obtaining the context of the environment and the human within, from location and interactions with objects, from which human activities are inferred.

This area of research faces the challenge in identifying complex and generally very diverse activities which every person performs individually, and which are often interrupted or executed in varying order. Some of the more complex activities are for example: "preparing a meal", "washing dishes" or "cleaning the home".

Besides detecting just the gestures and motions of the human, it is often of particular interest to identify the location as well as objects the person was touching, moving, or using. For that, research has used various approaches, among others vision-based systems in the kitchen (Lei *et al.*, 2012; Intille *et al.*, 2006; Logan *et al.*, 2007). On the other hand, vision based approaches have specific limitations with regard to their user acceptance and raise privacy violation concerns.

Motivated by identifying and maintaining healthy eating routines, Amft *et al.* (2005) and Amft and Tröster (2008) have been focusing on detecting the details of eating activities. Hereby, various sensors were used, including inertial sensors to capture human motion, microphones for chewing sounds and electromyogram electrodes for muscle activity during swallowing. Motion was obtained from the user's lower and upper arms, such that eating and drinking gestures can be segmented and classified, hereby relying on the similarity of piecewise linear patterns, similar to our query-based pattern matching approach.

Recent work has also considered using capacitive sensors, both deployed in the environment and well as on the users' body, to facilitate localization, detection of important or critical events (e.g. falls) and activity recognition (Braun *et al.*, 2012; Grosse-Puppenthal *et al.*, 2012; Cheng *et al.*, 2013).

Using inertial sensor data only to infer actions and activities in this domain is often prone with false detections. Since a person is actively interacting with objects during such activities, detecting human-object interactions is considered as very important. To this end, various approaches have been proposed that rely on detecting the objects through RFID (Schmidt *et al.*, 2000; Philipose *et al.*, 2004; Patterson *et al.*, 2005; Fishkin *et al.*, 2005).

Combining multiple modalities, specifically human motion information from the inertial sensor, and human-object interaction from an RFID sensor, yields a more detailed insight and improves the activity classification performance (Stikic *et al.*, 2008; Stikic, 2010). Using the object interaction information also has the positive feature of potentially reducing supervision or simplifying annotation of the sensor data.

Unfortunately, integrating an RFID reader into a wrist-worn device and achieving a sufficient reading range as well as robust detection of grasped objects is not trivial. For others to be able to reproduce our results and aiming at serving the gained know-how to the community, we have taken the effort in extending our own wearable inertial sensor with an RFID reader and a custom-built bracelet-antenna. Additionally, we present a novel benchmark method that allows to evaluate and compare such custom designs.

2.3.4 Fitness Applications

The advances in the wearables domain have a huge impact on the consumer market. What started as step counters for walking or jogging or GPS trace recorders, has by now arrived at a multitude of interesting fitness applications, both as apps for smartphones or as dedicated wearable devices, such as Sony Smartband, Samsung Galaxy Gear Fit, LG Lifeband Touch, Waterfi Nike+ Fuelband, Adidas miCoach Smart Run, or the Fitbit devices family. By monitoring gym exercises, free-style jogging, generally tracking the overall activity levels of the users, and even incorporating additional information about the food intake, the devices and applications aim at motivating the wearer for a more balanced and more fitness-aware life-style.

While the usability and aesthetic design aspects tend to be superior to research prototypes in research, specifically due to the commercial interest of the manufacturers, with regard to the scientific approach there are major drawbacks. First and foremost, the sensor devices are mostly closed source and do not provide any access to the raw sensor data. The activity intensity levels are usually computed from epochs with a duration reaching from 1 second up to 2 minutes, thus discarding the original sensor data, which is actually necessary to obtain the shape-based features from the signal, similarly to actigraph devices mentioned above. On the other hand various existing applications are able to count gym exercise repetitions and offer statistical overview and guidance for optimized schedule.

The other problematic aspect concerns the power-efficiency of the hardware for long-term logging applications. Since most actively-used smartphones need to be recharged on daily basis, the hardware manufacturers for their fitness wearables rely on this widely accepted user behavior. Thus, the wearables are designed to last for this period of time with no additional effort being made to achieve much longer time spans, such as weeks, as desired for our long-term activity recognition scenario.

2.4 Wireless Sensor Networks Applications

Many research scenarios motivate the deployment of wireless sensor networks through the suitability of small sensors to densely monitor infrastructure, such as buildings, bridges, roads, rails. The huge diversity in sensors hardware, types of applications, deployment procedures, methodical and implementation approaches is astounding. One of the driving forces for monitoring structures and detecting relevant events is the goal for improving safety and organizing maintenance tasks. Various scenarios with alarm or control systems also motivate the detection and monitoring of critical events. This section will therefore present several application scenarios and frame our train monitoring study amid these related work.

Sensor networks have become a popular tool for various applications, due to being able to cover and monitor large areas and drastically reduce the intrusion into existing environments as well as disturbance of its inhabitants. The ability of wireless sensors to span a sensing and communication network with minimal resources by using small, robust, power-efficient and inexpensive hardware, highly benefits large-scale monitoring application. Such applications traditionally aim at periodic sampling of sensor values for long time periods, in order to obtain a detailed overview on physical phenomena in the environment. Many sensor network applications focus hereby on collective observation of slowly changing physical values, including temperature, humidity, gas concentrations in the air or particle concentration in the water.

Other popular sensor network applications aim at detecting sporadic events, such as abrupt rising or falling of temperature and humidity, extremely high or hazardous concentrations of gas or pollutants in the air. The ability of the sensor nodes to detect such events directly at the source is of great advantage to the whole network, allowing to significantly reduce the amount of wireless communications within the network, thus preserving the limited power supplies (Figure 2.10). Sensor networks have been increasingly deployed in scenarios with the aim to detect, monitor and report on more complex critical phenomena, such as seismic activity (Werner-Allen *et al.*, 2006), disaster detection (Ramesh *et al.*, 2009) or emergency scenarios (Gao *et al.*, 2007, 2008). These applications require high-fidelity sensor data that preferably should be analyzed in or close to real-time, which conflicts with the fact that wireless sensor nodes within a network tend to be heavily constrained by their hardware capabilities and resources.

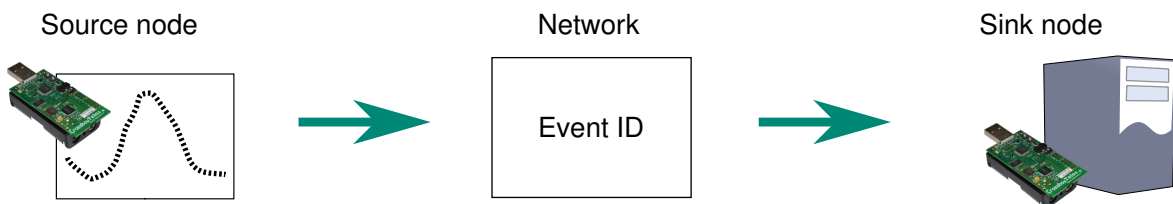


Figure 2.10: A wireless sensor network (WSN) is used to sense physical phenomena in the environment and report them to a base station for further analysis. Since wireless communication is very expensive (wrt. energy consumption), detecting and classifying events from the sensor data directly on the sensor node, will result in energy savings and longer sensor network lifetime.

A multitude of research, including (Mazarakis and Avaritsiotis, 2007; Gupte *et al.*, 2002; Dutta *et al.*, 2005; Keawkamnerd *et al.*, 2008) describe different application scenarios, where detection and classification of rare or sporadic events is of particular interest. The sensors that were deployed in these scenarios rely on vision-based, acoustic, seismic, magnetic and infrared sensors, and facilitate distributed observation of an area, aiming first and foremost at spotting and classifying ground vehicles or humans. While the scale of these deployments varies a lot, the need for energy-efficient sensors accounts for the features to be relatively simple to compute. For their car toll system application, Keawkamnerd *et al.* (2008) follow a similar approach to our work by choosing simple features (vehicle length, the average observed energy and peak-patterns in the signal) to detect and classify various ground vehicles, such as cars, pickup trucks, vans, buses and motorcycles.

Vibration sensors are often used for monitoring and ensuring infrastructure safety, such as by Huang *et al.* (2010) or Kim *et al.* (2007), where particular frequencies in the raw data have been considered as well as various complex features utilized.

Railway safety and train detection plays an important role both from practical as well as research point of view (Palmer, 2012), spawning several application scenarios utilizing different types of sensors. Angrisani *et al.* (2010) present a system for short-term deployments that uses accelerometers to detect arriving trains in order to warn maintenance personnel working on tracks. To enhance railway safety, Wang *et al.* (2006) deployed a vibration sensor on running trains, aiming at detecting rail deformations during motion. Donato *et al.* (2004) show how electromagnetic sensor arrays can be used to detect and count wheels. Aboelela *et al.* (2006) motivate the use of wireless sensor networks for railroad operation monitoring, aiming at increased safety and improved efficiency of railway maintenance.

Reducing the energy consumption in wireless sensor network deployments, is often very crucial for the lifetime of the deployment. Data compression approaches, such as by Reinhardt *et al.* (2009), aim at reducing wireless communication payload. In the wearable and mobile sensor research domains, limited power supplies require more data processing directly at the device, avoiding energy-consuming transmission or storing to local memory. Sun *et al.* (2011) aim at reducing the communication payload by detecting activities directly on a mobile device which extends the lifetime of the sensors.

Our work also focuses on a wireless sensor network application scenario, where observed events cannot be detected with simple threshold approaches and require on-line data processing. The aim is to predict train types and estimate train lengths by means of their vibration footprint directly on the sensor nodes. The events in this scenario tend to occur sporadically and last for only a short time period, but result in lots of data, since sampling is carried out at relatively high rates. The ability to detect, classify and monitor such events with sensor nodes deployed along railway tracks would allow to deploy such a network for various long-term railway applications.

2.5 Summary

This chapter presented relevant related work on time series abstraction, which is specifically necessary for dimensionality reduction to facilitate more efficient time series analysis. With the focus on long-term application scenarios with inertial sensors sampled

at relatively high frequencies, the amount of sensor data is huge. Considering inertial data as a time series, and being interested in identifying and classifying recurring patterns with shape-based features, our goal is to reduce the amount of sensor data while preserving the shape of the signal.

To this end, we propose a data logging prototype that is able to sample accelerations at a frequency of 100 Hertz and last for multiple weeks. We then deploy this data logger in two scenarios to obtain inertial data and to evaluate our approaches and techniques.

For our first application scenario, namely human motion analysis and activity recognition through query-based pattern matching, we propose an adaptation of a piecewise linear time series approximation technique that is almost twice as fast as the original version on human inertial data. With the approximation at hand, pattern matching can be carried out much faster than on raw sensor data. Additionally, we also propose to even more speed up the query-by-example matching by using the K longest segments (instead of all) of the pattern.

In the second application scenario, we focus on railway monitoring through the classification of vibration events based on the shape of the vibration footprint. Both the data abstraction as well as the classification need to be performed on a hardware and computationally constrained sensor node, which does not allow complex frequency analysis. Instead, we conduct a study revealing that simple and efficient-to-compute features can preserve the essential information from the signal and thus can be used to discern the events with a common SVM classifier.

Detecting physical or leisure activity can aid actigraphy and yields huge information gain for long-term monitoring in medical applications, such as bipolar patients monitoring. The main goal is the finding of recurring patterns that are characteristic to specific activities and can be used as weak detectors in a bag-of-words classifier. To this end we propose to use the salience of motif detections instead of the commonly used SVM and HMM classifier systems. We hereby rely on the piecewise linear data approximation which we consecutively convert to a symbolic representation. We show that this approach performs well on a multitude of activities, is able to match the performance of the common time and frequency domain features in combination with a SVM classifier, while being much faster.

Finally, we propose to improve activity recognition of our dense motifs approach by adding further sensor modalities. By detecting object interaction through RFID sensing, these detections may not only reveal which and how the user handled particular objects or tools, but also improve activity classification performance, especially in cases where activity classification is very hard to do solely from acceleration data.

3 Designing an Efficient Activity Logger

One of the hard challenges in creating and deploying robust activity recognition systems is the deployment of wearable sensors that are unobtrusive and lightweight, and therefore are limited in power and processing resources. This has also been identified in earlier research on mobile sensing platforms, for example by Choudhury *et al.* (2008).

Several publications in activity recognition research have observed that long-term operation of the wearable inertial sensor units is critical for various types of application domains. Long-term monitoring of psychiatry patients, for example, can be considered as one of the more challenging scenarios, due to its constraints regarding the type of sensors, its power efficiency, as well as usability considerations. While commercially available actigraphs are able to provide abstracted data in form of activity counts, few wearable platforms allow the users' activities to be captured as raw acceleration data for more than a few hours. The design of such modules therefore can be expected to impact such systems considerably.

This chapter presents a data logger that is specifically designed for long-term deployments to capture raw acceleration data (Figure 3.1) as efficiently as possible.

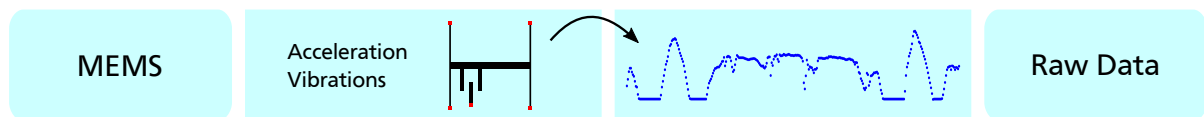


Figure 3.1: Capturing human motion with MEMS accelerometer sensors at a predefined rate will produce what we call raw sensor data. For long-term deployments, an efficient design of the sensor device is crucial.

3.1 Motivation

In our case, due to lack of available off-the-shelf sensor platforms that would fit the requirements, a custom sensor unit had to be designed and built. This has led us to consider building a prototype that would explore long-term activity recognition approaches with a focus on the recording equipment. A trade-off between two features was prevalent throughout this design phase:

User acceptance. A main requirement is that users accept to wear the wrist-worn units continuously, day and night. Design, size and weight are important, as pointed out by Narayanaswami *et al.* (2002), but also functionality: Although the unit was meant to record data, many users, especially in the medical domain, would only wear the device if it provided them the function of a basic wrist-watch, both for its functionality and for wearing something that would raise questions.

Low-power operation. On the other hand, the device needs to be power-efficient enough to be able to record inertial data at a high sampling rate for extended



Figure 3.2: The custom-made inertial data logger was designed for long-term (day and night) recording of data for activity recognition at 100 Hz, while simultaneously taking user acceptance requirements for wearing such as device into consideration. We focus on the evaluation of low-power design choices.

periods of multiple weeks at a time on a single battery charge, thus requiring components to be turned off or in sleep mode whenever possible.

We will focus specifically on the latter requirement of low-power operation, while assuming the constraint of user acceptance, which in our case led to the inclusion of a particularly small battery and the addition of an OLED display. Turning off the display whenever possible, and turning it on only when required by the user, is hereby the straight forward choice in preserving limited power resources. Figure 3.2 shows the current prototype.

Aiming at efficiently capturing human motion in its full detail at 100 Hz and storing it to local memory over long stretches of time, the remainder of the chapter presents the choices for designing the data logger that we have built for our experiments, as well as identifying the potential for power-efficient operation and logging of data.

3.2 Wrist-worn Unit Design

The custom-built platform is centered around a Microchip PIC18F46J50 microcontroller, which embeds in a small-scale form-factor key components for acquiring and recording inertial data. Among the most relevant features embedded in the microcontroller are the real-time clock, multiple internal oscillator circuits, digital and analogue communication interfaces, and a full-speed USB 2.0 communications module.

The real-time clock is specifically important to obtain accurate time stamps during logging, which are then used in the visualization of the sensor data to the user through a human readable time axis. Furthermore, the time stamps are necessary to synchronize the sensor data with user annotations, whether kept in diaries or added interactively on recall basis.

The main sensor unit of our platform is the 3-dimensional ADXL345 microelectromechanical system (MEMS) accelerometer, which is able to obtain accelerations in a range between ± 2 up to ± 16 g with sampling frequency up to 3600 Hz. The accelerometer sensor is connected via a Serial Peripheral Interface (SPI) digital bus with the microcontroller. In our experiments, the accelerometer sensor was configured to a sensitivity of ± 4 g at 10 bit resolution and a sampling rate of 100 Hz. The accelerometer unit itself comes with important features, such as a double-tap and fall detection, low-power

modes and an internal FIFO buffer that allows to transmit sensor values in bursts to the microcontroller. This latter feature is specifically important, since it allows the main processor to switch to power-efficient sleep modes or perform other tasks between these communication burst.

Due to the amount of data being generated at 100 Hz, a local flash memory is needed to store the acceleration data along with the time stamps. Non-volatile flash memory suits the application scenario demands, since it is available off-the-shelf in small form factors (microSD cards) and can preserve stored sensor data even when the battery runs out of energy. Connectors and circuitry are available on the sensor board for attaching the microSD card and for storage of sensor data. The microSD card is then transparently accessible via USB as a so-called mass storage device that appears to the user as a common memory stick with FAT16 file system.

In addition to the time and acceleration values, ambient light intensity is obtained through a photosynthetic diode. The light sensor is hereby sampled approximately once in a second.

The prototypes are powered from miniature Li-Polymer rechargeable batteries with a capacity of 180 mAh. For user-friendly recharging of the battery via USB the sensor is equipped with the MAX1551 charger integrated circuit.

To meet the requirements of long-term 24/7 deployment, the unit is packed in a custom shock-proof case and provided with an anti-allergic textile wrist strap. Furthermore, a miniature OLED display is used for visualization purposes and can display time, date or current sensor values. The display is by default turned off and can be activated by the user by double-tapping the watch.

3.3 Experimental Setup

This section focuses on the experimental setup to obtain current consumption figures for our prototypes. We first present the hardware setup to record the current drain traces. Aiming at performing reproducible tests for all our test cases, we propose to use a benchmarking platform. Lastly, to access how much a sensor would last in a real-world scenario, we deploy the sensor for multiple days to measure overall consumption.

3.3.1 Measuring Current Consumption

Critical in the power evaluation of the different parts of our prototype is the way we measured the individual components' energy consumption footprints. This section covers the details of our method to measuring and record the energy consumed by our prototype in particular modes. After presenting the basic principles of operation, we provide the details on our measurement setup based on the Arduino Due¹ platform. Large parts of this experiment were carried out by our technical staff: Martin Zittel and Michael Bräunlein.

Figure 3.3 depicts the schematics of the voltage measurement circuit with which we are able to obtain detailed recordings of current consumption, shown later in the eval-

¹ <http://arduino.cc/de/Main/ArduinoBoardDue>.

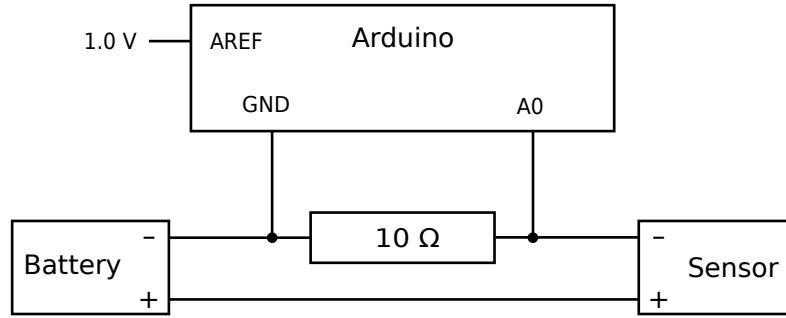


Figure 3.3: The schematics to measure current consumption via voltage drop over a resistor: To acquire and log the current consumption footprint at a high-enough speed and sensitivity for further detailed analysis, we use the Arduino Due platform supplying a reference voltage of 1.0 V.

uation section. Following the Ohm’s law, we can from the measured voltage drop over a sufficiently small resistor compute the current drawn by the sensor device:

$$I = \frac{V}{R} = \frac{1}{10}V \quad \text{with a constant resistor } R = 10\Omega$$

To access the current consumption of different operation modes of our prototype, we need to consider the current drain over the operation’s duration, which is given by the area under the curve for the current draw measurements.

Most relevant information for power profiling of a sensor device can be gathered by obtaining (a) the baseline current consumption of the sensor device when in low-power idle or sleep mode, (b) current consumption of different hardware components used to perform various tasks, such as sampling the accelerometer, writing data to persistent memory, or displaying information on the display. To obtain current consumption for different operations, we need to first identify these in our voltage data set, extract corresponding voltage readings and compute the area under the curve, for which we use the composite trapezoidal rule (*trapz* in the numpy python library).

The Arduino Due is connected to a computer and used as a high-resolution voltage logger over the resistor R on the analog A0 pin against the ground GND pin. A reference voltage of 1.0 V is supplied to the AREF pin, and power for both the prototype and the Arduino is provided by a regulated bench power supply. The A0 pin is sampled through an ADC at full speed, whereby the resolution is set to 12 bit. The measured values are transmitted to a computer for logging and further off-line evaluation (Figure 3.3).

3.3.2 The Robot Arm Benchmark

In order to obtain comparable current consumption figures for all the different configurations, for each of the tests the data logger should be moved in a similar way and for a similar amount of time. A human being is not able to perform the motions for each test as reproducible as required, which motivated us for finding a suitable platform.

Luckily, with multiple outstanding robotics research groups at the university, we were granted access to the BioRob research robot arm of Prof. Dr. Jan Peters, and were allowed to use it for our experiments. The BioRob robot arm is able to perform very

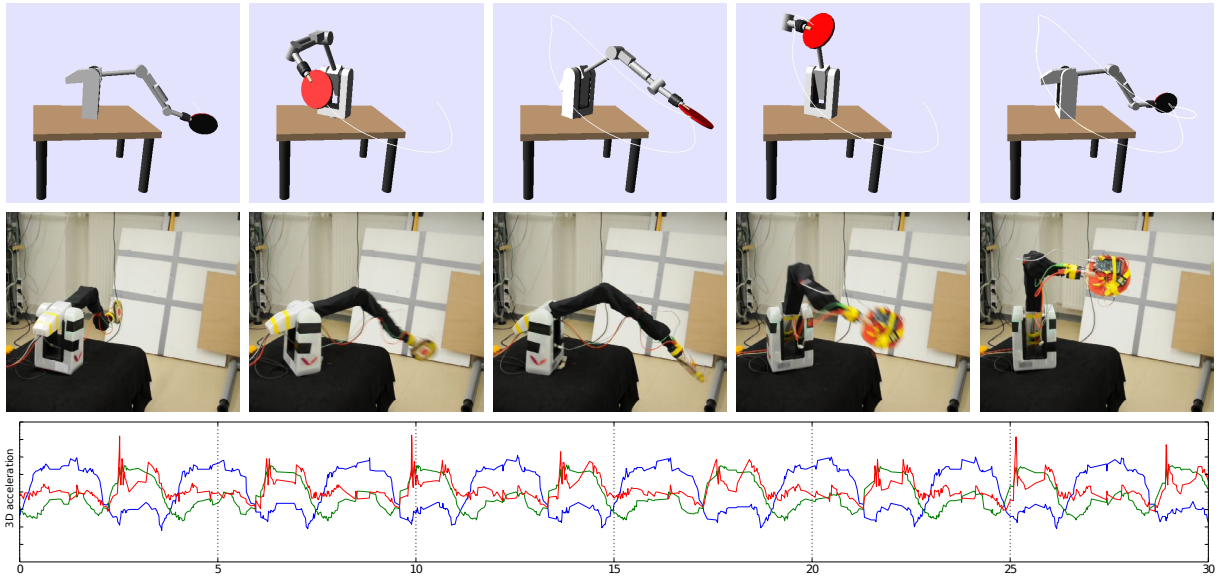


Figure 3.4: The BioRob arm was used to perform a set of motions for benchmarking purposes in a reproducible way. This way, multiple test cases with different firmware settings and hardware configurations can be compared. The simulation (top) depicts the programmed trace of one of the motions, while the photos (middle) show the sensor and measurement hardware attached to the platform. The acceleration data obtained by our data logger reveals the repetitive execution of this particular motion (bottom).

precise motions for a specified amount of time, or repeatedly for a given number. The motion can be programmed by either by manually defining the motion trajectory in space, or by providing the angular configuration of the joints and letting the robot compute the cheapest trajectory, whereby the latter approach was chosen for our evaluation. The robot was programmed by Hany Abdulsamad.

Figure 3.4 depicts the simulation of the motion trajectory as well as photos from the actual deployment, where the sensor and the measurement hardware was attached to the mounting platform on the robot arm. This way we were able to obtain the voltage measurements over the resistor from the Arduino platform, as well as the acceleration data from the sensor node.

Since the robot free time was limited, and as a research prototype it is not designed for long-term operation, we were able to use it for short-term recording of current consumption in different configurations.

3.3.3 Long-term Current Consumption Measurement

While the robot arm is a great benchmark utility for comparing different combinations and parameters, a real-world experiment is impossible to model. In order to access how long a sensor would last on a single battery charge in real-world conditions, multiple sensors were deployed to be worn at the human wrist continuously for full 10 days.

In this experiment, using the aforementioned hardware setup to obtain detailed current consumption figures is not feasible. Instead, the consumption is computed via the battery capacitance difference: Obtaining the capacity of the full battery before the

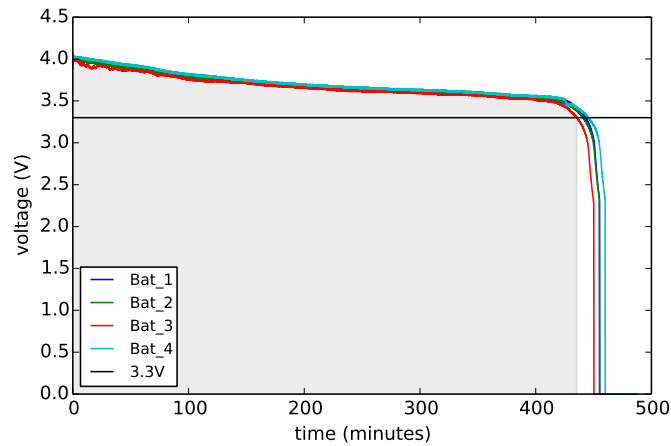


Figure 3.5: Determining the capacity of four fully charged batteries via a $110\ \Omega$ resistor. The area under the curve up to the nominal voltage of 3.3 V is to be considered (highlighted in gray for Bat_3). Similarly, we can obtain the remaining capacity of a battery that was powering a sensor in a deployment. The difference between the measured capacities before and after a deployment will amount to the electrical charge consumed by the sensor.

deployment (Figure 3.5), and the remaining capacity after these 10 days, we are able to compute the average current drain over the deployment time frame. Based on the acceleration data obtained and the overall activity intensity during this period, we can estimate how long a sensor would last when deployed for example for our psychiatric monitoring scenario.

Furthermore, attaching a sensor to a constantly moving or vibrating device will allow to evaluate the runtime characteristic of the sensor nodes on a single battery charge, and access the quality of manufacturing of the sensor or the impact of different configuration settings.

3.4 Evaluation

During the design of our prototype, following design choices were found to have a high impact on the low-power operation of the system, which will be evaluated extensively in this section:

1. Which component should control the sampling, the microcontroller or the accelerometer itself?
2. There are a multitude of microSD cards available on the market, does it matter which to use?
3. The OLED display is necessary for certain applications; how much does it cost with regard to power consumption?
4. How long will the prototype last on a single battery charge with a sampling rate of 100 Hz?

Test	Sensor	Sampling	Frequency	Low-power mode	RLE	microSD card
1	Basic	PIC 10 ms	100 Hz	normal	2	Transcend 1Gb
2	OLED	PIC 10 ms	100 Hz	normal	2	Transcend 1Gb
3	Basic	FIFO	100 Hz	normal	2	Transcend 1Gb
4	Basic	FIFO	100 Hz	low-power	2	Transcend 1Gb
5	Basic	FIFO	100 Hz	low/auto	2	Transcend 1Gb
6	Basic	FIFO	50 Hz	low/auto	2	Transcend 1Gb
7	Basic	FIFO	25 Hz	low/auto	2	Transcend 1Gb
8	Basic	FIFO	100 Hz	normal	2	SanDisk 2Gb
9	Basic	FIFO	100 Hz	normal	2	Transcend 2Gb
10	Basic	FIFO	100 Hz	normal	2	SwissBit 1Gb
11	Basic	FIFO	100 Hz	low/auto	0	Transcend 1Gb

Table 3.1: Overview on the different test cases, varying different parameters, such as the sampling method and frequency, low-power modes, run-length encoding, and the microSD card.

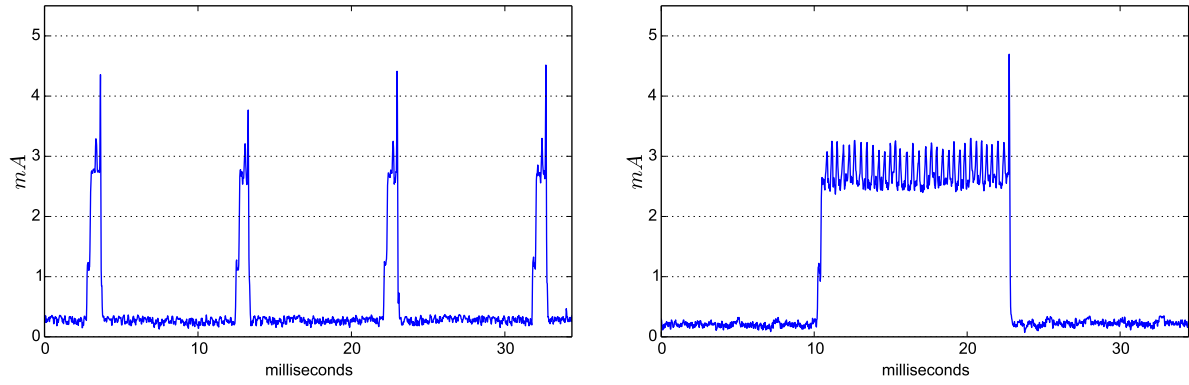
To investigate these questions thoroughly, a number of short tests were performed to obtain current drain figures with the setup presented before, whereby settings and components have been varied. We considered in our investigation microcontroller vs. accelerometer FIFO sampling, different data compression settings, three different low-power modes, four different flash cards, the effect of using the OLED display as well as the impact of adjusting the sampling frequency. These test cases are summarized in Table 3.1.

A second test aimed at measuring total current consumption for a long-term deployment, as one would expect in real human activity recognition scenarios. For that, we deployed two sensors that have been worn continuously day and night for full 10 days. By measuring the capacity of the full battery before the test and after the deployment, the current consumption for this time frame can be computed.

3.4.1 Which Component Should Control the Sampling?

In most sensor unit implementations where sensor data need to be acquired at equidistant intervals, the micro controller is commonly the unit that times and polls for new data from the sensor. Such typical behavior for a 3D MEMS accelerometer is depicted in Figure 3.6a: every couple of milliseconds (10 in this case, due to sampling at 100 Hz), the sensor unit is woken up from a low-power sleep mode via a timer interrupt, to communicate with the accelerometer (via SPI) and acquire a new value tuple (consisting of the x, y, and z axis). This causes every 10 milliseconds a small peak in power consumption, taking about a millisecond.

Many recent MEMS accelerometer chips come with a large set of digital support functions, however, including an operation mode which lets the accelerometer do the acquisition of new 3D acceleration samples for storage in a local FIFO buffer. For the ADXL345 used on our prototype, this buffer holds 32 samples, which means filling a buffer takes a bit more than 300 milliseconds for our target sampling rate of 100 Hz. Figure 3.6b shows the typical current draw pattern in such a case. Additionally, it is pos-



- (a) Typical behavior when sampling from the microcontroller every 10 ms, causing the small peaks of approx. 1 ms width. (b) Utilizing the accelerometer's FIFO results in communication bursts, where 32 acceleration samples are transmitted at a time.

Figure 3.6: Current consumption traces showcasing the two sampling methods: (a) microcontroller polling for samples every 10 milliseconds, and (b) accelerometer sampling with its internal FIFO buffer. Note that transmitting 32 samples at a time reduces the overall communication overhead and allows the microcontroller to sleep for longer periods of time, or conduct other computations, if necessary.

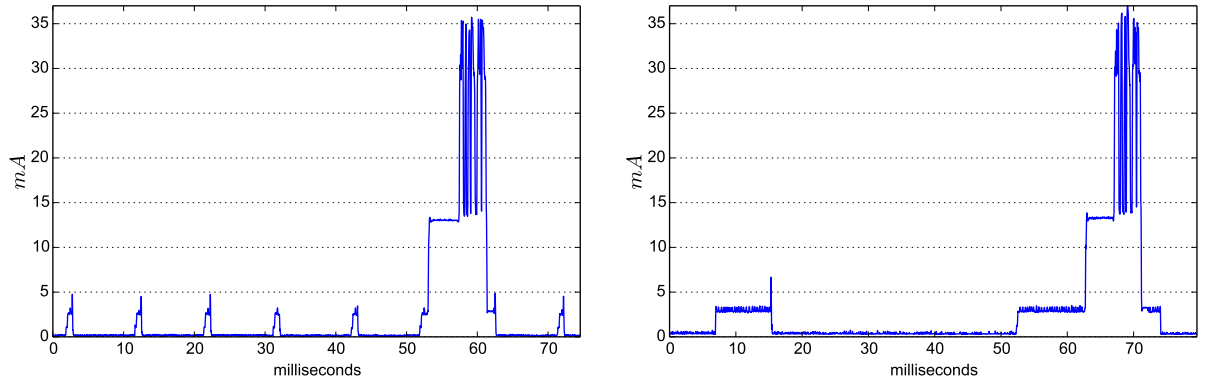
sible to invoke the accelerometer's power-saving functionalities that cause more noise but has a slight effect on the current draw as well.

After obtaining initial current draw figures for sleep mode (0.24 mA), the FIFO or polling communication (2.8 mA), and writing to the microSD card (13.5 mA), a set of small tests lasting 7 minutes was conducted in order to evaluate different operation modes and hardware components (listed in Table 3.1). The result is that FIFO sampling is more efficient: First, while the accelerometer is collecting the sensor samples, the microcontroller can be put to low-power sleep mode to preserve energy. Second, transmitting the 32 values at a time results in a reduced communication overhead and almost the half the current drain (Table 3.2, sampling).

3.4.2 What a Difference an SD Card Makes

After the acquisition of the sensor values, these are typically first stored in a buffer inside random access memory (RAM). Once this buffer in the micro controller's volatile memory is filled up, it needs to be offloaded to permanent storage. Many wearable devices that are used to record fine-grained sensor data nowadays utilize flash memory, either on-board flash chips or replaceable storage cards (e.g., microSD). Such replaceable cards have two main advantages: First, these can be easily bought in large quantities and at appropriate sizes of multiple Gigabytes. Second, broken or full cards can be easily replaced without affecting the lifetime of the sensor device itself.

In our study, writing data to the SD card is the most expensive operation with regard to current consumption (not considering the operation of the OLED display). With our main goal being able to perform activity recognition from sensor data where we specifically rely on subtle detailed information in the signal, there is also the need for



(a) Microcontroller sampling every 10ms, and writing sensor data to flash memory. **(b)** Communication bursts transmit 32 values, and writing sensor data to flash memory.

Figure 3.7: These current consumption traces show the communication between the microcontroller and the accelerometer (small peaks) as well as writing to the microSD card (big peaks). In (a), the writing sensor data to flash memory operation requires approximately 10 milliseconds, thus fitting between two samples (the small tips before and after the big peak). In (b), the communication in bursts is interrupted to offload the data to flash memory.

high-frequent sampling (at 100 Hz). Obviously, storing the raw sensor data will result in lots of writing operations, impacting the lifetime of the sensor. Carefully designing and implementing the logging routine yields a huge power efficiency potential.

Figure 3.7 shows examples of current consumption traces for sensor polling and burst communication, along with the writing of sensor data to the microSD card.

One of the possible approaches to reduce the amount of write operations is to compress sensor data on-line in the microcontroller’s RAM, before storing it to the microSD card. For our sensor device, we use run-length encoding (K-RLE) (Capo-Chichi *et al.*, 2009), which is a very common and widely used method to compress data, with a variable threshold $K > 0$, already as mentioned in the related work section. In our case, the two advantages of K-RLE are: (1) it compresses identical sensor values and preserves subsequences with a varying signal, and (2) it can be used to filter out noise in the signal, thus performing very efficiently on flat sensor data, when correctly choosing the threshold. Choosing $K = 2$ over $K = 0$ for run-length encoding in our tests resulted in a significant reduction of write to flash operations from approximately 23 to 6% of overall consumption (cf. Table 3.2, tests 5 and 11).

Considering our data logger, the study revealed that is also mandatory to carefully chose appropriate microSD cards. To show their impact on the overall current consumption, we considered four microSD flash cards from three manufacturers, namely Transcend, Sandisk and SwissBit, with capacities of 1 GB and 2 GB.

Figure 3.8 shows the findings regarding these cards, with an unexpected result: the 2 GB Transcend card turned out to consume almost three times the current of its 1 GB version or the 2 GB card by Sandisk. The 1 GB SwissBit card has a low plateau current drain of 2.8-3 mA, matching the level of microcontroller and accelerometer communications, and a very short peak of approximately 45 mA lasting 1 millisecond. This is also reflected in the overall consumption figures of the tests 3, 8, 9, and 10 in

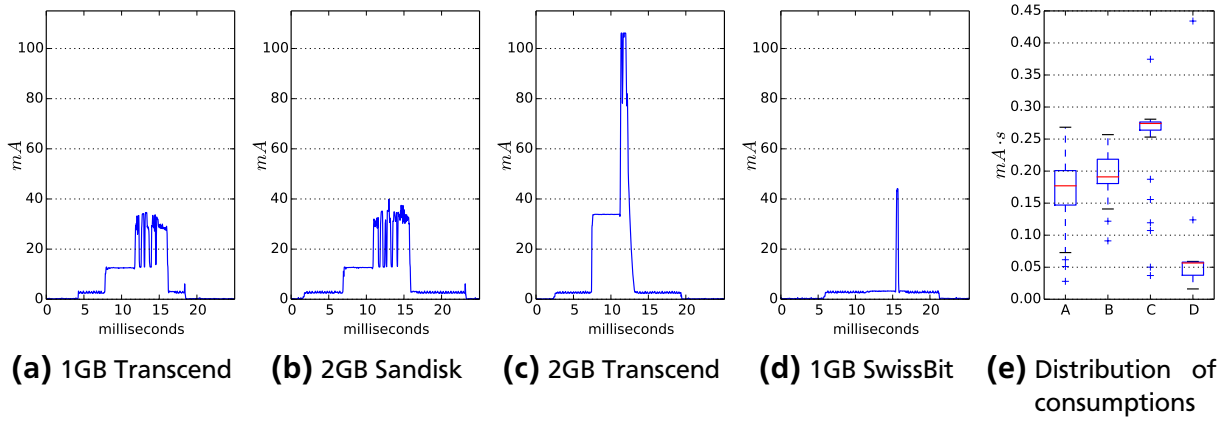


Figure 3.8: Current consumption traces of the four microSD cards used in the experiment. The 1GB Transcend and 2GB Sandisk turn out to be very similar. The 2GB Transcend card (c) has a much higher peak consumption, while requiring few milliseconds less to finish its task. The 1GB SwissBit card (d) has a significantly lower power consumption, at a much higher purchasing price. Computing the area under the curve results in total amount of current spent for the writing-to-flash-memory operation. After extracting these characteristic peaks from the obtained voltage dataset, the distribution of per peak current consumption is shown in sub-figure (e).

Table 3.2. Hereby it is necessary to note that the SwissBit card contains single-level cell flash memory, whereas the other cards have multi-level cell flash, and therefore comes at a much higher price (10 fold), but yields advantage with regard to power consumption. The total current drain of the writing operation (namely the area under the curve) is therefore much lower than of the other microSD cards (Figure 3.8e), making this more expensive card much more preferable in this comparison.

The conclusion of this evaluation is that cheap consumer cards need much more thoroughly testing before being used in such long-term deployments. Just relying on the data sheets with a given peak current drain figure only is not sufficient. What matters is the actual current consumption trace that will reveal the details of this essential hardware component. Obviously, choosing industry-oriented single-level cell flash memory has a huge advantage of a very low power consumption, at a high monetary cost.

3.4.3 The (Battery) Cost of an OLED Display

Our prototype is equipped with an OLED display, which is programmed to show current time and date for a few seconds whenever the user double-taps the wrist-worn prototype. This was particularly a requirement for several long-term trials, in which many users reported unwilling to wear a unit on the wrist that would look unfamiliar enough to raise questions. By making it look and function as a wrist-watch with the addition of the display, acceptance was much higher. However, this component comes at a higher size and production cost for the entire prototype, and undoubtedly at an impact on power consumption. Figure 3.9 shows a subsequence of the OLED display current consumption just before it was turned off by the double-tap from the user.

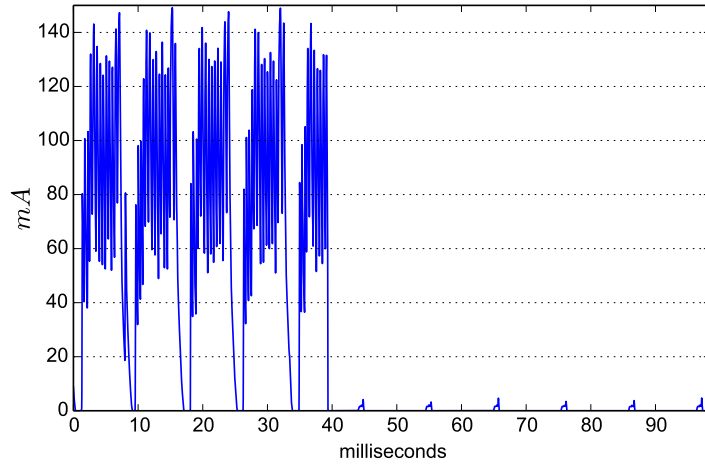


Figure 3.9: Current consumption trace of a data logger with the OLED display. Using the OLED display results in a high current drain, represented by the high peaks on the left side. After the OLED display has been turned off by the user (double-tap feature), 10ms polling from the microcontroller becomes visible.

The OLED display requires 3.3 V power supply for the integrated display driver, which nicely fits our data logger design. On the other hand, the OLED display also requires an additional supply of 12 V for its back-light. To achieve this, a step-up circuit is necessary, consisting of multiple additional components. The drawback of this approach is the reciprocal-proportional dependency of voltage and current: to achieve a step-up from 3.3 V to the required 25 mA at 12 V (according to the data sheet), we need to supply at least 90 mA. Our measurements show that due to the step-up circuit, the OLED display and all other hardware components, the consumption peaks reach up to 140 mA at 3.3 V. Current work therefor focuses on a more efficient implementation of the display.

Although the display was turned on only for a short period of time during the test, the consumption of the OLED and the step-up circuitry reaches more than 98% of the total current draw (Table 3.2). With this result, while not representative for real-world usage, it becomes very important to efficiently design the usage of the display interface. This particularly means that the display has to be turned off as often as possible when not needed by the user.

3.4.4 The (Obvious) Impact of Reduced Sampling Frequency

Varying the sampling frequency has a direct impact on the amount of sensor data generated by the accelerometer unit. While we aim at high sampling rate of 100 Hz to capture full detail of human motion (as well as vibrations), lowering the sampling rates can still be tolerable for various applications. For example, for sleep monitoring and actigraphy scenarios, raw sensor data is not exceedingly critical and is often abstracted by windowed variance or activity counts, whereby the window or epoch size is generally relatively large (in the range of minutes).

With that, we face a trade-off between data granularity and life time of the sensor, which directly impacts the usability aspect. Reducing the sampling rate from original 100 to 50 or 25 Hz results in a proportional decrease the transfer of samples from

Test	Total	Writes to SD		Transfer of Samples		Display	
	mAs	mAs	%	mAs	%	mAs	%
1	130.72	4.76	3.6	63.65	48.7	347.28	98.1
2	353.93	1.74	0.5	4.64	1.3		
3	93.40	5.78	6.2	29.22	31.3		
4	72.78	7.40	10.2	26.15	35.9		
5	84.68	6.19	7.3	28.65	33.8		
6	61.37	3.84	6.3	13.64	22.2		
7	46.80	2.12	4.5	7.28	15.6		
8	103.29	13.21	12.8	29.26	28.3		
9	84.85	7.41	8.7	28.33	33.4		
10	103.40	1.44	1.4	30.48	29.5		
11	95.71	22.28	23.3	26.44	27.6		

Table 3.2: Comparison of the electric charge consumed in total, and for the three main operations: (1) writing to microSD flash memory, (2) transfer of acceleration samples from the accelerometer to the microcontroller by polling or bursts, (3) and the operation of the OLED display. The electric charge is obtained by computing the area under the current drain curve.

the accelerometer to the microcontroller (cf. tests 5, 6, and 7 in Table 3.2), as well as in the amount of write operations to the microSD card. While the sensor node will still consume current for basic operation (microcontroller computation, sampling by the accelerometer, real-time clock, and the loss in the circuits) on a level of approximately 0.24 mA, the overall consumption is reduced.

Thus, a reduced sampling rate will significantly improve the life-time of the sensor, which was also confirmed by multiple long-term tests. Sensors, configured for a sampling rates of 100 as well as 50 Hz were worn continuously for 14 days at the wrist. At 100 Hz, the sensors were able to obtain sensor data for up to 11 days, before the batteries were drained completely. With a sampling rate of 50 Hz, even after 14 days the batteries were not completely drained, and the sensors would have been able to log for additional couple of days.

3.4.5 The 10-day Deployment Test

The evaluations presented before have focused on the impact of different settings and components on the current consumption of the prototype data logger. Aiming at deploying the data logger outside laboratory conditions at the wrist of various users, in order to obtain continuous day and night human motion data, an evaluation is required that considers current consumption over a comparable time frame.

For this, two sensors have been deployed for capturing human motion for a time frame of 10 days. The sensors were worn continuously day and night, and were allowed to be only taken off during showering or bathing. The configuration for these sensors was as follows: 100 Hz sampling through accelerometer's FIFO, low-power modes fully

enabled, RLE with $K = 2$, and the inexpensive Transcend 1 GB microSD card. The OLED display was not used at all during the logging period.

The overall current drain of the sensor was obtained over the capacity change from a fully charged battery before the deployment to the capacity state after the 10 days. The delta of these values, divided by the time frame, provides us with an average current drain figure, and with that allows to estimate runtime for this particular sensor-battery pair. For example, a sensor running with the configuration above for 10 days of logging drained about 93 mAh from the battery.

With that, one day of operation approximately drains 9.3 mAh from the battery. With a nominal battery capacity of 180 mAh, the sensor node should theoretically last more than 19 days.

Experience, on the other hand, shows that this estimated runtime is not reached. The reason for this are the physical characteristics of the flash memory: First, the high peak consumption of the microSD flash card (approximately 36 mA, cf. Figure 3.8a), and secondly, its relatively high operating voltage of 3.3 V with very little deviation tolerance. While the microcontroller, the accelerometer sensor and other components on the sensor board can tolerate lower operating voltage, writing to flash memory will fail as soon as the battery voltage drops below 3.2 V. With this constraint in mind, the runtime of a sensor node with the given setup accounts for up to 14 days.

3.5 Conclusions

This chapter presented the experience gathered in designing and developing a wrist-worn and low-power activity logging unit, which is able to record 3-dimensional acceleration data at a sampling rate of 100 Hertz for two weeks on one battery charge. Using the form-factor of a wrist-watch to safeguard user acceptance, we specifically focused on the choices and parts of our prototype that have the biggest impact on energy consumption of the whole unit. After presenting the details on how we obtained our measurements using an off-the-shelf low-current acquisition setup, we contribute with these findings in particular:

- Accelerometer-based sampling has shown to result in slightly better energy figures than micro controller-based sampling, requiring shorter idle times between samples.
- The choice of microSD card manufacturer and size showed strong variations on the energy footprint of the whole unit, with a potentially large influence on the battery usage. The current draw for some cards consistently reaches 100 mA for storing data, whereas for others 35 mA was measured. A SwissBit 1 GB card turned out to have the lowest current consumption for write operations.
- The OLED display, though appreciated by the users wearing our prototype, has an enormous impact on energy consumption. This adds importance to any mechanisms that turn off the display whenever it is not needed by the user.
- The 10-day test has shown that the sensors have consumed 93 mAh on average. Taking the microSD flash requirement for an operational voltage of 3.3 V into

account, a rechargeable battery with a capacity of 180 mAh is able to power the sensor for approximately 14 days.

The following chapters present the two aforementioned application scenarios in the activity recognition and railway monitoring domains. The data sets used in the corresponding studies were obtained with the current (or earlier versions) of the data logger, sampled at 100 Hertz.

4 Efficient Data Abstraction with Linear Segments

In the previous chapter we have presented an accelerometer data logger which can be deployed for long time spans to capture accelerations, vibrations and orientation in space. This chapter presents an efficient piecewise linear abstraction technique for the output of such units, namely accelerometer time series, with the main focus of reducing the amount of data and at the same time preserving the shape of the signal.

4.1 Motivation

Abstracting raw sensor data can significantly reduce the amount that has to be processed in later steps, thus speeding up the analysis process. Preserving the original signal in its abstraction is often mandatory, for example when the data needs to be visualized for interactive annotation purposes or detailed visual inspection. Performing early data abstraction with efficient abstraction algorithms directly on the sensor devices would allow compression of streaming sensor samples at its source, resulting in much less data to be stored on the sensor, thus preserving limited energy resources and extending the lifetime of the sensor.

In long-term activity recognition with wearable acceleration sensors, where human motion is recorded over long stretches of time, large sets of inertial sensor data have to be analyzed. These data sets contain physical actions of the sensor's wearer that have been captured continuously for multiple weeks or months. Striving to capture human motion as detailed as possible, the sensors are generally sampled at relatively high frequencies resulting in huge amounts of data, which consequently often burden post-analysis of the recorded data.

To tackle this computational problem, this chapter proposes a new data abstraction method that approximates the recorded accelerometer time series by piecewise linear segments and thus retains the original shape (Figure 4.1), that is fast on large data sets and also well-suited for human motion data. Moreover, it is efficient enough to be implemented directly on a sensor node with only few adaptations to meet the embedded platform characteristics, such that the sensor data can be abstracted in an on-line

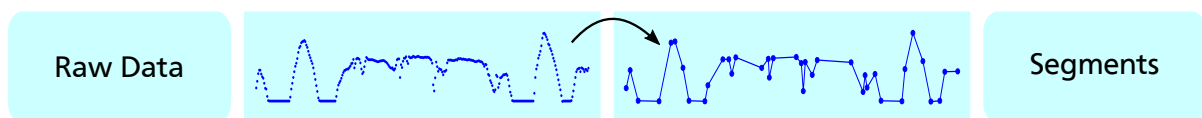


Figure 4.1: Linear segments as shape-preserving features. The shape of the signal is being preserved while the amount of raw sensor data is significantly reduced. In activity recognition, this allows much faster pattern patching.

fashion. Relying on the shape of the signal as a feature for pattern matching purposes, using the abstraction of the time series allows for much faster matching, since the computation is performed on significantly less data points.

The following sections will motivate through a specific human activity monitoring scenario our version of the abstraction algorithm to approximate human inertial data as efficiently as possible, as well as a novel pattern matching technique for finding similar subsequences in large activity logs. We show that our proposed algorithms are faster on human acceleration streams than the traditional ones while being comparable in accuracy to spot similar actions, benefiting post-analysis of human activity data. Additionally, we investigate in an experimental study the advantages of the on-line approximation directly on a sensor node.

4.2 Application Scenario: Interactive Analysis of Human Motion Data

As motivation for the work, we focus on the challenge that lies in the interactive analysis of large amounts of recorded high-frequency inertial data, where users or researchers can view the activity log as a time series, zoom into interesting areas, and select subsequences that exhibit characteristic patterns for particular activities. Figure 4.2 illustrates a prototype of such an application, where the selecting of subsequences in the time series results in the system finding closest matches elsewhere in the data. This would be beneficial for off-line annotation of the data, identification of motion gestures, and detailed comparison of similar activities.

As presented in Chapter 2, previous work has applied a large variety of approximations and features to accelerometer data, such as mean and variance, Fourier coefficients, wavelet matches (Rajpoot and Masood, 2005), and symbolic approaches (Shieh and Keogh, 2008). Approximation techniques that retain detailed visual features from the original data such as Piecewise Linear Approximation of wearable inertial data has not received much attention, apart from (Amft *et al.*, 2005) in which the authors use SWAB to segment out gestures. The analysis techniques on long-term data sets have mostly been limited to resource-hungry and off-line prototypes or systems based on machine learning algorithms such as boosting, topic models (Huynh *et al.*, 2008), and hidden Markov models. Figure 4.3 shows part of such an activity data set of 24/7 recording, which includes sensor data from an accelerometer and a light sensor, along with annotation labels for different activities. The bottom plot shows a more detailed view on the motion patterns in the sensor data while the subject was walking.

Inspired by data mining research, e.g., (Morinaka *et al.*, 2001; Keogh *et al.*, 2001b), we present modifications of algorithms to approximate long-term human acceleration data efficiently, and quickly find matches. These modified algorithms can be applied to activity classification, as will be shown later, but here we focus on *query by example* of large activity data sets. Therefore, the proposed methods allows a user to select a subsequence in the data, for which the system returns other subsequences that match the query in structure, as shown in Figure 4.2. This can be used to find other occurrences in the data where the wearer performed such a characteristic action ("*when else did this walking pattern occur?*"), or to identify stretches of time for an activity ("*how long does this walking pattern go on?*").

The main properties of the presented approach are the following:

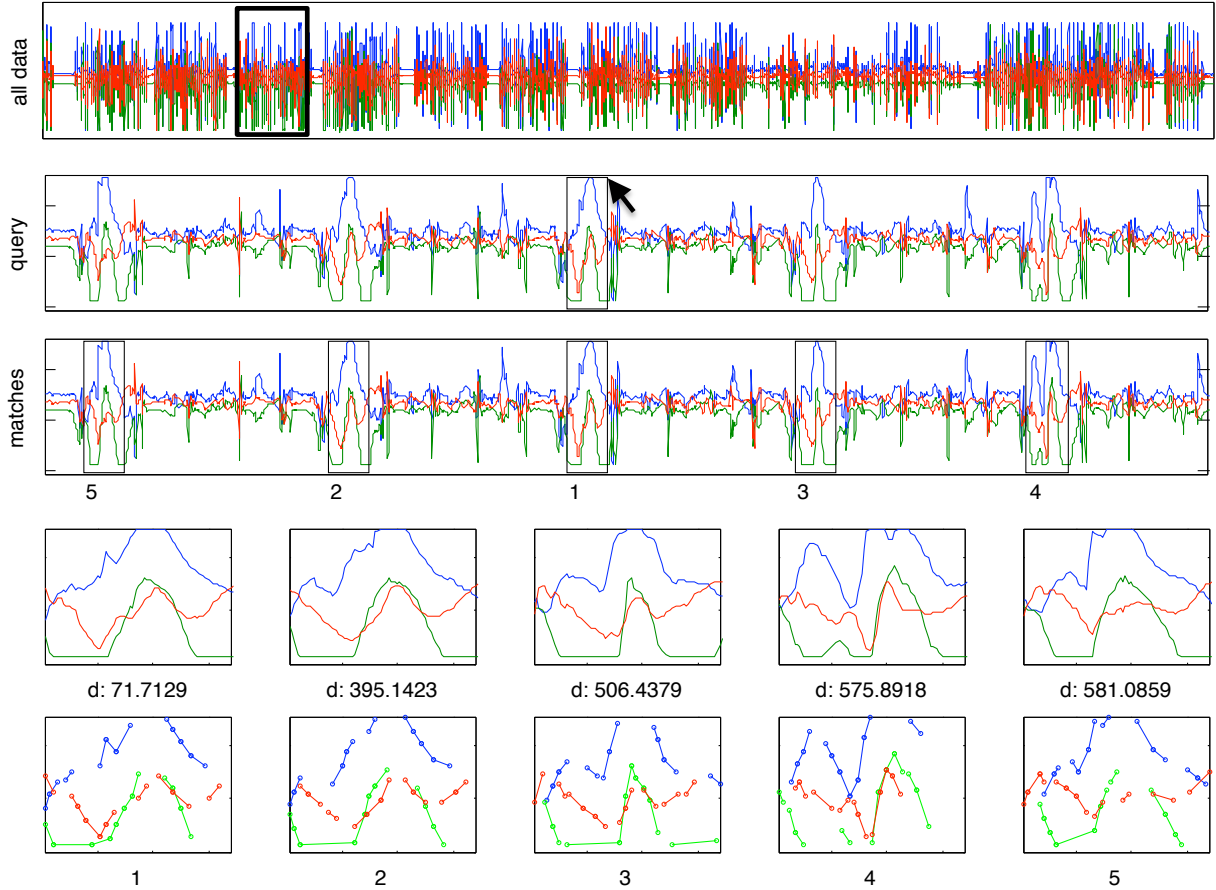


Figure 4.2: A subsequence from a large 3D acceleration data set (top plot) is used to query (second plot) for matching subsequences where a similar motion pattern occurred (third plot). The 10 bottom plots show the closest matching subsequences to the selected query in detail (top row: the original raw data, bottom row: the approximations).

- **Fast on large data sets.** Multi-day acceleration data contains tens of millions of samples typically, needing algorithms that allow fast searching through all data.
- **Human acceleration-specific.** As the time series is known to be from human activities, optimizations can be made with respect to the traditional algorithms.
- **Reduces logging on the sensor.** The approximation can be implemented on a low-power sensing module so that less data is stored, thus needing less power.

In the following we present a two-tier set of algorithms: one that **approximates** the inertial data so that it can be stored in less memory and be processed faster, and one that **matches** subsequences of these approximations to find similarities within a large time series of human activity data. To achieve our goal, several requirements need to be fulfilled: First, as both researchers and subjects have proven to be able to browse time series plots of inertial data well, as mentioned by Van Laerhoven *et al.* (2008), the data needs to be approximated so that it can still be represented visually as a time series. Second, the algorithms that do the approximation of data need to be fast enough so that it does not hinder the loading of data, and accurate enough to capture the essence

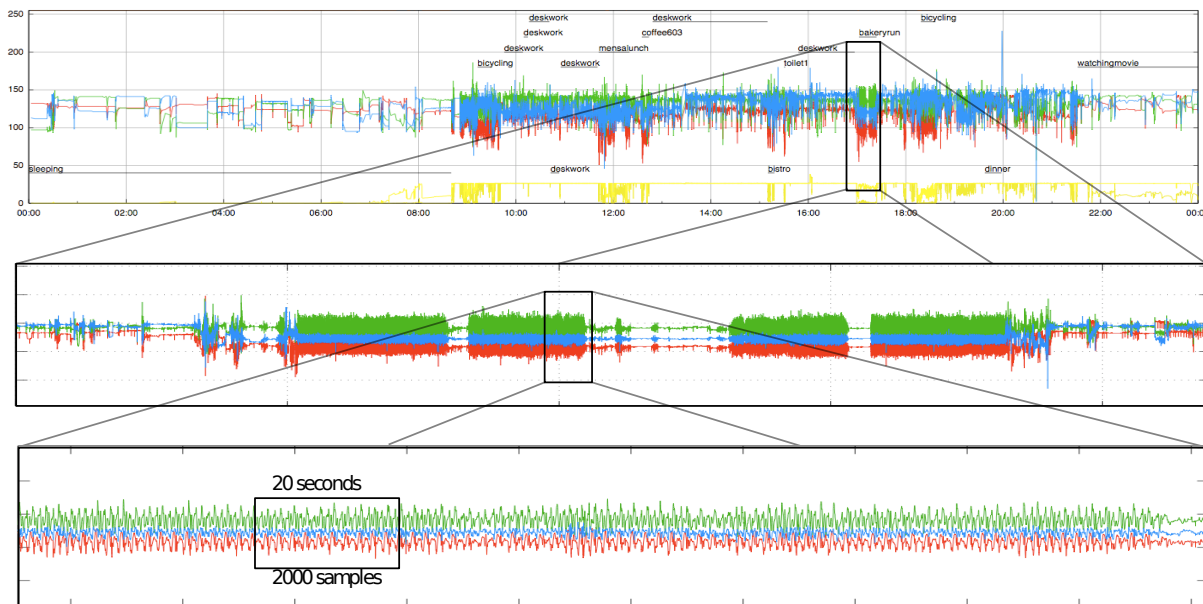


Figure 4.3: Typical time series plots from a long-term activity dataset, obtained from a wrist-worn sensor. Each day has 8.64 million time stamped 3D accelerometer samples, and additionally light and ambient temperature data. Acceleration data is focused on as the most descriptive modality, since it contains both posture (e.g.: the sleeping postures during the night segments) and motion data. The zoomed-in region at the bottom shows acceleration data from the subject walking.

of the motion patterns. Finally, the matching of patterns needs to be fast enough to allow interactive applications, while being accurate enough to do qualitative matching.

4.3 Approximation of Accelerometer Data

This section introduces the algorithm to approximate human acceleration time series efficiently, so that it is represented by a smaller amount of data without losing the intrinsic nature of the underlying activity.

Key to our approach is the approximation of a time series of human acceleration data, into a representation of linear segments that is efficient to manipulate and faster to process than the raw sensor data. The linear segments can be visualized in an identical way to the original data in a time series plot, while the number of data points is significantly reduced.

Our approach presented in this section is based on the SWAB algorithm (Keogh *et al.*, 2001b), and is an adaption of this algorithm specifically for physical activity data from accelerometer sensors. The next sub-sections will present the original and our modification in detail.

4.3.1 SWAB

The original SWAB algorithm works by approximating the time series by well-chosen linear segments that are closer to the data than the on-line Sliding Windows method,

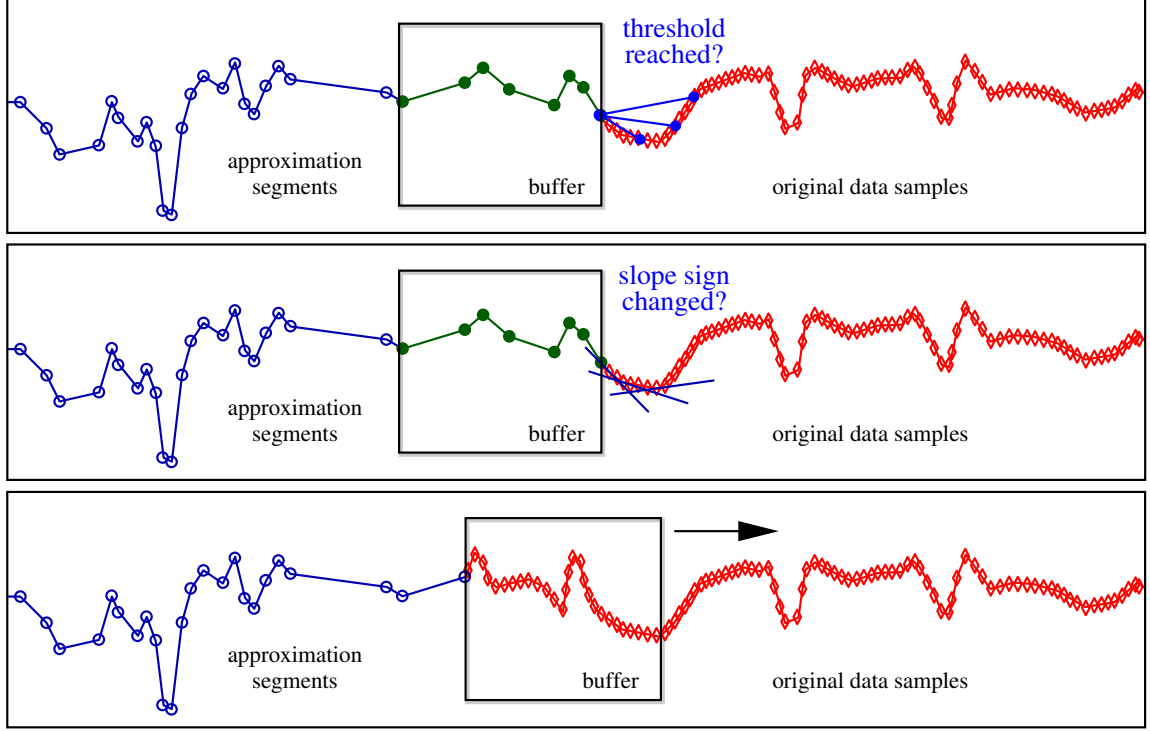


Figure 4.4: The original SWAB algorithm approximates a time series to a set of segments, with a sliding buffer in which bottom-up approximation is performed. Our adaptation replaces the computationally expensive Sliding Windows phase (top) by moving the buffer up to the data point where the slope changes sign (middle). After sliding the buffer (bottom), the segmentation process starts over. Note that the buffer size is variable within certain bounds and adapts automatically due to the shape of the signal.

while being still an on-line approach. Figure 4.4 sketches the basic operation of SWAB: A buffer window is slid over the original time series, in which Bottom-Up segmentation is performed. The left-most segment is then produced as the next approximated segment, while the buffer window is moved to the right, to the next original data point for which the Sliding Windows approximation cost overruns a threshold.

More precisely if the segment between the i th and j th data points, x_i and x_j respectively, is called S , the cost of approximation of the subsequence $(x_i, x_{i+1}, \dots, x_j)$ by S is calculated by

$$c(x_i..x_j, S) = \sqrt{\sum_{n=i}^j \left(x_n - \left(x_i + (n-i) * \frac{x_j - x_i}{j-i} \right) \right)^2},$$

which is then done for every new data point x_j until the cost c overruns the cost threshold. In that case, the new buffer is extended to x_{j-1} and the next approximation segment is searched with the Bottom-Up approach in the buffer.

4.3.2 mSWAB

For the original SWAB algorithm optimizations with regard to the processing speed are possible for particular data, by for instance incrementing the sliding window with multiple samples instead of one, which showed beneficial in case of electrocardiography (ECG) data. SWAB's standard version moves a sliding window, recalculating an approximation cost and matching it to a threshold for every additional sample of raw data.

Our adaptation, as shown in Figure 4.4, exploits the property of accelerometer data, which tends to heavily fluctuate by producing characteristic peaks in the time series, and instead moves the window on to the next data point when the slope's sign changes between positive and negative, or zero. This means that instead of having to iteratively calculate c , one simply has to calculate the slope between adjacent data points x_j and x_{j+1} and stop when the signum function changes value:

$$\text{sgn}(x_j - x_{j-1}) \neq \text{sgn}(x_{j+1} - x_j)$$

This speeds up the process as it requires a single test per sample ($O(n)$ with n the samples the buffer is shifted over), instead of recalculating costs over the segment ($O(n^2)$ regardless whether sum of squares or the L_∞ norm is used for the cost calculation). Although the Bottom-Up part of SWAB remains still costly, substituting the Sliding Windows approach leads to a significant effect when the accelerometer data is sampled at a high frequency or in constant subsequences (i.e., when no movement occurs). The latter occurs very frequently, especially in long-term data which include resting and sleeping segments.

A second change to the original algorithm uses a suggestion made by Junker (2005) to merge the last two produced segments if their slope is the same. Listing 4.1 shows the source code, highlighting the differences to the original SWAB algorithm.

Up to now, we have presented the one dimensional version of the algorithm. This technique can be extended to handle multi-dimensional data as well, thus producing a multi-dimensional approximation with linear segments. Depending on the application, a multi-dimensional approximation of the sensor data can be of particular interest. For example, the multi-dimensional version of the algorithm can be used for abstracting human motion in time and 3D space. We have implemented a multi-dimensional version of mSWAB and used it in chapter 6 to abstract human motion and then to detect recurring motion patterns of interest. An example of multi-dimensional approximation of raw inertial data by our mSWAB is shown in Figure 4.5.

To assess the impact of the proposed modification, we compare in the following section the performance of our mSWAB algorithm to other existing techniques on real-world human inertial data.

4.3.3 Evaluating the Approximations

With the modified SWAB at hand, our focus now lies with assessing the efficiency of the approximation algorithm, meaning the execution speed and the quality of the approximation, which we measure by computing the overall approximation error.

Listing 4.1: Here, the original SWAB algorithm abstracting *timeseries* with cost threshold T , has its Sliding Window heuristic modified to increase the algorithm's speed. To create less data, segments with similar slopes are merged.

```
[segs] = mSWABsegs(timeseries, len, T)
win_left=1; win_right=bufsize;
while(1) % while there is new data:
    swabbuf = timeseries[win_left:win_right];
    BUsegs(swabbuf, bufsize, bu_segs, T); % Bottom-Up approx. of buffer
    segs = [segs; bu_segs(2)]; % add left-most segment
    n = size(segs);
    if slope(segs(n)) == slope(segs(n-1)), % merge last segments ...
        merge_last2(segs); n = n-1; % if slope is equal
    end;
    win_left = bu_segs(2).x; % update left border of buffer window
    if (win_right < len), % find right border of buffer window
        i = win_right+1;
        s = sgn(slope(i, i-1));
        while sgn(slope(i, i-1)) == s, % while sign has not changed ...
            i = i+1; % grow buffer window
        end;
        win_right = i; % update right border of buffer window
        bufsize = win_right - win_left;
    else % no more data available ...
        segs = [segs; bu_segs]; % add remaining segments from buffer
        break; % exit endless loop
    end;
end;
```

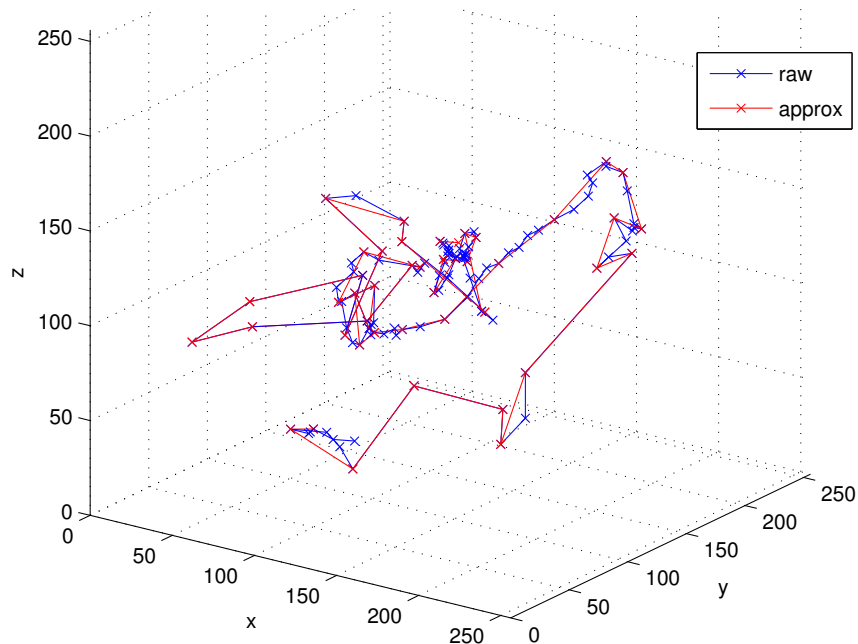


Figure 4.5: An example of raw sensor data approximation with the 3-dimensional version of our mSWAB algorithm.

Abstraction technique	time to reduce 24h of data by			time to process 24h of data
	50%	10%	5%	
Mean and Variance	0.1	0.1	0.1	0.1
Discrete Fourier	219.0	383.0	428.0	318.0
Sliding Windows	2.7	6.7	14.0	188.8
SWAB	8.0	7.7	7.9	36.1
mSWAB	6.8	2.4	2.1	19.2

Table 4.1: Approximation methods and the time it takes to approximate the original raw sensor data. The second, third and fourth columns vary the parameters so that the algorithms produce respectively half, a tenth, and one twentieth of data representing the original raw. The last column shows how many seconds the algorithms need to transform a day worth of data (on average, with best performing parameters).

For this, we conduct an experiment where a time series from a human-worn accelerometer recorded by a wrist-worn accelerometer logger (Van Laerhoven *et al.*, 2008) has to be approximated. The device, a less recent version of the current prototype shown in Figure 3.2, has been used to obtain 3D accelerometer data at a speed of 100 Hz that was stored on a microSD card. Since the data was uploaded approximately every 24 to 48 hours, the resulting data set consists of sequences with about 8.5 to 17 million 3D acceleration samples each.

Several basic approximation techniques are evaluated for execution speed on 36 hours of consecutive accelerometer data, with approximately 13 million 3D samples. For this, we compare our modified SWAB to the original algorithm, along with the Sliding Windows technique, which all belong to the class of Piecewise Linear Approximation (PLA). Additionally we also present the performance of the segmented mean and variance and the Discrete Fourier transform.

As can be seen from a 3D acceleration data example in Figure 4.6, the linear segments produced by the PLA techniques can be visualized relatively easy as a time series plot, showing several characteristics of the original time series. The PLA techniques are represented by Sliding Windows in this figure, while SWAB and mSWAB produce a slightly different approximation. This figure also explains in particular why Piecewise Linear Approximation fits so well in the query by example application presented in this chapter: The segments allow a fast displaying, as well as zooming in and out, in a time series plot of the accelerometer data, with close similarity to the original time series.

In our evaluation, the parameters for each technique were chosen for the following scenarios: The first two vary the parameters of the methods so that they produce either fine-grained or coarse-grained approximations of the original data. The third was chosen so that the approximation methods produced the same amount of approximated data each (in bytes). To guarantee fairness, all algorithms were implemented in C/C++ and executed on a regular desktop computer.

Table 4.1 shows the time (in seconds) it takes for the various algorithms to approximate 24 hours of accelerometer data, averaged over all days in the data set. For the results in the first three columns, the parameters of the respective algorithms are set so

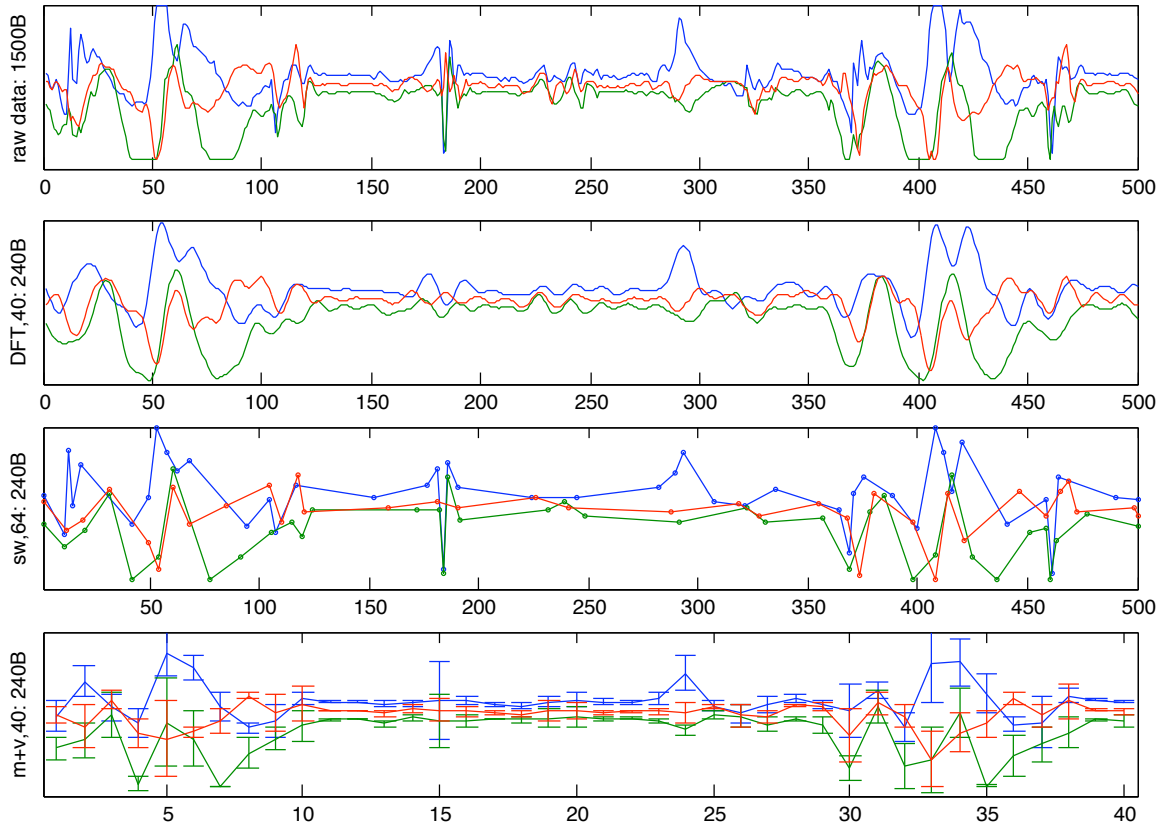


Figure 4.6: Representations of the most popular methods, all approximating the 3D acceleration time series (top plot) within 240 bytes of data. The resulting 240 bytes have been visualized back to a time series representation to illustrate what information is typically retained. The first 40 DFT coefficients are here transformed back with the inverse Fourier transform (2nd plot), the third plot shows the Sliding Windows approximation segments (with threshold 64), and the last plot displays segmented mean and variance. The original as well as our modified SWAB algorithms (not shown here) produce better approximations than the Sliding Windows technique.

that the amounts of data that each algorithm produces are approximately equal. The results show that our modified version of SWAB is indeed faster than regular SWAB, much faster than DFT or Sliding Windows, yet a lot slower than the segmented mean and variance. Implementing mSWAB on a microcontroller is feasible, however, as it does not require floating point arithmetic, nor large data buffers.

Figure 4.7 shows results in detail between SWAB and the proposed mSWAB. There is a tradeoff of speed and approximation error: mSWAB becomes significantly faster from an early threshold T , owing to the difference in implementation of the sliding buffer, however, the error (normalized Euclidean distance over the length of the data set) is much higher for mSWAB.

To get a more deeper insight on how significant these performance differences of mSWAB compared to SWAB is, we have extended the evaluation to verify the following two claims: (1) whether mSWAB does indeed approximate the original accelerometer

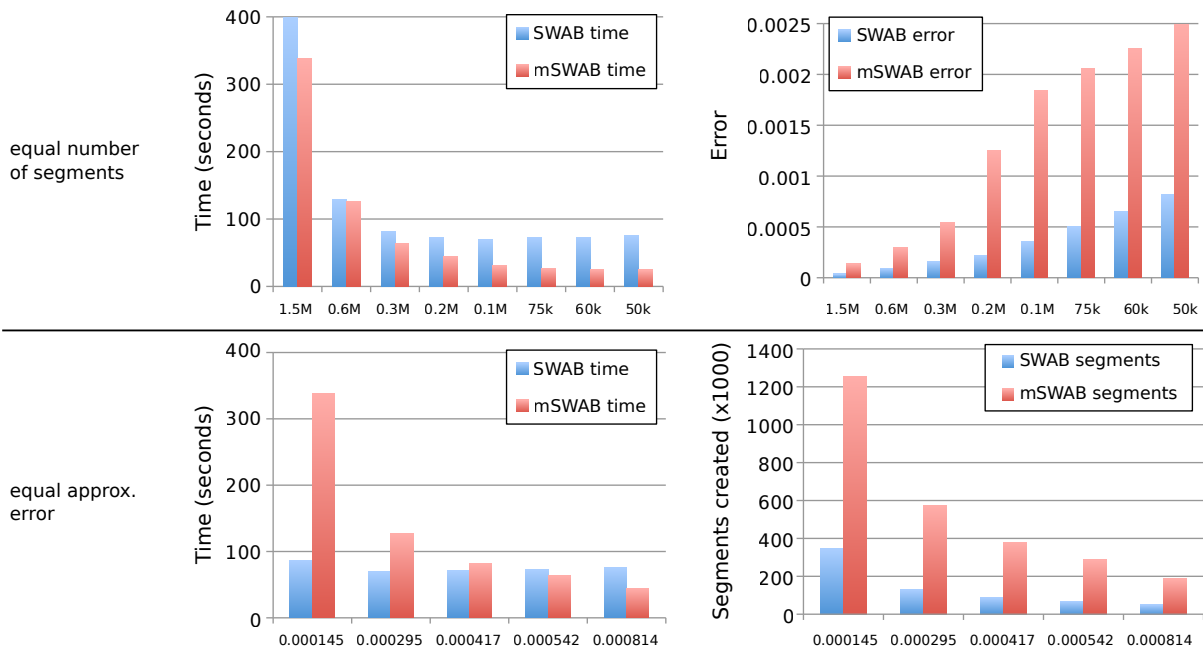


Figure 4.7: A closer look at the results from SWAB and our modified version of SWAB. By varying the threshold and comparing results when the number of segments match (top) or the error between approximation and original data match (bottom). As hypothesized, the mSWAB is faster than SWAB, but at the cost of deviating more from the original data. The bottom plots show that speed is less of an issue to match the error of SWAB and mSWAB, a lot more segments need to be created with mSWAB.

data faster than the generic SWAB, and (2) whether the quality of approximation is indeed comparable to that of SWAB.

In order to obtain meaningful results for our proposed application, three data sets of activity data were used that are similar in structure and configuration to the one depicted in Figure 4.3. Each data set contains the continuous 3D acceleration data from a wrist-worn sensor, and spans between 24 and 48 hours. The subjects that were monitored in these can be characterized as leading a regular life with normal levels of activity. Similarly to the first evaluation, all tested algorithms, namely Sliding Windows, SWAB and mSWAB, were implemented in C++, and compiled with full compiler optimization turned on. The tests were all carried out on the same computer with a CPU clocked at 3.2GHz.

The left plot in Figure 4.8 shows the residual error (i.e., the sum of squares of the vertical differences between original data and approximation segment, for all segments) for the Sliding Windows, the SWAB algorithm, and the proposed mSWAB algorithm, for a cost threshold set between 1 and 50. The initial buffer size was set to 100, which is essentially the amount of raw data streaming in per second. Prior verification has shown that this value works well and contains the recommended 5 to 6 segments mentioned in by Keogh. There is little difference between the performances of SWAB and mSWAB, confirming that the Bottom-Up buffer within the algorithm works identically for both implementations. The results also further confirm those from (Keogh *et al.*, 2001b),

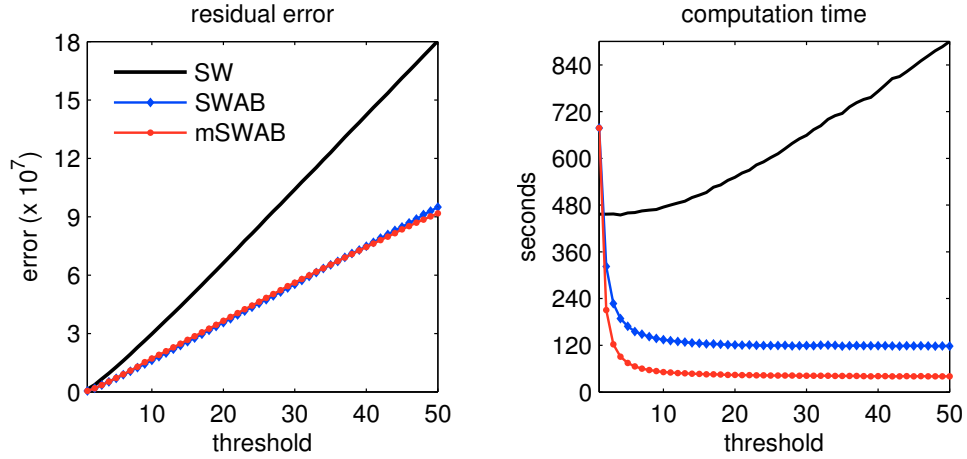


Figure 4.8: mSWAB evaluated against the Sliding Windows and SWAB algorithms on the long-term data. Left: the residual error for a varying cost threshold. Right: the time in seconds needed to approximate the data. Initial buffer size for SWAB and mSWAB was set to 100.

showing that the approximation segments from the Sliding Windows algorithm are further apart from the original data than those of SWAB and mSWAB.

The right plot in Figure 4.8 shows the execution speed in seconds, for the cost threshold of the approximation algorithms between 1 and 50. Identically to the residual error plot, the initial buffer size for SWAB and mSWAB was set to 100. Sliding Windows can be seen to be in the same range for a cost threshold of one, then veering off and steadily increasing as the cost threshold increases. The mSWAB algorithm does indeed display a faster execution speed compared to SWAB, owing to the sliding heuristic of the buffer window: Instead of steadily increasing the segment and re-doing the cost calculation over an increasing set of data points, the change of slope between successive data points is monitored.

The remaining questions are: a) Does this larger approximation error between segments and raw data have any effect on actual matching? and b) Is the algorithm efficient enough to be implemented directly on a wearable data logger with very restricted computational capabilities? To answer these questions, the next sections present a case study each.

4.4 Case Study – Subsequence Matching for Activity Recognition

First, considering the activity recognition scenario, where patterns selected by the user are used to query for similar patterns in sensor data.

After obtaining the approximation of the original inertial sensor data through mSWAB, this section covers the matching of patterns in nearest neighbors-based classification of these approximations. Hereby, we use the well known dynamic time warping technique as a base line. Additionally, as a faster alternative, we introduce a new heuristic that uses Euclidean distances between the K longest segments of a pattern.

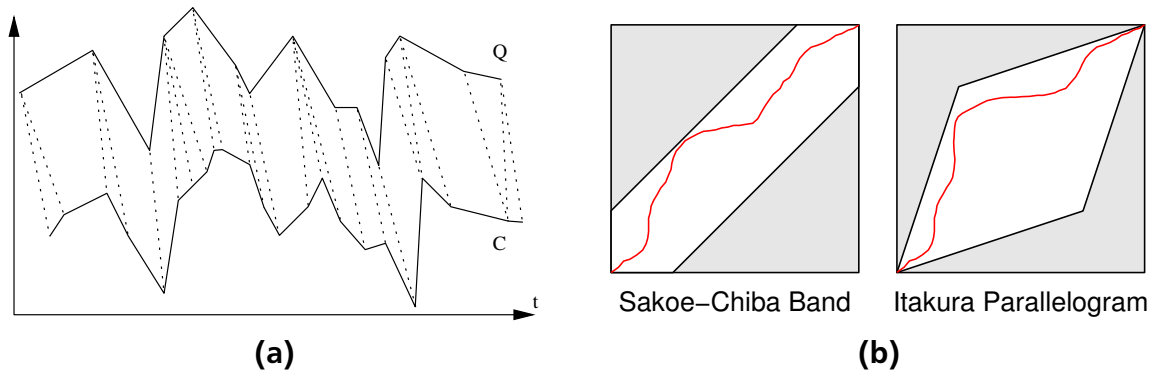


Figure 4.9: a) Matching subsequences Q and C with dynamic time warping (DTW) aligns data points to the optimal counterpart (dotted lines). b) DTW is often bounded, restricting the warping paths by local or global constraints (e.g., "Sakoe-Chiba Band" or "Itakura Parallelogram" are shown as white areas).

4.4.1 Dynamic Time Warping

The distance between two time series of equal length $x = x_1, x_2, \dots, x_n$ and $y = y_1, y_2, \dots, y_n$ can be computed via the Euclidean distance:

$$\text{dist}(x, y) = |x - y| = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}.$$

If the time series lengths are not equal, which is often the case for subsequence pattern matching for the original scenario, the euclidean distance is not usable, since it does not consider appropriate mapping of points. Furthermore, the euclidean distance is a poor similarity measure if the time series are similar, but distorted in the time domain.

Dynamic time warping (DTW) is a widely used technique used in speech recognition, information retrieval and machine learning, to overcome small distortions in time between two time series (Bellman, 1957; Ratanamahatana and Keogh, 2004; Ko *et al.*, 2005; Lemire, 2009; Pham *et al.*, 2010). Given two subsequences, DTW optimally aligns or 'warps' the data points between the two time series (Figure 4.9a) and returns their distance, which then can be used in classifiers as a similarity measure.

To align two time series $Q = q_1, \dots, q_n$ of length n and $C = c_1, \dots, c_m$ of length m with DTW, an n -by- m matrix with squared distances of the time series elements q_i and c_j is created, and an optimal 'warping path' that characterizes the alignment of Q and C and minimizes the warping costs is computed. The warping path cost for distance matrix entry (i, j) can for instance be recursively computed with the distance function

$$\gamma(i, j) = d(i, j) + \min\{\gamma(i-1, j-1), \gamma(i-1, j), \gamma(i, j-1)\}.$$

The general approach is to compute all the squared distances in the matrix and then to choose the minimal continuous path. Unfortunately, this approach is of high time complexity, namely $O(n \cdot m)$. In practice, different local or global constraints can be used to decrease the number of paths that will be computed during alignment process, thus significantly speeding up the calculation. In our implementations we considered

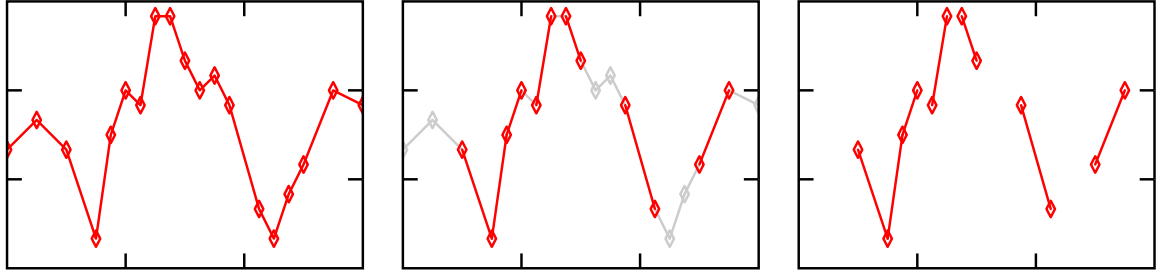


Figure 4.10: Selecting K longest segments for pattern matching, with K in this example being 7. In our query-by-example scenario, for a subsequence, the K longest segments are chosen to be then compared against the time series.

two common bounding techniques: a band and a parallelogram originally proposed by Sakoe and Chiba (1971) and Itakura (1975) respectively, where only paths are considered that lie within certain bounds (Figure 4.9b).

4.4.2 K Longest Segments

Most subsequences of interest tend to have a high number of segments, resulting in slow matching when done with Euclidean matching or dynamic time warping. In order to speed up matching, we propose to limit the subsequences to those segments that are likely to be most descriptive. These are in our data assumed to be the K longest segments per dimension. We argue that this is sensible as the large segments tend to cover either large peaks or large stable regions in the subsequences for our accelerometer data, both of which are important for characterizing motion patterns within physical activities.

For matching, these K longest segments are selected and compared against the segments in the subsequence it is compared to. The distances to the closest matching segments, using Euclidean distance, are then summed to obtain a distance between the two subsequences. When used in our query-by-example scenario, the second subsequence is created from the contents of a sliding window over the entire time series. Thus, closest matching subsequences to a query subsequence can be found.

The choice of the number of segments K greatly affects the speed of the algorithm, as well as the accuracy of approximation. The higher K becomes, the more distinctive the resulting set of segments will be in matching and the more time is needed to find closest matches to all segments.

4.4.3 Evaluation of Matching Methods

To evaluate how accurate the matching works of the DTW and K longest segments methods, we use a data set that contains 15 very similar target classes: For 5 test subjects, three person-specific activities are recorded that are known to be highly challenging in activity recognition: “walking”, “climbing stairs” and “descending stairs”, as shown in Figure 4.11. The data set incorporates fatigue and sensor strap loosening by recording per test subject all activities 5 times in a row and on two different days, resulting in



Figure 4.11: The walking data set consisting of 5 test subjects performing “walking”, “climbing stairs” and “descending stairs” activities. To evaluate our approximation and matching algorithms, we try various approximation and matching parameters combinations, on the challenging 15 classes, where the activities as well as the person need to be discerned.

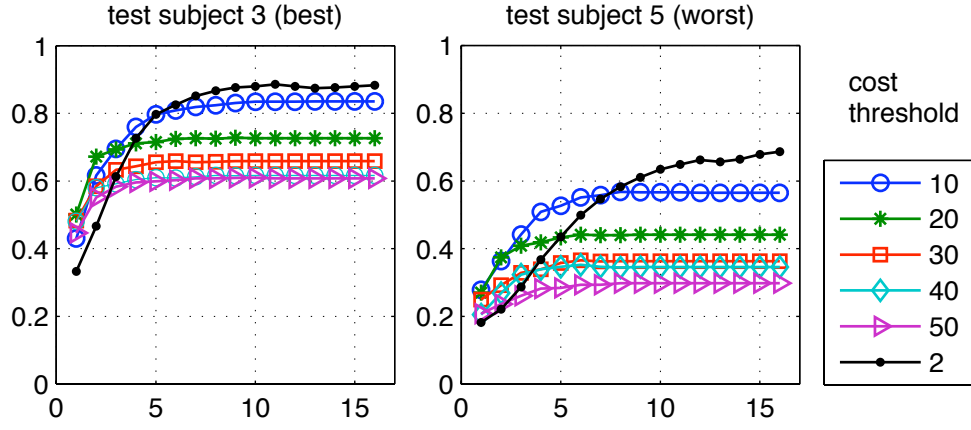
Threshold	DTW accuracy (%)		KLS accuracy (%)		Execution time (sec.)	
	worst	best	worst	best	DTW	KLS
2	63.4	88.0	50.3	85.0	88.6	12.7
10	56.6	83.4	41.7	83.5	15.5	7.5
20	44.1	72.6	31.2	72.6	7.0	3.6
30	36.3	65.8	28.6	63.0	4.0	2.2
40	34.5	61.6	19.5	58.2	2.5	1.5
50	29.7	61.0	12.4	49.1	1.7	1.1

Table 4.2: Comparing the performance of DTW and K longest segments as in Figure 4.12 with accuracy (in %) and execution time (in seconds), using 10 for both SC-band width and K , and varying approximation threshold.

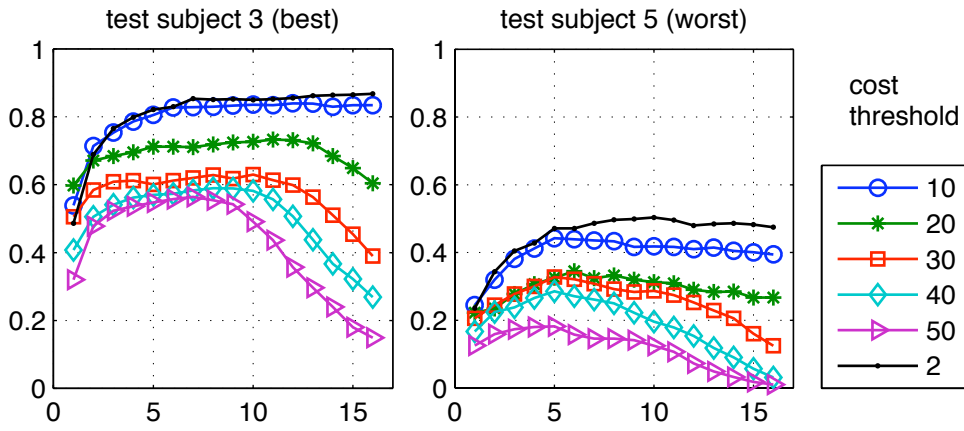
many examples per class with inter-subject deviations. The entire data set consists of about 1.1 million samples, spanning over 2 hours.

The above-described DTW and K longest segments algorithms were used to match and classify via nearest neighbors classification the training part of this data set, using 30-fold cross validation, to the remainder testing part. DTW was used together with a Sakoe-Chiba band being varied from 1 to 16, as was the parameter for K longest segments, K . The target classes were chosen to be as challenging as possible, not only containing the activity but also which person performed the activity. Detections in unlabeled (‘background’) data were counted as false positives.

As illustrated by the best-performing data in the left plots of Figure 4.12, K longest segments tends to equal the performance of DTW for SC band and K of 10, and using a low enough cost threshold. The right plots in Figure 4.12 illustrate that DTW performs in few isolated cases considerably better with lower cost thresholds. DTW however, as can be witnessed in Table 4.2, results exactly then in far longer execution times. It is therefore important in future work to consider data from more test subjects.



(a) Accuracies using DTW with SC-band from 1 to 16 (on the x-axis)



(b) Accuracies using K longest segments with K from 1 to 16 (on the x-axis)

Figure 4.12: Accuracy results for the best and worst test subjects in the data set, approximated with mSWAB for a range of cost thresholds, and classifying with nearest neighbors using: a) Dynamic Time Warping with Sakoe-Chiba band, or b) K longest segments.

4.4.4 Study Conclusions

Long-term activity recognition has, due to advances in miniature sensing techniques, become a field in need for fast machine learning techniques. Recordings of long-term inertial sensing trials tend to be many and large, producing time series which are hard to analyze using conventional classification techniques.

In this study we have presented a piecewise linear approximation algorithm for time series, which based on the SWAB algorithm and was modified to work faster on human acceleration data. The resulting approximation can be plotted in time series plots just as the original data, allowing inspection and annotation of the data. While in our experiments the modification allowed for a significantly faster approximation, it also resulted in a larger error with respect to original data, the impact on the classification based on matching with bounded Dynamic Time Warping, however, resulted in similar performance.

Additionally, we have proposed a matching routine for the approximated data that uses the Euclidean distance between the K longest segments in the approximation. This method has in an empirical study shown to be about twice as fast as bounded Dynamic Time Warping, while resulting in similar accuracies when the value K is chosen well.

The two main contributions for our long-term activity recognition research are: first, an implementation of the approximation algorithm directly on the wearable acceleration logger allows to store segmented approximations rather than the raw data. Second, the fast matching of segments allows efficient query by example data mining: by selecting regions in the time series plot of a data set, matching regions are found that can be used as a tool for post-annotation and -analysis of the data.

In Chapter 6, we will use the mSWAB algorithm as the first early data abstraction step in an activity recognition system that uses learning methodologies that are capable of modeling richer representations of activities, such as models capable of learning (sets of) typical motion subsequences per activity.

Aiming for the proposed on-sensor implementation and on-line approximation of streaming sensor data, the next section presents a study on implementing the mSWAB algorithm on a wireless sensor node with constrained hardware resources,

4.5 Case Study – Embedded mSWAB for WSN Applications

Following the idea of early data abstraction that leads to faster analysis of the recorded time series, such as pattern searching and matching, we argue that it is very useful to create sensor data abstraction directly at the source, i.e. on the sensor node. First, at a cost of additional processing, the amount of data to be stored on a sensor node, or (in case of a wireless sensor network application) to be sent through the network is significantly reduced. To this end, in this chapter investigates the implementation of the proposed mSWAB algorithm on a sensor node.

After implementing the bottom-up approximation as well as the slope sign change part for general purpose personal computers and evaluating their performance on already available off-line data sets, previous experiments indicated that this modification is efficient enough to be implemented on an embedded sensor platform (Van Laerhoven and Berlin, 2009; Van Laerhoven *et al.*, 2009). In this section we will look at the implementation of this algorithm on wireless sensor nodes in general.

4.5.1 emSWAB

The basic functionality of a sensor unit running embedded mSWAB (emSWAB) is as follows: a sensor is being sampled at a specific fixed frequency, producing what we define as raw data values. The new sampled data are forwarded to the emSWAB algorithm that decides whether to store the value to a buffer or, if enough values are available in the buffer already, to run the bottom-up approximation step, thereby producing the next approximating linear segment. This segment is stored to a buffer that eventually will be wirelessly transmitted into the sensor network for further analysis at the base station (see Figure 4.13). Hereby the segments are represented by data points that consist of an index (Δ_i value in time in amount of samples, to the previous data point) and the corresponding sensor value.

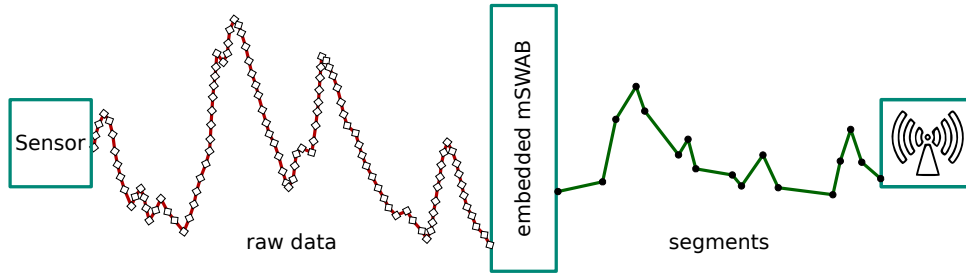


Figure 4.13: Basic functionality of emSWAB on a wireless sensor node. The emSWAB takes a buffer of raw data, computes a bottom-up approximation, and produces the next segment as the final result of the approximation step. The buffer is then filled with new raw data. The produced segments are buffered and finally wirelessly transmitted to a base station.

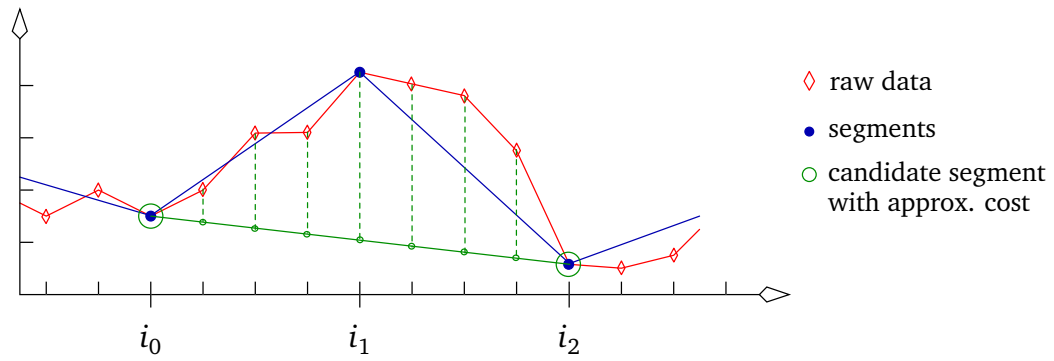


Figure 4.14: Example for the computation of the approximation cost (sum of distances) for a candidate segment during the bottom-up approximation step.

One of the main issues we know in advance we will be facing when porting mSWAB to an embedded system is that a floating point unit is not available in hardware. Thus, following two adoptions were necessary and are important to be noted:

First, the approximation cost function was changed from Euclidean Distance to sum of distances. This way, instead of computing the sum of squared distances and then computing the square root, we only look at absolute distances between the raw data points and their corresponding interpolated value on the segment. This adoption reduces the computational load for the microcontroller, as well as the memory requirements, by avoiding costly implementations of the square and the square-root functions.

Second, when computing the approximation cost between an approximating segment and raw data points, corresponding interpolation points on the segment need to be computed (Figure 4.14). For example: we want to compute the approximation cost of the candidate segment that will be created when the two blue segments will be merged. Given the two points for the candidate segment $(i_0; v_0)$ and $(i_2; v_2)$, both consisting of an index and the sensor value, we can utilize the linear interpolation formula

$$v_{interp} = v_0 + (i_0 + i_{interp}) \frac{\Delta_v}{\Delta_i} \quad , \quad \text{where } \Delta_v = v_2 - v_0 \text{ , } \Delta_i = i_2 - i_0$$

to compute the interpolation values for the indices $i_{interp} \in \{i_0 + 1, i_0 + 2, \dots, i_1, i_1 + 1, \dots, i_2 - 1\}$. The approximation cost c is then the sum of absolute distances between the raw data values and the interpolated points:

$$c_{(i_0, i_2)} = \sum_{k=i_0}^{i_2} |v_k - v_{interp, k}| \quad .$$

The lack of the floating point unit on a microcontroller is especially grave for the division step in the formula above that will cause additional error. For example, this error becomes in particular obvious when considering a candidate segment with a positive slope that is still smaller than 1: if $(i_0 + i_{interp}) \cdot \Delta_v < \Delta_i$, the division will result in zero. From this follows that the interpolated point equals the raw data value, and this results in a distance and thus in the approximation cost of zero.

To preserve as much accuracy as possible, we slightly transform the interpolation formula, forcing the division by Δ_i to be the last step in the computation:

$$v_{interp} = \frac{v_0 \cdot \Delta_i + (i_0 + i_{interp}) \cdot \Delta_v}{\Delta_i} \quad .$$

This transformation has been verified to better preserve the accuracy than the traditional way to compute the interpolation, and therefore was implemented in emSWAB.

Additionally, emSWAB can be further optimized by reducing the amount of repeating computations. For example, when the initial bottom-up approximation costs for a given buffer of raw data have been computed, a copy of those values can be stored and partially reused later. Once the approximation is finished and the next segment is produced, only the costs for those points will be removed that have been merged out by this particular segment. Other raw data points and their corresponding initial approximation costs remain. The approximation costs only need to be computed and stored for new incoming raw data values.

To allow easy portability to different hardware platforms and operating systems, we have chosen to implement our emSWAB algorithm as a library in standard C(95/99).

With the emSWAB algorithm now described in detail, we will next present the experiments methodology and evaluate its performance on several data sets of which the data could be captured within a wireless sensor network.

4.5.2 Experiments Methodology

The raw sensor data that is used during these experiments was taken from various public data sets. Examples of the data can be seen in Figure 4.15. Hereby, the data sets ECG 1 and 2, Power and Space are subsequences of the data sets that are freely available at <http://www.cs.ucr.edu/~eamonn/discords/>. The data sets Sleep and Hapkido are subsequences of data sets that are freely available from the ESS group.

The two ECG data sets represent two different anomalies in a normal heart beat, where the first is an oscillation of high frequency and the second is a small bump. The Power data set represents power requirements/loads that were recorded at a powerplant and spans multiple days. The Space data set shows a normal cycle in a Space

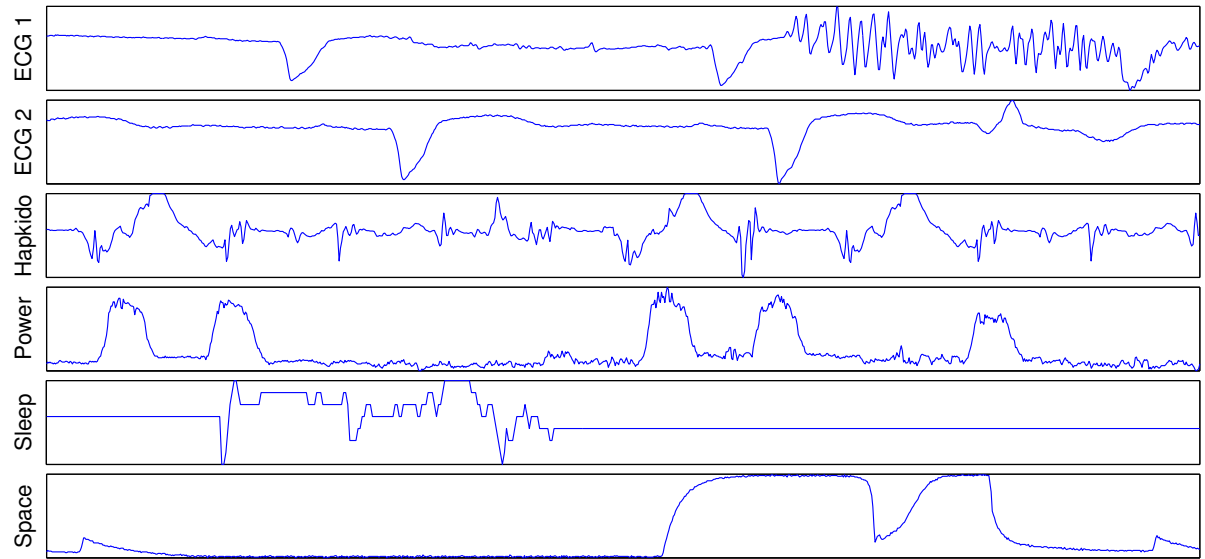


Figure 4.15: Different publicly available data sets used in the experiments. The data sets were chosen for their variety of the signal, displaying characteristic patterns. Whenever present, we have chosen the subsequences that contained anomalies, such as high variant section in the latter part of ECG 1 (top plot) or the sudden drop in the “energized” phase of the Space data set.

Shuttle Marotta Valve time series, with an anomaly during the “energized” phase, where the signal shall stay high. The Hapkido data set shows human activity data, namely acceleration data of a sensor that was attached to the ankle of a person performing Shinson Hapkido training. Finally, the Sleep data set shows accelerometer signals of a person sleeping and changing her sleeping position during the night.

The main aspect about the used data sets are the essentially different types of signal in terms of occurring patterns: on the one side of the spectrum there are long spans with flat signals (as in Sleep or Space) and constantly varying sensor values on the other side (as in Hapkido). Hereby, for our experiments and evaluation, the original sampling frequency is not considered important. More important is the shape of the signal and its preservation during the on-line approximation and abstraction on the sensor node.

We use these data sets to compress the signal utilizing our approach as well as the forwarding of raw data and the run-length encoding (RLE) technique. With our approach, once a new sensor value is sampled, it is forwarded to the emSWAB algorithm. The resulting approximating segments are buffered first and transmitted via radio to its neighbors or a base station when enough segments have been accumulated. Hereby, the segments are represented by data points consisting of a Δ_i value (number of samples that were abstracted to the previous segment point) and the actual sensor value.

Our evaluation goal is to be able to provide specific performance figures and estimated power consumption for the algorithm that is easy to reproduce. To achieve this, we have ported the emSWAB algorithm to Contiki (Dunkels *et al.*, 2004). Contiki is a very popular and widely used open-source, multi-platform and multi-tasking operating system for embedded wired and wireless sensor networks. It is written in C and is designed for variety of microcontroller- and microprocessor-based sensor nodes that have limited hardware resources. One of the main aims is a low-power radio commu-



(a) TelosB wireless sensor node.

ENERGEST_TYPE_	Consumption (mA)
CPU	1.8
LPM	0.05
LISTEN	20
TRANSMIT	23
LED_G	20
LED_Y	25
LED_R	25

(b) Current consumption figures.

Figure 4.16: a) The TelosB wireless sensor node based on the MSP430 microcontroller is used as the target hardware platform for this evaluation. b) Contiki OS with its built in software-based power-profiling mechanism Energest allows for energy consumption estimations, with figures from according data sheets.

nication in wireless sensor node networks. The Contiki OS project has various tools and implemented features, whereby the two most important for our work are the Cooja network simulator (Osterlind *et al.*, 2006) and the power-profiling mechanism Energest (Dunkels *et al.*, 2007).

Most experiments were conducted using the Cooja cross-level sensor network simulator. As the experiment target system we have chosen the very popular Tmote Sky / TelosB sensor node as the hardware platform. The node is a 8MHz MSP430-based board with 10kB RAM, equipped with a CC2420 IEEE 802.15.4 compliant radio chip, 1 megabyte external flash memory, and optional sensors. The most important current consumption figures, as noted in the Crossbow TelosB data sheet, are:

- MSP430, sensors and circuit: 1.8mA in active and 5.1μA in sleep mode,
- Radio transceiver: 23mA in active, 21μA in idle and 1μA in sleep mode.

Using Contiki's software-based on-line power-profiling mechanism Energest (Dunkels *et al.*, 2007), we were able to record how long the sensor node was staying in the low power mode (LPM) or was busy doing some computation (CPU), for example emSWAB computing the approximating segments, or was wirelessly transmitting (TX) data to the base station. Since our experiments focus on the sensor node sending abstracted data only, the time spent for actively listening or receiving data (RX) is negligible. By logging the time spent in each of these four states (modetime) and applying the corresponding power consumptions, current or overall power consumption per state over a period of time can be estimated, by for instance:

$$Power_{CPU}(mW) = \frac{\text{modetime}(ticks) \cdot 1.8(mA) \cdot 3(V)}{\text{frequency}(\frac{ticks}{s}) \cdot \text{runtime}(s)} .$$

Besides being able to use current power consumption to adapt sensor nodes behavior as it is for example done in related work, our primary goal is to evaluate how much power is used for computing a linear approximation of the signal.

Some figures on the sensor node images

The footprint of our experimental Tmote Sky / TelosB module image with emSWAB implemented in Contiki is 364.120 bytes, where less than 1% is taken by the emSWAB image. On a different sensor node, featuring a PIC18-based microcontroller from Microchip instead, our emSWAB implementation has a footprint of 2440 bytes.

The buffer size for the bottom-up approximation step was initially set to 20 values, bounding the buffer to minimum 10 and maximum 40 raw data values. When the approximation is computed, a buffer with corresponding indices that are stored as 16 bit unsigned integers is additionally needed. Thus, our implementation results in a buffer of 40 unsigned bytes (sensor values) and a buffer of 40 unsigned 16 bit integers (indices), requiring 120 bytes of memory. To be able to store and work with merging costs for adjacent pairs of segments, an additional array of unsigned 16 bit integers is needed. In order to speed up the costs computation (as described previously), two arrays are used, resulting in additional memory requirements of 156 bytes.

The structure implemented as union used to store the segments that are produced during the approximation step is 40 bytes large. Its size can be varied, depending on the size of the wireless communication package. We chose a value of up to 20 data points (19 segments) that are represented by an index and the actual sensor reading.

To avoid losing new raw sensor readings, a raw data buffer of size 120 is used, adding 120 bytes to the overall memory requirements. The buffered sensor readings will be copied into the emSWAB buffer. Once the approximation is computed, the data points that have been merged to one resulting segment will be deleted.

4.5.3 Results

After discussing the methodology and presenting some figures on the sensor node images, in this section we want to evaluate the performance of emSWAB for different merging threshold against the raw data dissemination and the run-length encoding (RLE) techniques.

In Figure 4.17, two plots visualize the Energest power estimation logs: the upper plot for RLE and the lower plot for emSWAB with a merging threshold of 5 and an initial bottom-up buffer size of 20. Here, the Hapkido data set was used. Note that this is only a fraction of the whole log that was recorded during the simulation, since visualizing the log entirely would make it unreadable. The X-axis represents the sensor reading ticks, whereas the Y-axis shows the number of CPU cycles that were spent in the four different system modes.

Both plots show that most of the time the sensor node stayed in low-power mode. Sampling and storing a new sensor value is almost negligible when abstracting the data with RLE, therefore CPU usage only appears when the data is wirelessly transmitted to the network (MAC overhead). The emSWAB plot, on the other hand, shows that more computation is required. However, the amount of data that needs to be wirelessly transmitted is reduced, resulting in a decreased amount of wireless communications: with the emSWAB approach 4 times, against 7 times with RLE.

Since in the example given above a relatively low merging threshold of 5 has been used for emSWAB, the resulting approximation is close to raw data. Increasing the threshold will result in a more coarse grained approximation. This will lead to more

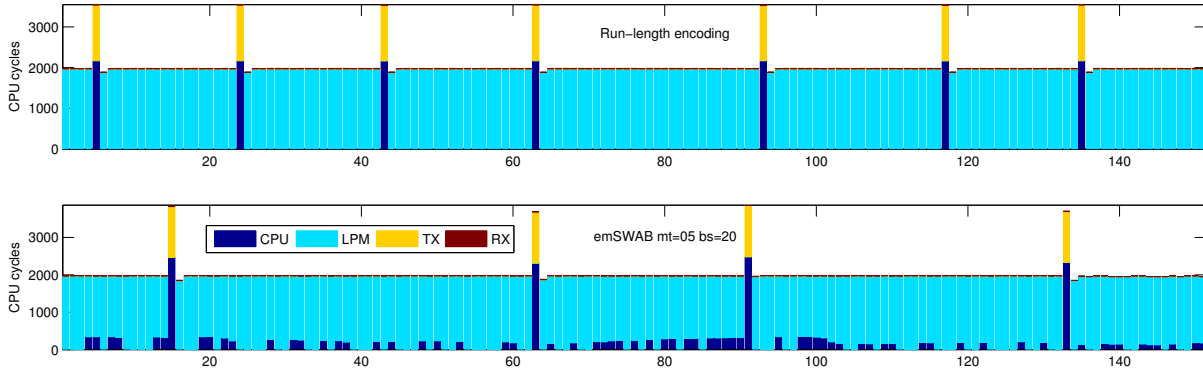


Figure 4.17: Power consumption estimation with Energest: comparing the performance of RLE and emSWAB using the Hapkido data set. The plots show how long (in terms of CPU cycles) the system stayed in one of the four modes: CPU - time spent on calculation, LPM - low power mode, TX - wirelessly transmitting the abstraction, RX - listening or receiving (negligible). In this example, RLE performs very similar to raw data dissemination (not shown here). Obviously, the emSWAB algorithm requires more time for computation, but profits through less data that needs to be wirelessly transmitted.

computation on the one hand, thus increasing the CPU load, but will reduce the amount of data to be transmitted.

The six plots in Figure 4.18 present emSWAB's performance on the data sets mentioned before (see Figure 4.15). To be able to compare the performance figures of the different sensor data abstraction techniques, and to decide which one performs better, we need to compute the relative time required for every system mode. This is achieved by normalizing the time figures for the four different system modes by the total runtime of the experiment. Additionally, emSWAB's main abstraction parameter is varied for every data set to show its direct impact on the algorithm's performance and the estimated power consumption. The values used for the merging threshold have been chosen as follows: $mt = \{5, 10, 15, 20\}$.

Figure 4.18a shows performance figures for the Hapkido data set. This data set has a fluctuating signal that leads to a poor performance of the RLE technique. A fine-grained emSWAB approximation ($mt = 5$) demands more time for its computation, thus resulting in a higher CPU load, but is balanced out by a smaller-sized abstraction. This results in lower package size and thus in less wireless communication, thereby preserving battery power. Increasing the merging threshold and by this forcing a more coarse-grained approximation will reduce the footprint even more, as can be seen in the plot, but result in a higher approximation error that might become critical for preserving the shape.

Figure 4.18b shows performance figures for the Sleep data set. This data set is different in nature compared to the Hapkido data set, in that it contains long periods with constant sensor values: the signal stays flat, is shortly interrupted by a jump to another level, and then stays flat again. The good performance of the RLE abstraction is therefore not surprising. emSWAB does perform well on this data, too, but at a much higher cost in terms of computation time, preventing the sensor node from entering the desirable low-power mode.

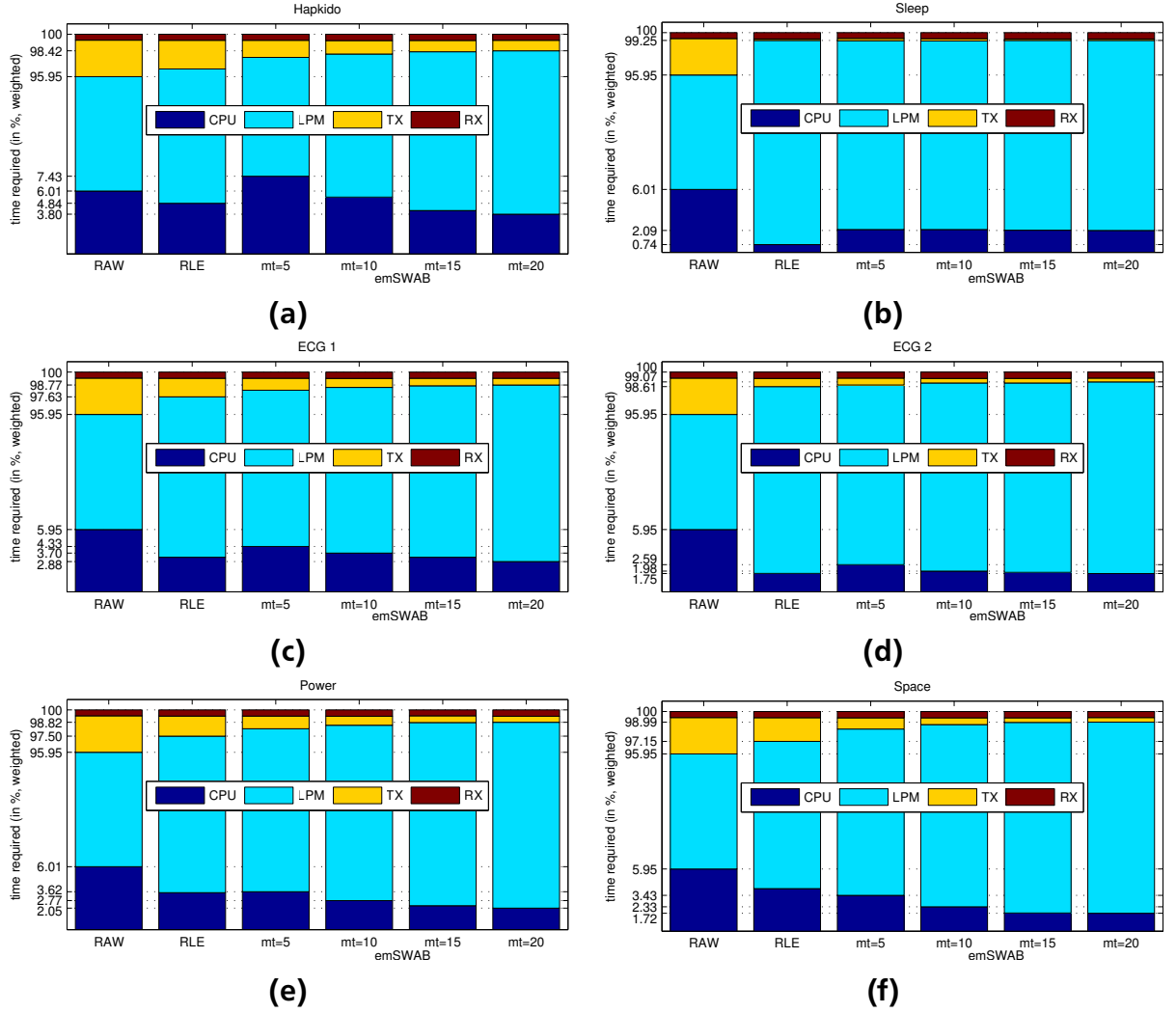


Figure 4.18: Energest performance comparison of raw data dissemination, RLE and emSWAB for different merging thresholds on six datasets. The y-axis shows time spent in the four different states (normalized, in percent).

Figures 4.18c and 4.18d show performance figures for the two ECG data sets. The main difference in these two data sets is the type of the anomaly present in the signal. ECG 1 has a part with highly varying signal that prevents RLE to perform as good as it does on ECG 2 that does not contain this kind of anomaly. emSWAB performs well on this combination of flat and high variance signal, outperforming RLE and matching the time needed for computation (and MAC communication overhead) when approximating with a merging threshold of 20.

Since the signal in the ECG 2 data set has no high variance part, RLE produces an abstraction of a smaller size. Also, emSWAB performs better on this data set, outperforming RLE even for the small merging threshold.

Figure 4.18e shows performance figures for the Power data set. Due to reoccurring characteristic patterns as well as lots of noise in the signal, RLE performs worse than on clean data as it is the case for Sleep or ECG 2 data sets. In this case emSWAB performs much better, especially with a higher merging threshold. Filtering out the noise and thus considerably reducing the footprint of the approximation, emSWAB even outperforms

RLE in terms of cumulative CPU load (approximation computation plus the wireless communication overhead).

Figure 4.18f shows the performance figures for the Space data set. The signal in this data set contains less noise than in the previous one, but RLE is still not performing optimal. This is due to the underlying shape of the signal that contains long periods of slowly climbing or falling sensor values. emSWAB's linear approximation does preserve this linear shape of the signal, reducing the amount of data to be transmitted, outperforming RLE both in terms of the footprint as well as the time needed for computation and the communication overhead.

In this section, we have presented the experiments methodology, some figures on the sensor node images and the Energest performance figures for our emSWAB approximation technique that we have compared to the raw data dissemination as well as the RLE methods. In the next section we want to draw some conclusions from these results and also indicate directions for future work.

4.5.4 Study Conclusions

With the experiments and the results presented in this study, we can first conclude that the modification and optimization made SWAB runnable on a microcontroller-based sensor node. Experiments with this implementation of emSWAB, although currently conducted in the Cooja simulator, show good performance on different data sets with varying signal characteristics.

The comparison of emSWAB's Energest performance figures to the figures of the commonly used RLE technique have shown that emSWAB can provide a good piecewise linear approximation of the signal that preserves its shape and - on most of the data sets used in this study - has a smaller footprint. The abstraction's size is especially crucial to a wireless sensor network as it has a direct impact on the amount of wireless communication in the network and therefore on the battery lifetime.

Additional computation overhead that is needed to produce the emSWAB's approximation is balanced out by the reduced amount of data that needs to be transmitted wirelessly throughout the network. This way, battery power can be preserved much better than with RLE or just transmitting raw data.

Our approach is especially targeting data with patterns essential to the sensor network application. This, of course, does not hold for sensor data that is similar to for instance that in the Sleep data set, as in this case other abstractions such as RLE perform better, both in terms of the abstraction size as well as the CPU load. Future work can consider further optimizations of emSWAB and possible combinations with other abstraction techniques, depending on the signal.

Besides the on-line adaptation of the approximation technique based on the sensor data, current power consumption, power consumption over a time span or the (estimated) remaining battery power can be utilized for adaptive sensor node behavior. For our approach, the adaptive behavior would mean that (based on the power consumption figures) the sensor node will automatically increase or decrease the merging threshold to allow more coarse- or fine-grained approximation of the sensor data. Increasing the merging threshold, thus reducing the number of approximating segments, will result in less data to be wirelessly transmitted and thus preserve battery power. On

the other hand, if the approximation needs to be especially fine-grained, and enough battery power is available, the merging threshold could be reduced allowing to preserve even finer details in the signal.

4.6 Conclusions

We presented in this chapter the applicability of early data abstraction techniques that preserve the shape of the time series. First, we presented our on-line mSWAB piecewise linear approximation algorithm, that computes Bottom-Up approximation on sensor readings in a small buffer window and updates the buffer borders very efficiently by considering slope sign changes in new sensor values.

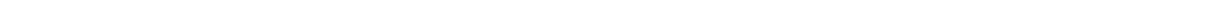
To show the potential of our modification, a case study motivated by activity recognition research has been carried out, where closest matches of query patterns, chosen by the user, have to be identified in huge time series as efficiently and accurate as possible. Such query by example pattern matching can be also employed with patterns that have been previously accumulated in a data base, thus allowing to perform the search for specific known patterns without human presence. The advantage of piecewise linear approximation allows for visual inspection on all stages, both approximation as well as pattern matching.

Aiming at embedded sensor applications, such as wearable sensors for long-term activity monitoring in medical applications, or the long-term deployment of sensor networks to monitor environmental phenomena, the abstraction of sensor data should also be performed directly at the source. In a second study we showcase the on-sensor implementation of the mSWAB algorithm which had to be adjusted to meet the hardware and computational constraints of such custom sensor devices.

Considering the on-going development towards more powerful and yet more energy efficient processing units, while higher energy consumption for wireless communication or using a local flash memory storage remains, the advantages of early data abstraction on the sensor node become more important. Reducing the amount of sensor readings such that the shape of the signal, arguably its essence, will not only allow for faster post-analysis of gathered data, but also preserve battery power and extend the lifetime of the sensor during the deployment.

In the following chapter we will consider other means of encoding the shape of the signal for a wireless sensor network monitoring application, hereby focusing not on evaluating a continuous time series, but particular physical phenomena of short duration captured by the sensor nodes.

Besides just computing the piecewise linear approximation (e.g. through segments produced by mSWAB), in Chapter 6 we consider further encoding of the signal shape to facilitate not only fast matching of patterns, but specifically to facilitate the finding of previously unknown recurring patterns in a challenging activity recognition scenario.



5 Complex Event Classification in WSN

The data logger presented in Chapter 3 with its MEMS acceleration sensor is able to capture not only translation, tilt and rotation, which are the basis for human motion analysis, but also another type of acceleration, namely vibration. This chapter focuses on capturing vibrations with the same power-efficient and inexpensive data logger, and analysis of the vibration signal with the aim of detecting and discerning events.

In the previous chapter we have presented an approximation algorithm that reduces the amount of sensor data and at the same time preserves the essence (shape) of the original time series, and have shown how it can be applied for human motion detection and activity recognition. In this chapter, we consider sporadic physical phenomena that cause vibrations, for which the proposed piecewise linear approximation and pattern matching algorithms are less suitable. On the other hand, the shape of the vibration signal contains characteristic and relevant information that can be abstracted and used for event detection and classification. Consequently, encoding the shape information in appropriate features, as exemplary shown in Figure 5.1, facilitates data analysis with our “shape as a feature” approach.

This section motivates our work in the domain of wireless sensor networks (WSN) applications and argues for utilization of accelerometer sensor nodes that can capture, abstract and analyze streaming vibration data with our shape-based features approach.

5.1 Motivation

In contrast to human motion data captured continuously for long stretches of time with wearable sensors, which is being analyzed off-line for recurring patterns or activities, this work considers long-term deployment of sensors for monitoring the environment and detecting as well as identifying events. While the prominent wireless sensor networks scenarios consider relatively slow-changing sensor readings (e.g.: temperature, humidity, gas concentration or air pollution), where often threshold based approaches are used to trigger an alarm, many more applications face the problem of, what we call, complex and sporadic phenomena.

Capturing and analyzing vibration events has already been a motivation for many wireless sensor network application scenarios, e.g. seismological (Werner-Allen *et al.*, 2006), infrastructure (Kim *et al.*, 2007), business (Wang *et al.*, 2006) or military

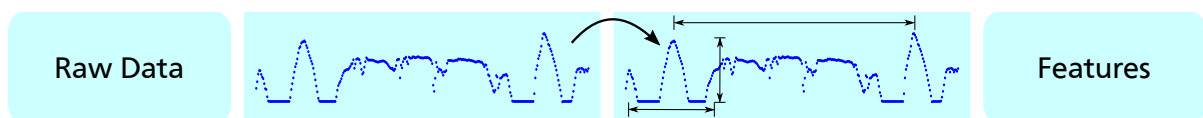


Figure 5.1: Encoding the shape of the signal as a set of features facilitates complex event classification directly on sensor nodes.

(William and Hoffman, 2011) monitoring applications. These scenarios often required specialized and expensive equipment tailored to these applications, with high sampling rates in the range of Kilohertz, demanding also more powerful techniques and analysis based on frequency domain features.

Following these WSN research, we argue that even with simple and inexpensive MEMS acceleration sensors, as presented in Chapter 3, vibration data can be captured and analyzed appropriately with shape-based features. With a relatively high sampling frequency on the one hand, and the limited hardware resources of a miniature sensor node on the other, the main challenge lies in on-line data abstraction and analysis, which would allow to transmit only the result of the analysis to the base station, instead of raw sensor data, thus facilitating power-efficient long-term monitoring applications.

Our approach for such long-term wireless sensor network deployment relies on a system that consists of

1. multiple miniature acceleration sensor nodes that are able to capture vibrations in the chosen type of monitoring scenario,
2. abstraction techniques and features that encode the shape of the vibration footprint and can be efficiently computed on the sensor node,
3. a straight forward classification system based on the shape features to discriminate the events directly on the sensor node.

Acceleration Sensor Nodes for WSN Deployment

With regard to acceleration sensors, we rely on the miniature, inexpensive and power-efficient hardware design presented in Chapter 3. The acceleration sensor is sampled at 100 Hertz, which is lower than the sampling rate of specialized hardware for capturing vibration data, but still sufficient in some application scenarios. An important feature of the accelerometer sensor is the ability to set a pre-defined acceleration threshold: Only when vibrations are detected that exceed this threshold, the acceleration sensor will wake up the main processing unit (the microcontroller) thus initiating on-line processing of the streaming sensor data.

The microcontroller is very power-efficient when in sleep mode, and still consumes little energy when computing. It is fast enough to process streaming data on-line, which allows to abstract the original data and extract relevant features almost in real-time. The sensor hardware used in the experiments lacks wireless communication possibilities, which can be solved by adding the required RF chip. Alternatively, our approach can also be implemented on already existing hardware platform.

Using the Shape of the Vibration Signal

To facilitate the detection and classification of events, we rely on features that encode the shape information of the vibration data, whereby we have consider not the signal itself (as with human motion), but highlight other signals' inherent characteristics. Our approach relies on abstracting the sensor data that preserves this essential information. Then, extracting relevant shape informations from this abstraction, and encoding these into features, is much more suitable for describing the underlying phenomena. Using fast feature extraction from streaming data that can be implemented on the sensor nodes, we show that on-sensor event classification can be achieved.



Figure 5.2: Our system’s concept: A sensor network deployed along railway tracks captures vibrations caused by passing trains. Immediately computing efficient features from streaming sensor data allows train type classification and counting wagons on the sensor nodes. In future deployments, these can be used to estimate train speed and detect worn-down cargo wheels.

In this work, we focus specifically on sensor data abstraction in an application where the sensors have been sampled at relatively high frequencies. Hereby, sampling rates range from hundreds of Hertz, for acceleration sensors and gyroscopes, up to thousands of Hertz for microphones. Using efficient and easy to compute features such as mean, variance, signal amplitude, and similar, abstraction of such sensor data is possible directly on the sensor nodes, even with their limited hardware resources. For applications where events require large amounts (i.e. multiple hundreds or thousands) of sensor readings to be adequately captured, computing such abstractions significantly reduces the amount of data in comparison to the original signal.

Using a scenario and real data from vibration signatures generated by passing trains, we show how with this approach the classification of passing trains and estimation of their lengths is possible on miniature sensor nodes.

Application scenario

Our approach of using the shape of the vibration signal as a feature directly on a distributed set of wireless nodes is applied to a railway monitoring scenario. Figure 5.2 presents our system concept, with a network of sensors deployed for capturing vibrations.

Figure 5.3 depicts one event from a data set recorded for the case study: The data set was obtained by deploying a network of sensor nodes that are equipped with sensitive inertial sensors at the railway tracks, capturing the vibrations caused by passing trains. We show that from the raw sensor data captured by the sensors in such a network, we are able to classify the type of train as well as the train’s length. To achieve this in a realistic setting, we limit our approach to a set of sufficiently efficient features that can be implemented on the sensor node in an on-line fashion, thus allowing on-sensor event detection, train type classification and length estimation.

The first experiment on railway tracks in Darmstadt shows that the chosen features produce good train type classification with up to 90% of trains correctly identified. The second feasibility study uses raw data obtained from a deployment on one of

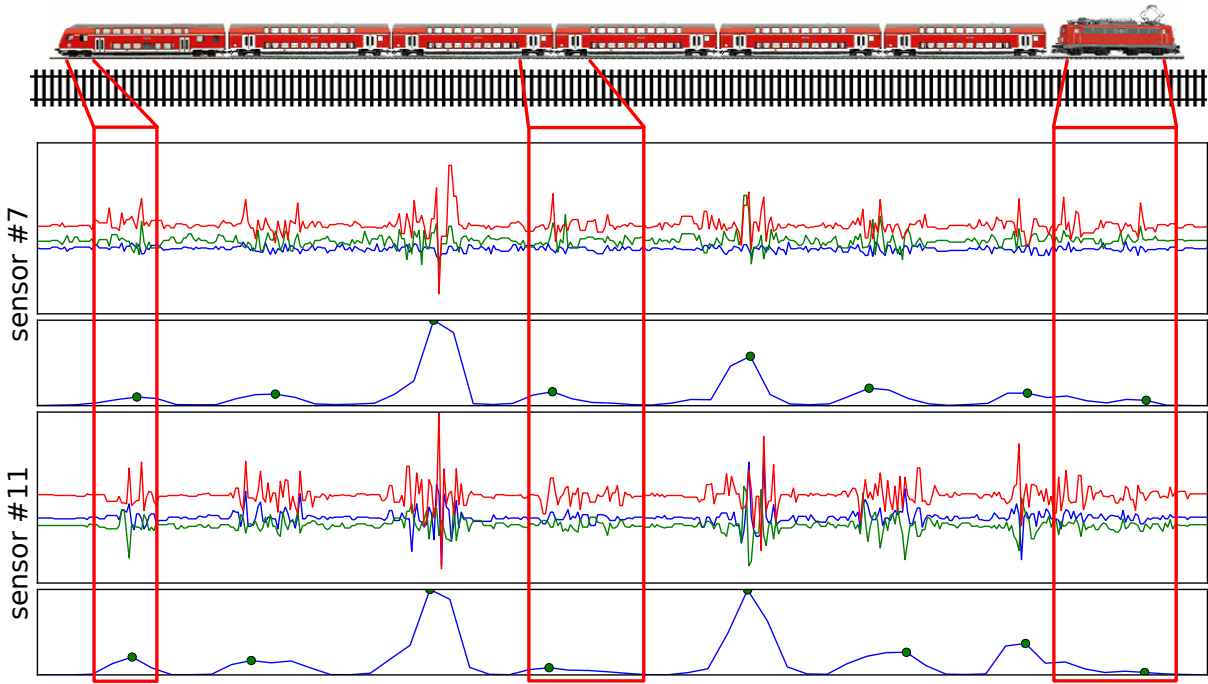


Figure 5.3: Miniature sensor nodes attached to the railway tracks capture the vibrations caused by passing trains. From the raw 3D acceleration data of these events (upper plots), features can be extracted that are characteristic enough to be used for on-sensor train classification (bottom plots). Using a network of such nodes makes the detection more robust and allows additional analysis, such as estimation of the train’s speed by using time delays between sensors.

Europe’s busiest railroad sections near Cologne, which was annotated with the help of video footage and contains vibration patterns of 186 trains. These trains were classified in 6 types by various methods, the best performing at an accuracy of 97%. To enrich the information about the train events, we also estimate the trains’ length in wagons. Visual inspection of the data shows further opportunities in the estimation of train speed and detection of worn-out cargo wheels.

The remainder of this chapter is structured as follows: Section 5.2 is dedicated to our feature extraction approach. In Section section 5.3 we present the experimental deployments from which we obtained our data sets, as well as the evaluation methodology and the evaluation results. Finally, we discuss our results in Section 5.4 and conclude the chapter in Section 5.5.

5.2 Encoding Shape of the Signal into Features

Focusing on the application of spotting and categorizing passing trains as a representative scenario for a wider range of application types, the main goal is to extract simple features directly on the sensor node itself, and propagate either these features throughout the network instead of the original raw data, or a classification based on these. In both cases the feature calculation is focused on, while we assume the classification to be either straightforward enough to also implement on the sensor node, or be done on a more powerful platform at the network’s sink.

For the remainder of this chapter we assume a sensor sampling rate of 100 Hz and the data resolution set to 10 bit. Although this is far from sufficient for exact vibration analysis, we argue that with using the basic features discussed in this section on data from low-cost but precise MEMS inertial sensors suffices to capture the events for the application's needs. The features discussed here will thus not rely on calculations and transformations in the frequency domain but instead approximate shapes and amplitudes within the signal to enable event classification.

Events are assumed to occur sparsely over the course of time, so most of the data acquired by the sensor is not relevant and can be discarded after verifying no events are present in the data. A windowed standard deviation calculation was found to accurately detect these flat signal sections between events.

Whenever an event occurs, the sensor node will thus detect the changing sensor values, including the start and stop times of the event and the event duration, and temporarily buffer the sensor data from the event for further evaluation. As the node's RAM tends to be limited, storing of the data stream can be done in an on-line fashion on peripheral memory such as an attached SD card. The feature analysis and calculation are thus limited to *on-line* algorithms: They are required to run incrementally on partial buffers of the event's data at a time.

Since the sensor has the time of event occurrence and also its **duration** in number of samples, the latter can directly be used as a distinct feature. This feature is similar to what others have used to detect types of ground vehicles, for instance by Keawkamnerd *et al.* (2008).

As we are interested in abstracting the vibrations pattern caused by trains, using the **overall variance** to describe a train event would be another higher-level feature. Extracting other features from the sensor data that describe the signal footprint requires a more detailed look at the signal properties. For this, the vibrations caused by each axle or carriage/truck of train wagons, can be extracted by a sliding window approach and represented as **variance peaks**, as shown in Figure 5.3. Counting these local maxima in the variance will create a feature that is expected to correlate to the number of wagons in the train.

The **amplitude** of the signal was also chosen as a feature. Hereby, either the real signal amplitude can be utilized or alternatively, since the windowed variance is computed for the above described peak detection, the maximum peak value as a representative for the amplitude can be used instead (see the bottom plots of Figure 5.3).

These four features were used in our first evaluation. For the second study, we added more features that were derived from the windowed variance and extracted peaks: the amount of vibrations of the trucks through maximum and **average of the amplitudes**, truck distances through the **average distance between peaks**, variety of wagon lengths or trucks constellations via **variance of peak distances**. Additionally, the overall **area under the variance curve**, as well as the **average area per peak** will be considered. For the off-line evaluation we compute this feature using the Python `scipy.integrate` library. Table 5.1 summarizes the proposed features used in our evaluations.

The ability to characterize the events directly on the sensor node with these features makes it possible to forward these few abstractions of the event instead of its original raw sensor data representation. When considering a wireless sensor network that

ID	Feature	Description
0	duration	event duration (vibration exceeding a threshold)
1	variance	total amount of vibration caused by the train
2	peaks	number of peaks extracted from windowed variance
3	max. amplitude	maximum peak value
4	avg. peaks	average distance between peaks
5	avg. amplitude	average peak amplitude
6	area	total area under curve
7	avg. area	average peak area under curve
8	var. peaks	variance of peak distances

Table 5.1: Overview of all features considered for train type classification. During the 5-fold cross validation on the data set using an SVM classifier, all possible feature combinations (with a minimum number of three) have been tested, whereby the features were also computed with a varying window size. The features 0–3 were used in the first study only, while the second study considered all of them.

should be deployed for railway monitoring tasks, a much more energy efficient way of notifying a base station or logging data for future off-line analysis is worth focusing on.

The presented feature routines are in essence the result of a trade-off between having highly-accurate vibration information but requiring a high amount of processing power, and settling for more abstract information while being able to do these calculations on more light-weight platforms. The features also do not require thousands of samples per second. From this follows that relatively inexpensive and power-efficient sensor nodes can be utilized with microcontrollers that can for instance lack floating point units.

5.3 Evaluation

The experiments discussed in this section rely on datasets obtained with the prototype sensor node, while placed at multiple locations on railroad tracks in Darmstadt and in Cologne. In the case of our studies, results needed to be reproducible and we therefore opted for continuous logging of raw data on the local flash memory, which is also a power-intensive operation.

The first deployment was carried out at two locations in Darmstadt and was intended as a “proof of concept” experiment. Since the deployment locations had to be easily human accessible, we chose a railway crossing and a train station.

In the second deployment, our focus was widened to evaluate more features for train type prediction, as well as extracting more information about the events for the railway monitoring scenario, such as length of the train (number of wagons). For that, we obtained a more challenging real-world data set along with detailed annotations of the vibration patterns. This time, railway company officials were involved in the planning of the experiments, who suggested the location for the deployment of the sensor nodes to be near Cologne, specifically due to the variety of train types and their maximum possible speeds. The particular spot featured four tracks running completely straight for multiple kilometers, thus allowing train speeds of up to 250 kilometers per hour.



Figure 5.4: Different train types that were recorded and classified during the evaluation. Fast inter-city and the city hopper passenger trains at the top, a fast regional passenger and a cargo train at the bottom.

The recorded datasets are presented next, whereby we give an overview on their characteristics and information content.

5.3.1 Trainspotting Data Sets

Darmstadt

The data obtained in the first deployment comes from two separate recordings that were conducted on different railroad tracks in different locations, in order to make the data set more diverse. The combined data contains in total 247 events, where an event is defined as a train passing by, whereby 182 of the trains were annotated with the corresponding train type as they could be traced by to the available train schedule. The remaining 65 train events were labeled with "unknown" and not used in the classification evaluation. Figure 5.4 shows some types of trains that were observed to be running on those tracks.

Figure 5.5 shows a part of the dataset, including details of a fraction of the data, and showing two exemplar events caused by a regional passenger train and a cargo train.

The first of the two recordings comes from a low duty railroad track and has approximately 24 hours of continuous sensor data, during which in total 53 trains passed by. This track services only smaller passenger trains that consist of a two-car articulated unit (Figure 5.4, upper right), whereby trains consisting of multiple such wagons were spotted during rush hours. The second recording was carried out on a busier railroad, contains 35 hours of continuous vibration data with 194 train events in total. This track is used by a higher variety of trains, such as inter-city and regional passenger trains as well as cargo trains. The sensor node was this time deployed nearby a train station that is served by some regional passenger trains. While some trains were passing the station without slowing down, others did halt at this station, thus adding more diversity to train speeds and thus their signature's length.

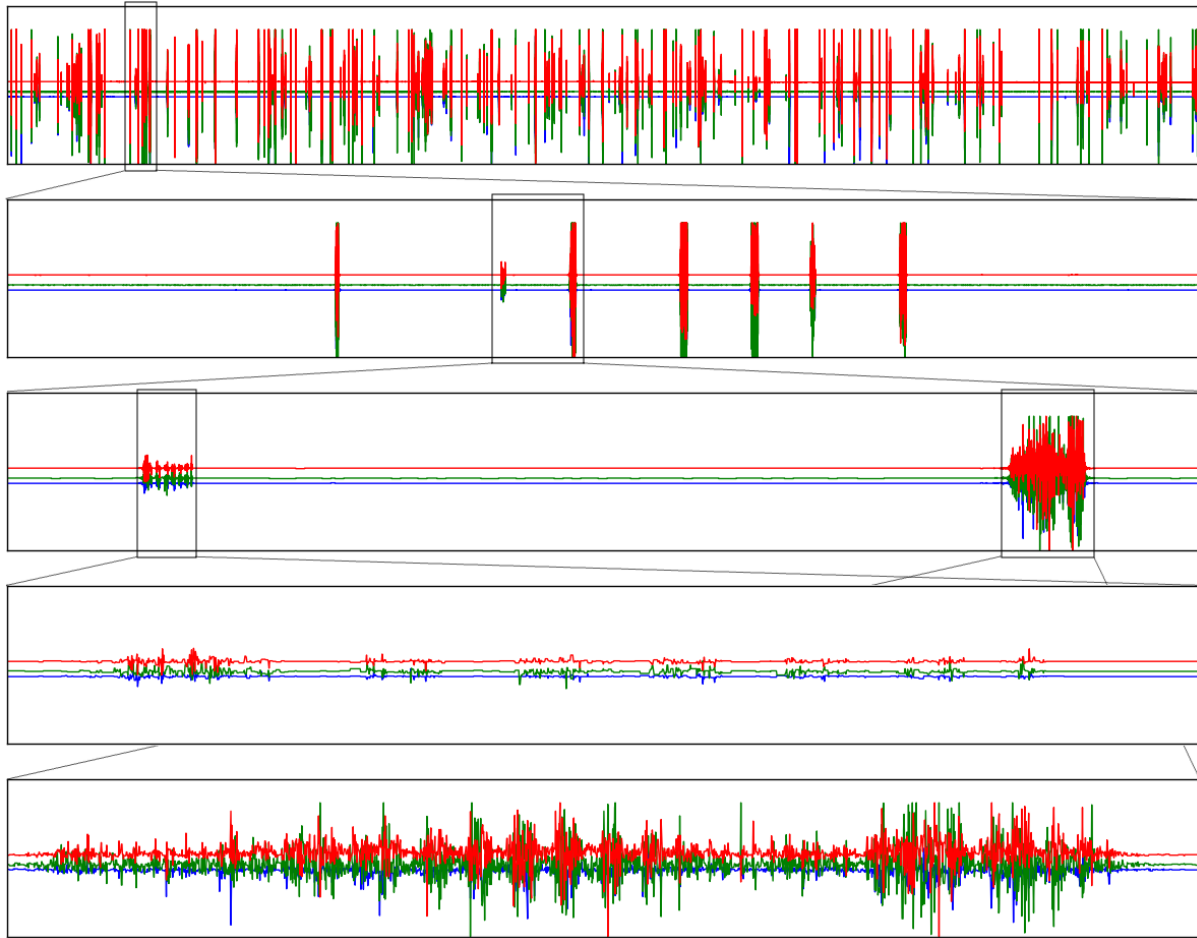


Figure 5.5: Part of the data set, showing approximately 35 hours of sensor data (top plot): The data contain sparse but complex events caused by passing trains. Vibrations patterns shown in the two bottom plots were caused by a regional passenger train accelerating from the nearby station (duration 16 seconds) and a cargo train with loaded wagons (duration 30 seconds). The proposed features were tested on this data with two common classifiers.

Since the dataset contains many different train types that mainly differ by name due to their scheduled tours, but in reality turn out to be similar regarding the type of wagons and locomotive used, a decision was made to group the annotations in four main categories. The categories reflect main train types as they tend to be found on European railroads. The three inter-city passenger train types (ICE, IC, EC) mostly consist of same type and number of wagons are therefore grouped as class A. The two regional train types (RE, RB) were categorized as class B. All types of cargo trains were put together in class C, while so-called city-hopper passenger trains form class D.

Cologne

In the second deployment, the sensors have captured 186 train events in total, of which 141 could be annotated based on video footage recorded during the deployment. Figure 5.6 illustrates the video data with a series of frames from video footage capturing a single locomotive passing by. Table 5.2 provides an overview on the six different train



Figure 5.6: Snapshots from the video recordings showing in this case a single locomotive passing by on the 2nd high-speed track. These videos were used as ground truth for the evaluation of type of train and train composition (wagon count).

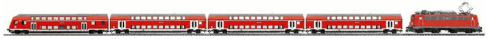





Type	Description	Count	Model
Regio	passenger trains connecting cities	63	
CityRail	trains service city center, suburbs	15	
Cargo	various cargo trains	39	
Loc	single locomotives being transferred	10	
Thalys	French high-speed passenger train	5	
ICE	German high-speed passenger train	9	

Table 5.2: A total of 141 train events in raw sensor data could be annotated and used for the evaluation. Here, different train types and their count in the data set are shown. The model depicts the trucks' locations, which differ among the train types, with two trucks for a wagon (e.g. ICE) or one between them (e.g. CityRail, Thalys), resulting in characteristic vibration footprints.

types: four different passenger train classes – two types of high-speed trains, regional passenger trains and city rail trains – along with a cargo and locomotive classes.

Thalys, a French high-speed passenger train, typically consists of head and tail locomotives and 8 passenger wagons which are connected to a single continuous unit, resulting in 10 wagons in total. The ICE is a German high-speed passenger train which in our experiment typically contained 8 railmotor wagons. The Regio class contains the regional passenger trains that connect nearby cities within a region, but do not stop at stations in between. Regional trains consist of a locomotive pulling or pushing a number of bi-level wagons (as shown in Figure 5.2 or the corresponding model in Table 5.2). In our experiment, these trains' lengths were 3, 5, 6 and 7 wagons in total. The CityRail trains connect larger cities with its suburbs and other smaller towns nearby, and typically consist of two electrical units with 4 wagons each. In our experiment, these trains were running exclusively on the separate low-speed tracks.

The Cargo class has proven to be rather versatile, with one characteristic feature that all cargo trains have in common: at least one locomotive is pulling a highly varying number of wagons. Both the locomotives as well as the wagons themselves can be of different types (e.g., tanks, containers, car- or freight wagons), as well as different lengths and truck constellations. In our experiment, cargo trains had mostly one, some-

times two, locomotives with a total number of wagons ranging from 13 up to 43. The locomotives class was added due to single locomotives being transferred to another station. In the experiment, 10 such events have been captured, whereby both single as well as two connected locomotives have been observed.

Summary

With these data sets, we were able to perform extensive evaluations, in which particular focus is given to finding a set of efficient-to-calculate features that can be implemented directly on the sensor nodes for on-line train type classification. A second objective that has been identified as valuable information to automatically detect by the sensor network is the estimation of train length. The following section will present the proposed features, the classification performance and train length estimation results.

The next sections will discuss the implementation and parametrization of the features used with this dataset to classify the train events present in this dataset, as well as their evaluation and discussion of the results.

5.3.2 Evaluation Methodology

This section discusses which features have been extracted from sensor data with which set of parameters, and how they were evaluated by using them to classify the train events.

Since our evaluations were conducted off-line for reproducibility, a first step requires that from the large amount of data only the actual train events are detected from sensor data. This was achieved by computing the variance over a sliding window with a relatively small size of one second (using half a second for window overlap) and a small threshold to cancel out the small amounts of noise in the original raw data. With this step we also acquire the start and stop time of an event, thus being able to compute its duration. Additionally, the overall variance of this event is computed to give a rough estimation on how much vibration the train has caused.

In addition to these overall event features, a more short-term sliding window was used to extract dense vibration patterns that correspond to the wheel impacts on the rails. For this step, the choice of window size is crucial, as from this depends whether we are counting the axles, the trucks (also called 'bogie' or a 'wheel truck'), adjacent trucks, or whole wagons (carriages with multiple axles). This truck count will then be used as another feature for classification, but can also be used to estimate the train length or the train's configuration (e.g., in rail bridge monitoring).

To achieve this, we tested sliding window sizes on the interval from 100 up to 300 milliseconds (or 10 to 30 samples). Computing windowed variance resulted in the characteristic plots shown in Figure 5.7, where raw sensor data is in the upper and the resulting variance in the bottom plot. A window size of 160 milliseconds (16 samples) was found to perform best for train type classification. Using a peak detection algorithm based on the slope of the signal, the local maxima in the variance plot were found and highlighted as green dots. The number of peaks thus tends to correlate to the number of wagons the train consists of, and this feature can be expected to be of particular importance for distinguishing the cargo train class (an example of this can also be seen in Figure 5.7a).

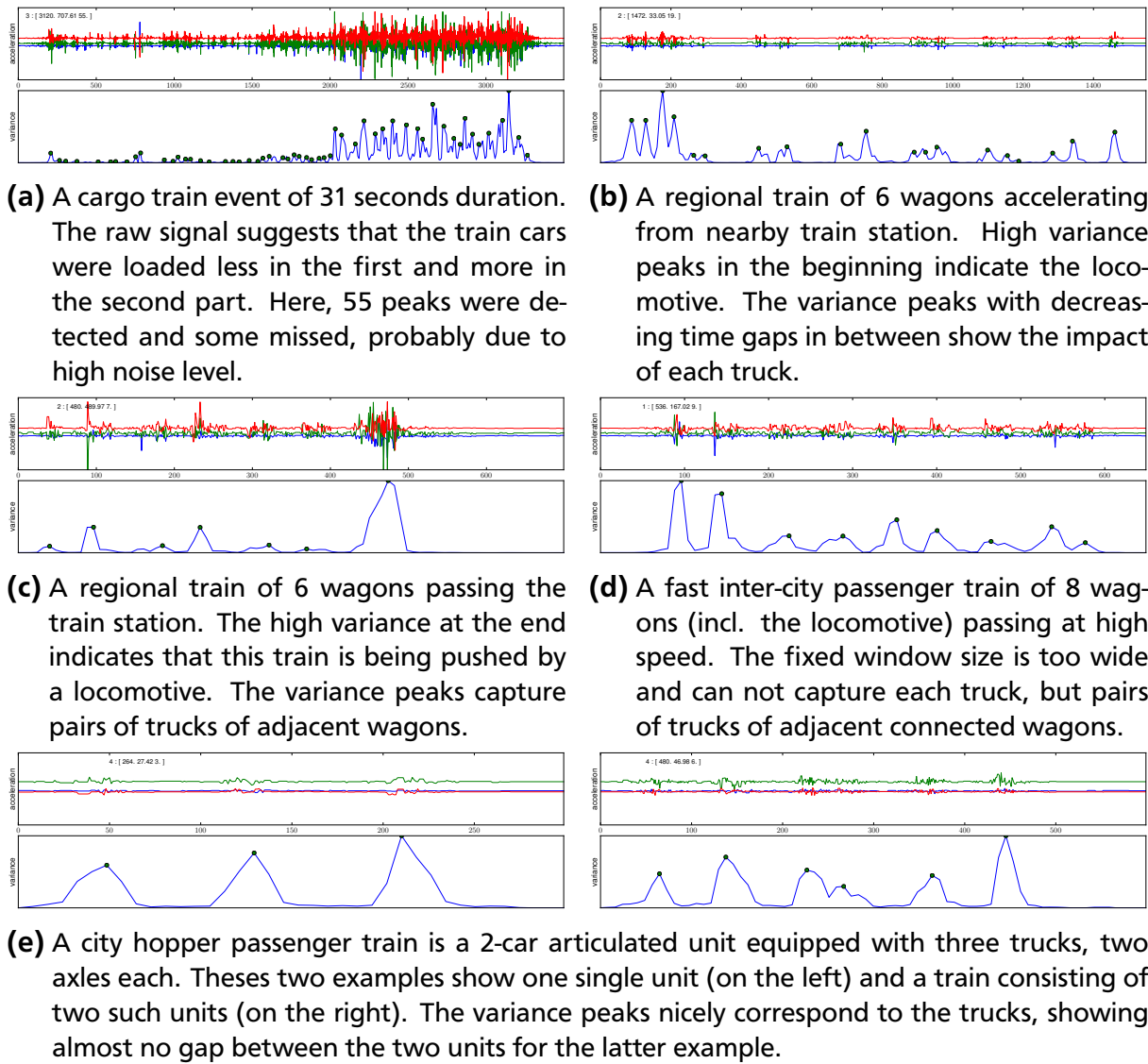


Figure 5.7: Examples of train events that were detected in the sensor data. Each subfigure shows raw acceleration data in the upper and variance computed on a sliding window of 0.16 seconds in the bottom plot. Peaks extracted from the variance are marked with green bullets. Due to various train assemblies and different train speeds, just counting the number of variance peaks computed on a fixed sliding speeds will not give a good classification performance.

With the window size of 160 milliseconds, an early analysis showed that counting the trucks to estimate number of passenger carriages worked fairly robust. Counting trucks to estimate the number of wagons for the cargo trains on the other hand turned out to be more error-prone, most likely due to the high vibration levels caused by the cargo wagons and the high variance of different train speeds (due to cargo trains often passing the nearby station at low speeds, and also speed limits that are for instance imposed during the night time or at peak times). Counting pairs of trucks (with two axles each) of adjacent wagons, on the other hand, can be distinguished more easily for all train types.

With these features calculated, a 5-fold leave-one-out cross validation study was conducted. All events were grouped per train class and those were divided in 5 folds, whereby 4 were used for training the classifier and remaining fold was used for testing. For each fold, the true labels of the cross validation part (given by the annotation of the dataset) as well as the labels obtained by the classification were stored, and afterwards used to build the confusion matrices. From a confusion matrix, we then compute the classification accuracy per class of interest as well as the total accuracy for the given set of features and chosen parameters.

Two classifiers were used for the evaluation: The K nearest neighbor (kNN) classifier was chosen particularly due to its simplicity and popularity. Additionally, a support vector machine (SVM) classifier was chosen for comparison. This way we are able to evaluate the features' performance, and test whether the choice of the classifiers has a significant impact, too. For the kNN classifier, k was set to 5 nearest neighbors, as it was found to produce best classification results. In the SVM case, a linear kernel was used.

On the second data set, based on the results obtained from the first evaluation, the following range of window sizes was found to be of interest for evaluation: 12, 14, 16, 18, 20 data points. Since the kNN classifier did not perform well on this more challenging data set, in this study we rely on the SVM classifier for the train type prediction. The feature space was normalized before being used for the evaluation.

This time, since the classes sizes significantly differ, the performance evaluation was carried out through a *stratified* 5-fold cross-validation, whereby the size proportion of the six classes was preserved. Hereby, all possible feature combinations have been tested (with three as a minimum features set size), resulting in 466 combinations. Multiplying this with the range of window sizes, we end up with 2330 cross-validation runs. In the following we present the cheapest (with regard to the number of features required) and best performing feature combinations.

5.3.3 Train Type Classification

This section presents and discusses the results of the kNN and SVM classifiers that were chosen for the evaluation on the two data sets.

Darmstadt

For the first one, the confusion matrices given in Figure 5.8 show the classification performance for the kNN classifier (top row, $k = 5$,) and SVM classifier (bottom row) and the four train categories. Hereby, different combination of features were tested (event duration, overall variance, maximum amplitude, peaks count), and the sliding window size was set to 160 milliseconds.

Relying on the event duration as a single feature results in a weak classification performance, as can be seen in the confusion matrix shown in Figure 5.8a. The fast inter-city trains are mostly confused with the city hopper trains, reaching a class accuracy of only 22.22%. The reason for this strong confusion lies in the relation of speed and train length: on average a short and slow city hopper train generates a vibration pattern that is as long in time as the fast inter-city trains which are longer but pass by faster (from 3 up to 7 seconds). The same observation holds for other train types as well, whereby

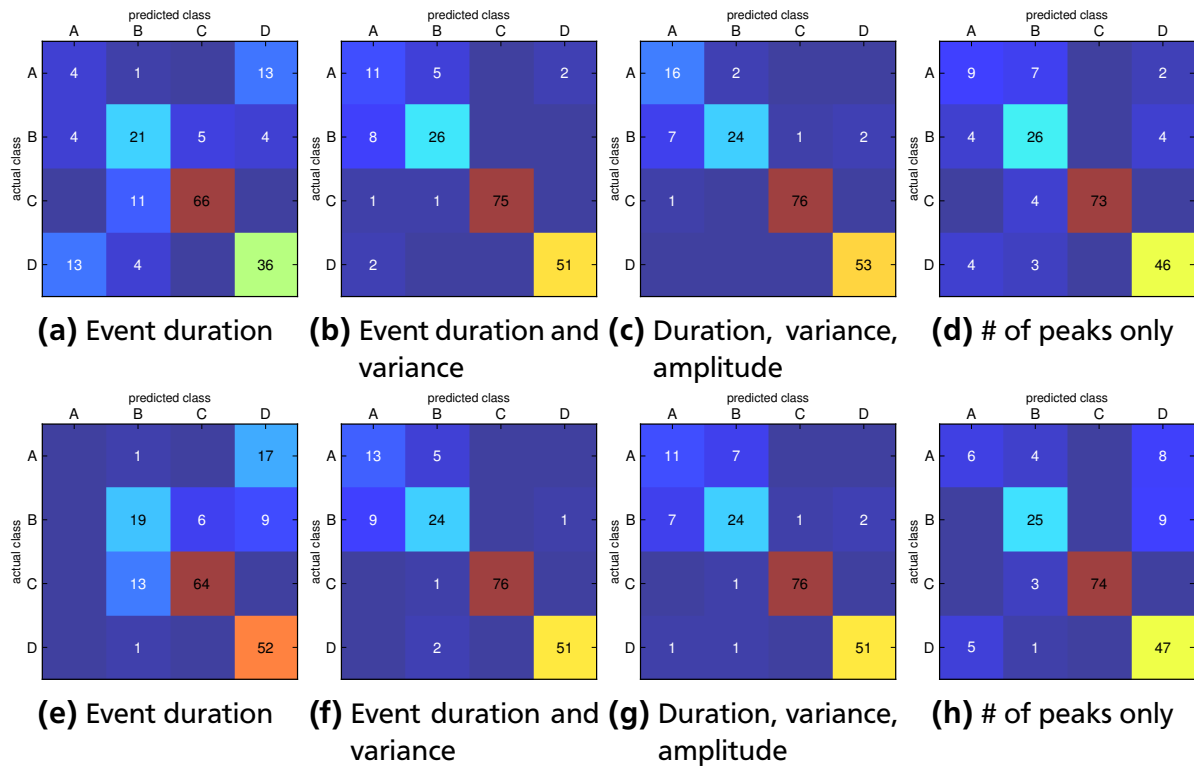


Figure 5.8: Summary of the kNN (top) and SVM (bottom) classification results for the four train type classes from the first data set and different feature combinations presented as confusion matrices. The classes are: A - fast inter-city trains; B - regional passenger trains; C - cargo trains; D - city hopper trains.

other train classes do not exhibit such high degree of confusion, reaching class accuracies of 61.76, 85.71 and 67.92% respectively. The SVM classifier's performance with event duration as the single feature is much worse for the inter-city train class: these are heavily confused with the city hopper passenger trains (cf. Figure 5.8e). The other classes perform considerably better, with 55.88, 83.12 and 98.11% per-class accuracy respectively, producing a total accuracy of 74.18%, and thus slightly outperforming the kNN classifier overall.

Using the total event variance as a second feature significantly improves the kNN classification's performance (Figure 5.8b). The accuracy for the inter-city class jumps to 61.11%, since the total vibration impact of a faster and heavier inter-city train is higher than of the light city hopper, the confusion at this spot is drastically reduced compared to the previous results. A similar but less significant improvement can be observed with the cargo train and city hopper classes. The confusions between the inter-city and regional trains on the other hand still remains, which most likely is due to their similar duration and vibration signature. This especially becomes graspable when considering just the regional trains that do not stop at the nearby train station and therefore do not slow down (cf. Figure 5.7 c) and d)). All train type classes thus gain a performance boost, now reaching 76.47, 97.4 and 96.23% in per-class accuracy. The total accuracy for the chosen set of features lies at 89.56%. Only slightly better results can be achieved by the SVM classifier, where the total accuracy reaches 90.11% (Figure 5.8f).

The number of peaks in the vibration pattern has been previously mentioned as an attractive feature candidate. Testing different window sizes from 0.1 up to 0.3 seconds in 0.02 steps revealed, to our surprise, that using the peaks as an additional feature does not improve the performance significantly. While there is no improvement for the kNN at all, the SVM classifier improvement accounts to 0.3 % over the previous feature set (with duration and overall variance). The reason might lie in the fact that the number of peaks corresponds heavily to the duration of the event already, and thus not offering much more information to distinguish train categories.

Since the windowed variance was computed with a fixed window width for all events, regardless of their duration, one possibility to improve on the per-class accuracies might be to adapting the parameters for the windowed variance according to the length of the pattern. By adding the maximum value of the windowed variance (which represents the variance amplitude), we are able to boost the total accuracy a little bit more, reaching 92.86%. When evaluating the performance of each class (Figure 5.8c), we observe that the city hopper train class is performing at 100% accuracy. Only one false hit happened with the cargo train class, being confused with a fast inter-city train. Some confusion still happens between the fast inter-city and the regional passenger train classes, which, as already mentioned, is most likely due to the non-stopping regional passenger trains that belong to the class. In this case, the kNN classifier shows better performance than the SVM, which reaches a total accuracy of 89.01%.

When considering just the number of peaks (wagons, windowed variance footprint), without duration or total variance of the event, the classification performance drops to a total accuracy of 84.62% for kNN (Figure 5.8d) and 83.52% for SVM (Figure 5.8h). Since the number of peaks corresponds to the duration of the event on the one hand, but also covers the vibration properties of the trains, the obtained performance lies between the performances of duration on the one hand and duration with overall variance as features on the other.

Cologne

The classification results obtained during the 5-fold cross validation were accumulated, and confusion matrices were computed averaged over the number of folds. Figure 5.9 shows the three most illustrative confusion matrices, along with their parameters, the window size and the set of features (cf. Table 5.1).

For the first evaluations with the first three features only (duration, total variance and number of peaks), the SVM classifier was able to reach an overall accuracy of 90.78% for the window size of 18 data points. Adding the maximum amplitude to the feature set led to an increase of total accuracy 93.62% for the window size of 16. Adding more features to the set or interchanging them would improve the accuracy in very little steps, such that many combinations would reach a classification performance with 136 out of 141 train types correctly identified (96.45% accuracy). Figures 5.9a and 5.9b show two confusion matrices with corresponding feature sets that were able to achieve this high classification performance.

The feature set consisting of feature IDs 1, 4, 5, 6, 7 and 8 (computed from the windowed variance with a window size of 16 data points) has reached the maximum possible accuracy of 97.16%, with 137 of 141 train events being correctly identified (Figure 5.9c).

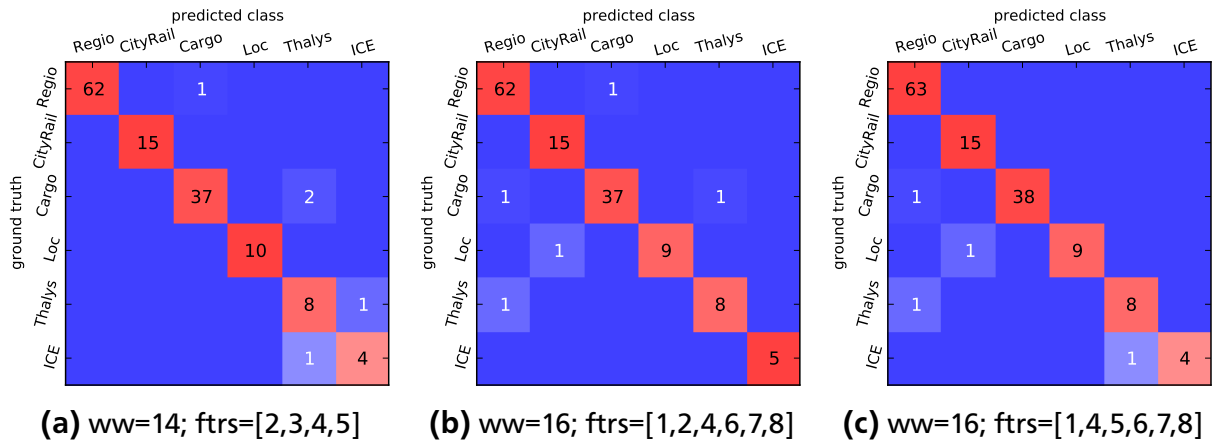


Figure 5.9: Exemplary selection of confusion matrices obtained during the 5-fold cross-validation with the SVM classifier. a) and b): These two matrices show an accuracy of 96.45%, for different sets of features computed on different window sizes. c) This confusion matrix shows the feature set performing best, reaching an accuracy of 97.16%.

These classification performance on our data set suggests that training an SVM classifier off-line and implementing it on the sensor nodes would allow to detect train events, compute features and predict train types directly at the signal source with high accuracy.

Predicting the types of passing trains with reaching accuracies up to 96% and 97%, would be sufficiently promising for several applications. Following our scenario, deploying a sensor network with such a SVM classifier implemented on each sensor node, it is still possible to improve on the classification performance. This can be achieved by utilizing the sensor network’s communication capabilities and let neighboring sensor nodes decide upon the train type by a voting mechanism amongst classifiers.

After evaluating the train type classification performance with features, the following section will give further insight on how well the train length estimation worked.

5.3.4 Train Length Estimation

To estimate the train length, we primarily rely on counting the number of wagons in the trains. This can be achieved by using the already introduced feature “number of peaks” as a basis. Additionally, using the train type obtained from the previous estimation step is used as a prior. The wagon count can furthermore be improved by a comparison and voting procedure amongst neighboring sensor nodes on the same railroad track.

Besides using the raw signal and computed features, it is useful to include inherent model knowledge about the train type constellations: The ICE and Thalys high-speed passenger trains as well as the CityRail trains consist of specific wagons only (locomotives are built-in or the wagons are motorized themselves). Regio trains consist of varying amount of wagons with a separate locomotive. While with these trains the axles constellations are fixed due to defined sets of wagons, the cargo train class poses a much higher variety: wagons with single, double and triple axles per truck, wagons of different lengths, and varying load are possible.

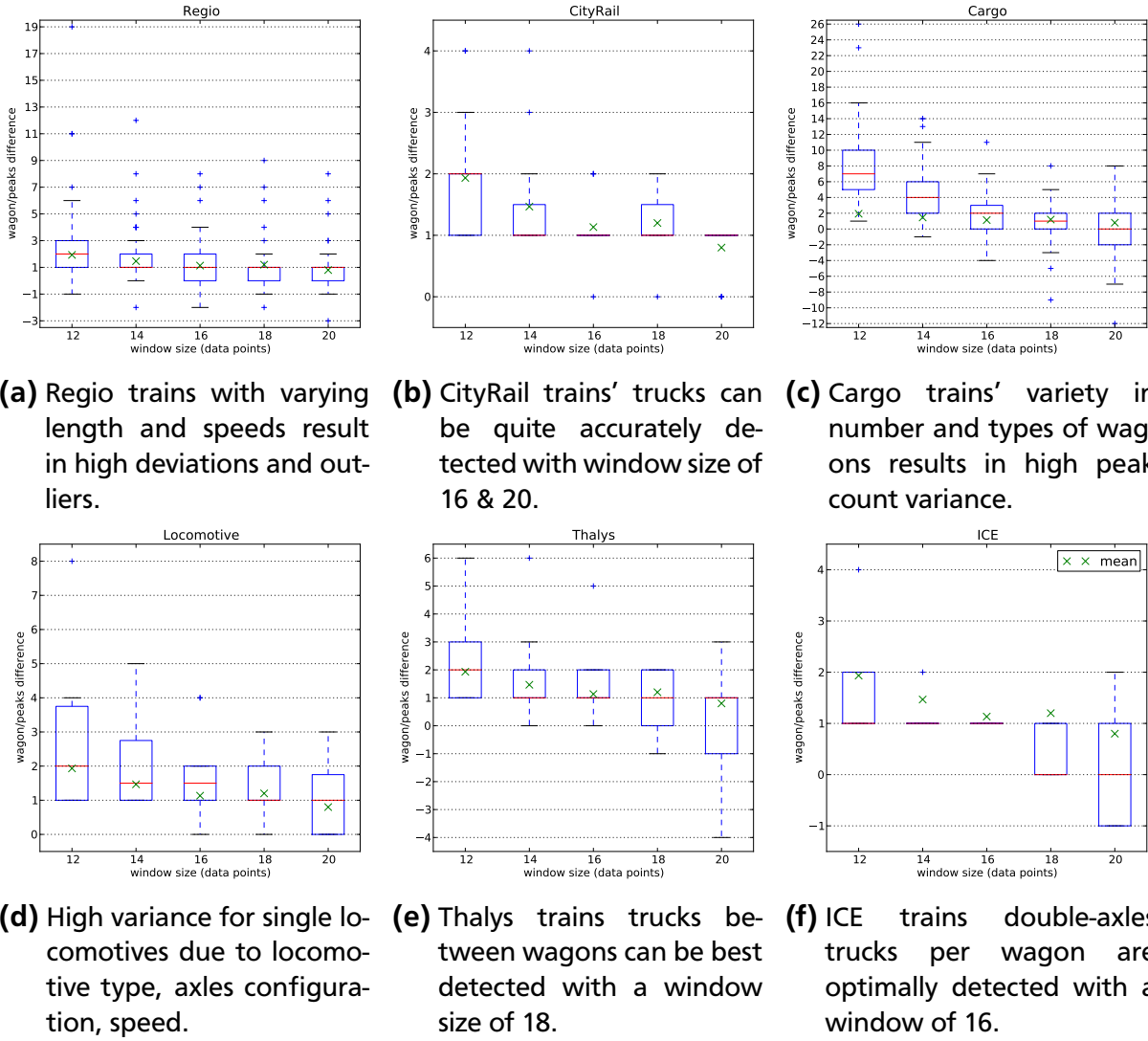


Figure 5.10: Differences between the real number of wagons and number of peaks computed from the windowed variance for each of the six train classes. The number of peaks and therefore the accuracy of the wagons count depends on the size of the sliding window and the train speed. From these results, window sizes of 16 and 18 data points are performing best for counting wagons. This is verified by the overall minimum squared-mean error of approximately 4.0 shown in Table 5.3.

Using the annotations from the video footage as ground truth (number of wagons) and the number of peaks extracted from the windowed variance, we use their difference for performance analysis. Since the peaks correspond to the trucks, the number of peaks usually is by 1 more than the amount of wagons in the train. This deviation of 1 can be visually recognized in the box plots shown in Figure 5.10. In addition, we compute the mean-squared error for the whole data set as well as for each individual class (see Table 5.3), whereby the deviation has been accordingly taken into account.

For a more concrete example, consider a regional passenger train with 7 wagons (including the locomotive). For this train, the peak detection algorithm extracts 8 distinctive peaks (cf. Figure 5.3). Hereby we observe that the first peak belongs to the

Window Size	Mean-Squared Error						
	Overall	Regio	CityRail	Cargo	Loc	Thalys	ICE
12	28.14	11.78	1.87	78.95	7.30	4.00	2.00
14	10.33	4.65	1.00	27.95	2.60	3.56	0.20
16	4.02	2.84	0.27	8.74	2.20	2.33	0.00
18	3.98	2.89	0.33	8.87	1.20	1.44	0.60
20	5.62	2.43	0.20	14.28	1.00	6.56	2.00

Table 5.3: Mean-squared error of the estimated train lengths for the whole data set and per class. The window size has a huge impact on the quality of the estimation. Overall, window size of 18 performs slightly better than a window size of 16.

first trucks (double axles) of the first wagon, the following 5 peaks belong to adjacent wagon trucks (two times double axles for passenger wagons), and the last two peaks represent the last wagon and the locomotive (which has triple axle trucks that can not be distinguished in the signal with the fixed window size). Due to Regio trains' variety in length (3, 5, 6 and 7 wagons including one or even two locomotives) and their varying speed when passing by the sensors, the relation of wagons to the number of peaks tends to highly deviate as well as show lots of outliers (Figure 5.10a).

The window size to compute the windowed variance from raw sensor data has a significant impact on the peak detection. On the other hand, leaving the window size fixed at the best performing size of 16 data points (for classification and length estimation) would deteriorate the system's performance, as a fixed window size for a fixed sampling rate of the sensor node leads to the issue of not being speed independent.

5.4 Discussion and Outlook

This section discusses the evaluation results and will point out particularly interesting findings for the underlying scenario.

First, good train type classification results (up to 92% and 97% accuracy) could be achieved on our data sets with proposed features. During the evaluation, suitable window sizes (16 data points) both for type prediction and train length estimation could be found. Better performance in this regard can be achieved through implementing distributed voting among neighboring sensor nodes inside the sensor network. This would allow the sensor nodes to compare decisions and remove outliers.

Estimating train length with a fixed window size bears the problem of not being speed independent. In case of a very slowly moving train, which is very likely to happen and has also been observed in our data set, the fixed window will lead to detecting separate axles instead of trucks, resulting in a completely misleading peak count. One possible approach to tackle this issue would be the introduction of a variable window size. This would, on the other hand, lead to a more complex feature extraction routine and result in more computation on the sensor node and therefore in a higher power consumption.

Besides the problems addressed in this work, the recorded data set allows to extract much more information useful for the railway monitoring scenario.

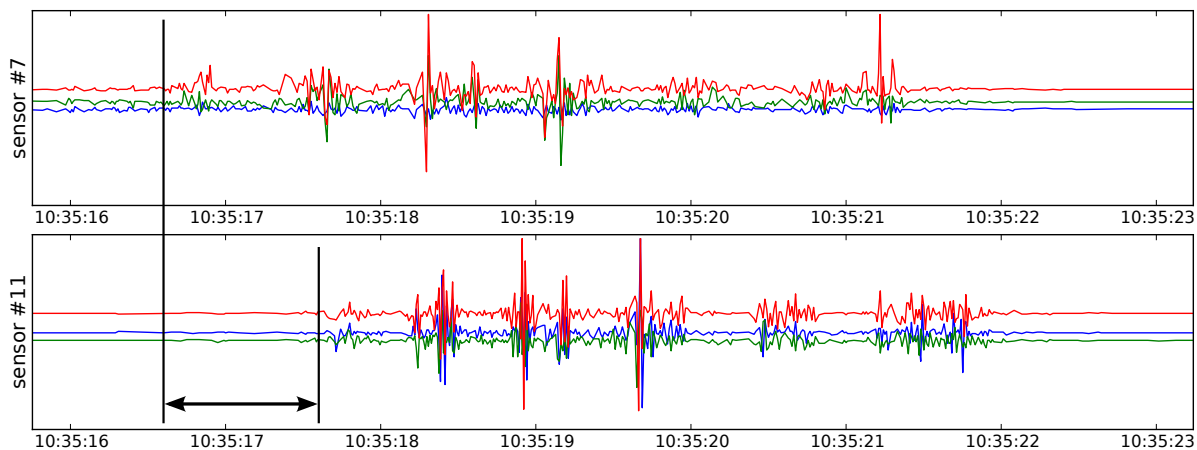


Figure 5.11: Two sensor nodes showing the vibrations footprint caused by a passing regional passenger train. Knowing the distance between the sensor nodes (10 meters in our deployment) and the time delay of the event between two sensors (markers in the plots) will allow estimating the train speed.

Estimating train speeds belongs to this category of very useful details and can be achieved with multiple sensors placed at a predefined distance which, in our experiment, was 10 meters. Detecting a passing train on two sensors and then computing the time delay between event arrival seems to be an easy and reasonable approach. For this, time synchronization inside the sensor network is important, but can be nowadays considered as a solved problem. An example for the feasibility is shown in Figure 5.11: vibrations caused by a passing regional passenger train are captured by two sensors in 10 meters distance from each other. By aligning these raw sensor data in time, the delay became visible.

Another very promising application for such a sensor network would be the detection of worn-out or defect wheels. Figure 5.12 shows data from two sensors which have picked up extreme accelerations caused by a passing cargo train. These extreme amplitude peaks in the raw sensor data are most likely caused by worn wheels (having lost their roundness due to abrasion caused by blocking when the train brakes). These worn-down wheels could cause damage to rails or rail bed, as well as the wagons themselves, thus making the detection of such events particularly interesting.

5.5 Conclusions

In this chapter we presented and evaluated the suitability of a sensor network consisting of tiny, inexpensive sensor nodes for a train monitoring application. It relies on sensor data from 3D MEMS accelerometers that are able to capture vibrations caused by running trains. To enable in-network event detection and train type classification, we proposed a set of features that encode the shape of the vibration signal, and can be computed efficiently in an on-line fashion directly on the sensor nodes.

We have carried out two studies that involved deployment of sensors at railway tracks, recording real-world data sets and video footage for annotation purposes. With these data sets we conducted evaluations of the proposed features. In the second more chal-

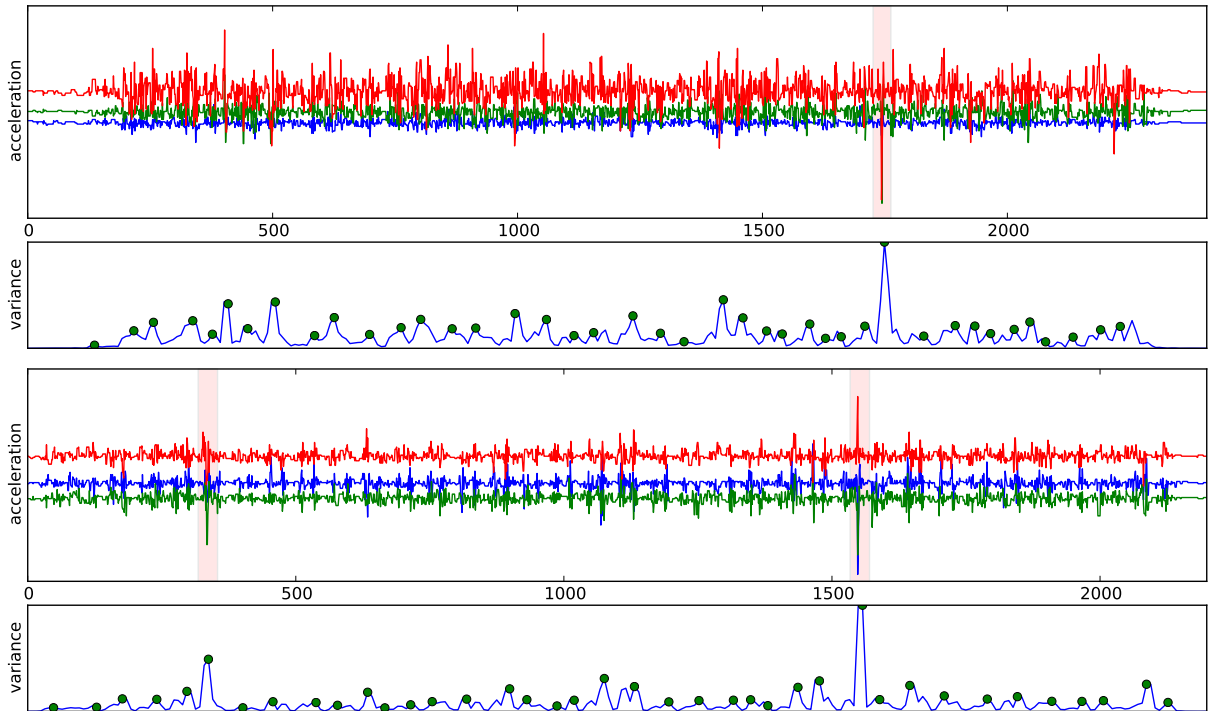



Figure 5.12: The sensor nodes have picked up extreme impacts (large peaks in raw data and variance plots) from a passing cargo train, caused by a defect wheel that is not completely balanced due to wear during braking. Since these wheels could cause damage to rails, rail bed, and the wagons, detecting such events automatically would be of significant interest as well.

lenging study, we found using an SVM classifier and a combination of features extracted from a sliding buffer with window size of 16 data points, to produce optimal results for this data set. The SVM classification performance for this optimal combination reached 97% accuracy. The length estimation performance accounted to 3.98 mean-squared error for the whole data set.

To summarize, we have presented a system that

1. uses power-efficient sensor nodes with accelerometer sensors to capture vibrations at a sampling frequency of 100 Hz,
2. relies on the shape of the vibration signal and encodes it into features that can be efficiently computed in an on-line fashion directly on the sensor node,
3. is able, based on these shape features, to perform optimal train type classification as well as to estimate train lengths,
4. and has shown to yield further potential for railway monitoring applications, including speed estimation and detection of defect wheels.

The findings of this study specifically show how using the shape of a time series, or the shape of its abstraction, can be productively utilized in long-term wireless sensor network application scenarios, where harsh computational limitations of the hardware platform and strict power-efficiency demands prevail.



In the next chapter we will return to the human activity recognition domain, where we are specifically interested in automatically accumulating recurring motion patterns that can be used as evidence for activities in a simple and efficient classifier.

6 Activity Recognition Through Dense Motif Discovery

In Chapter 4, we proposed a method to abstract raw sensor data by piecewise linear segments that significantly reduces the amount of data and enables faster query-based pattern matching. Relying on the shape of the signal as a feature, and assuming that different activities have distinguishable characteristic motion patterns, we are interested in building a system that is able to perform activity classification autonomously, specifically without any interactive input from a user in form of selected patterns.

To this end, this chapter proposes an activity inference system for deployment in long-term monitoring scenarios, to detect selected physical activities in week-long continuous data. Our approach relies on a customized data mining method that performs early sensor data abstraction and discovers motifs as evidence for activities (Figure 6.1). An extensive study investigates both accuracy and execution speed of the system and its sub-components. Results show that this method can be used to detect many physical activities robustly against a large amount of background data, with a comparable precision and recall to conventional approaches, in approximately the time it takes to download and graphically present the recordings from the sensor.

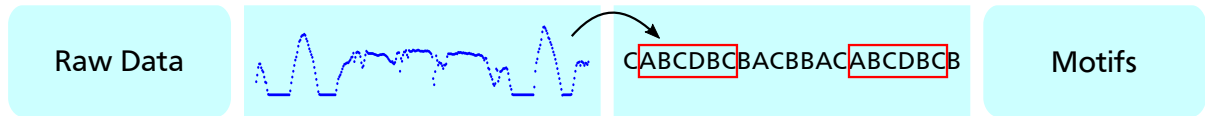


Figure 6.1: Abstracting the acceleration time series to its symbolic representation allows to efficiently obtain recurring patterns (motifs).

6.1 Motivation

Our recognition approach is targeting a class of long-term applications that remains challenging, with one of such scenarios being psychiatric patient monitoring, which aims at following mood and behavioral trends by recording activity data over a period of typically *several months*. Existing commercial actigraphy solutions are able to record the level of human activity and to detect sleep and wake cycles for such long deployments. Unfortunately, activity levels are a too coarse-grained abstraction and thus are not suitable for activity recognition purposes, which is specifically desirable as it would give more insight on the patients phases and well-being.

In the psychiatric monitoring scenario, some general problems in activity recognition are bypassed: Patients already keep detailed diaries of their activities so that supervised learning methods can be employed, and only a few key physical activities linked to daily routine are of interest in the logged data. Other requirements, however, form

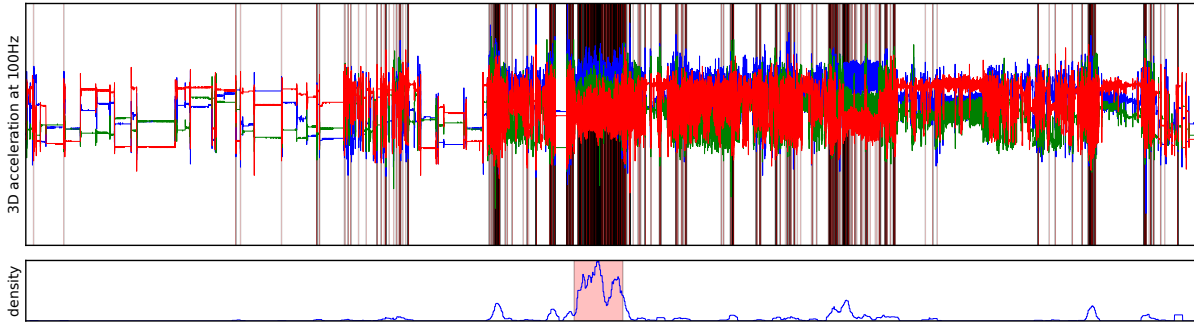


Figure 6.2: Day-and-night recordings from a wrist-worn acceleration sensor (top) are analyzed for certain physical activities. Saliency detection upon motif discovery (bottom) is used to find typical activity patterns (black marks) as supporting evidence for the activity.

novel challenges: Sensors should record for long stretches of time, the large amount of logged data needs to be analyzed fast, and detection needs to be robust against a deluge of background data. On the other hand, the actual motion patterns that would represent particularly interesting activities are not known and have to be obtained automatically, without additional effort for the patient or the doctors.

Figure 6.2 illustrates the dense motif discovery method that forms the basis of our detection approach. Occurrences of activity-characteristic patterns, so-called *motifs*, are searched for in new data, and the density of these is used to substantiate the presence of an activity. Motifs are discovered at training time by abstracting the raw acceleration samples in function of sequences of peak patterns, and efficiently searching for such patterns through a suffix tree. This search can be implemented in linear time, and performs a shape-based abstraction of the original time series. Classification is then implemented with a straightforward bag-of-words classifier. An extra advantage of this method is that the illustration of particular motif occurrences in the time series allows for visual inspection of the activity recognition before the classification step.

The main contributions of this work are threefold:

- A deployable system has been built consisting of a minimally invasive, wrist-worn sensor that is able to last for two weeks on a single battery charge while recording acceleration samples at 100 Hz, and a data analysis tool that can efficiently process the recorded data.
- A novel detection approach is suggested that is suitable for classifying and detecting activities in large amounts of long-term data, relying on local shape features within the acceleration signal and dense motif discovery.
- An extensive study with 33 participants who performed different activities was conducted to evaluate the proposed approach and discuss the applicability of the method for the application in mood monitoring scenarios.

The remainder of this chapter is structured as follows: First, a long-term monitoring scenario motivates the need for the proposed fast and accurate detection of when a user performs physical activities. The next section will present the details on the different

design choices and steps that constitute our method, such as the linear abstraction of inertial data and the use of suffix trees for finding motifs. A study then presents results on 33 week-long datasets, investigating how fast and robustly the chosen activities can be recognized among a large amount of daily activity data, followed by a discussion on general applicability of the proposed method.

6.2 Case Study – Bipolar Monitoring Scenario

We focus first and foremost on a practical detection method that is able to recognize particular activities within large datasets with a massive amount of background data, generally holding weeks of activity data at a time. As a case study of where this would be applied, we describe here the long-term monitoring of bipolar patients. Research in mood disorders, such as attention deficit hyperactivity disorder and bipolar disorder (Corkum *et al.*, 2001) relies frequently on the patients' self-reports, as well as semi-structured interviews with a psychiatrist, during diagnosis and therapy. Work with actigraphy tools in psychiatry (Wilhelm *et al.*, 2006; Teicher, 1995) has started to deploy wrist-worn sensors in conjunction with these tools that are recording the activity *intensities* observed for the patient from several seconds to minutes at a time.

Characterized by severe mood swings between manic or hypomanic, mixed, as well as depressive episodes, it is important in the diagnosis of a bipolar disorder to record the patient's activities over multiple weeks to months at a time. For mania for instance, energy levels tend to be high and activities tend to be performed in an interleaved fashion or especially vigorously. Similarly, depression tends to correlate with lower activity levels or in shortened key activities, from not performing them at all or sparsely, to not fully completing them. Apart from daily activities such as sleep and food intake, especially physical and leisure activities are very likely to be impacted: Patients might for instance stop playing tennis during depressions, or vigorously practice for several hours without breaks in a manic episode.

Interviews with psychiatrists resulted in a list of basic requirements that an activity recognition method should adhere to. These were grouped in three categories that are important to consider when designing an intelligent recognition system:

- **Supervised learning.** Patients are typically interviewed at regular intervals of several weeks, and provide log entries to report on performed tasks and their mood. Current actigraphs combine these reports with sensor data, so that the reports can be used to train patient-specific classifiers.
- **Week-long, 24/7 data.** Data needs to be captured continuously, as patients that go through depression or manic episodes are known to perform activities irregularly, including at night. The sensor units thus need to be robust and power-efficient to keep recording continuously, and the data will be substantial to process.
- **Activities with characteristic motions vs. lots of background data.** The number of activity classes to recognize is relatively small and can be determined by medical staff during first observations. This makes it easier for patients to keep track of what activities were performed, and means only few activities need to be detected amongst a large amount of background data that might produce false positives.

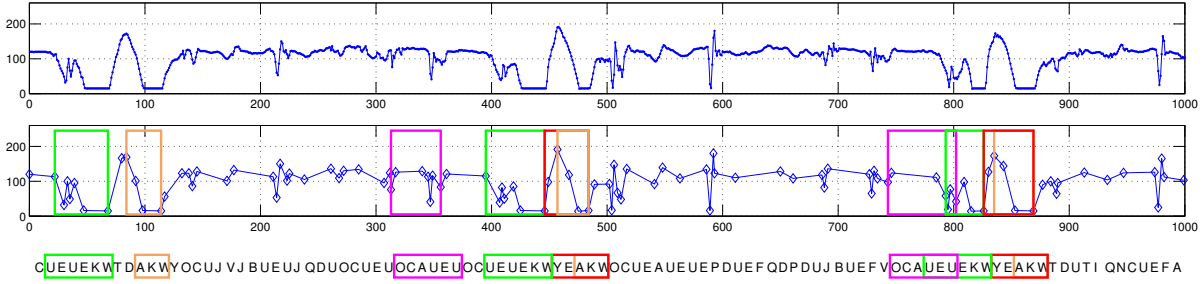


Figure 6.3: The raw 3D 100 Hz inertial data (top plot) are transformed by a piecewise linear approximation algorithm into segments (bottom plot) that preserve the shape of the signal to facilitate storage and analysis. The segments are subsequently abstracted in discrete symbols (bottom row) to allow fast discovery and matching of motifs. Occurrences for four motifs are highlighted by colored boxes; Note that we allow overlaps and variations in length.

We presented in Chapter 3 the details on our wearable sensor platform that was custom-designed and built focusing on long-term deployments, such as presented in this bipolar monitoring scenario. The next sections will thus present our automated approach to obtain characteristic motion patterns.

6.3 Dense Motif Discovery

This section gives an overview of the search and selection procedure for motifs from raw inertial data, and motivates the use of *dense* motifs. Early abstraction in multiple steps of the accelerometer data, together with a search-optimized data structure called suffix tree, guarantee that searching through weeks of data becomes feasible on standard computing hardware, and that classification can be done almost simultaneously with the downloading of the sensor data.

6.3.1 Method Overview

Motif discovery refers to the search for recurring sequences or patterns within a data stream. For real-world application scenarios such as ours, the original data tends to be noisy and hard to match exactly, posing a significant computational problem. Previous research has identified several techniques to represent the original data in a discrete symbolic string, where one or multiple data abstraction steps are necessary. Aiming at characterizing recurring motion patterns (i.e., activity-specific gestures) by the shape of the time series, our approach implements a discrete mapping through a two-step abstraction process.

Figure 6.3 illustrates how the proposed method transforms inertial data to a symbolic representation that facilitates the finding of recurring motifs: The original sensor data consists of 3D accelerometer readings that were equidistantly taken at a sampling frequency of 100 Hz. Using an on-line piecewise linear approximation algorithm, these inertial samples are abstracted to a set of linear segments, whereby the algorithm minimizes the residual error between the original samples and the produced segments. The

next step discretizes the segments into symbols based on the slopes of two adjacent linear segments. With the symbolic representation of the time series at hand, the next step utilizes the suffix tree representation of the target activity’s training data to accumulate a set of motifs. In the last step, the motifs that also have occurrences in the training’s background data (i.e., in the huge amount of data outside the target activity), are removed from this collection.

As a *dense* set of motifs is trained for, classification is done by searching new data for time windows in which motifs from one particular activity are frequently occurring. This is implemented with a bag-of-words classifier which uses the detected motif occurrences as evidence.

6.3.2 Approximation from Raw Data

The first abstraction step is essential from an efficiency point of view: Since the accelerometer sensor is sampled at a frequency of 100 Hz, capturing the essence of the gestures and typical motions performed by the sensor’s wearer, this also means that the dataset size grows quickly and becomes computationally challenging. Even with fast and lossless compression techniques, such as run-length encoding, 24 hours worth of continuous sensor data typically contain millions of 3D acceleration samples.

We argue that for motif discovery in inertial data, primarily the shape of the acceleration time series is important to retain. For our method we use a modification of the Sliding Window and Bottom-Up algorithm that we have presented in Chapter 4 and which has been verified to perform well on wrist-worn accelerometer data.

6.3.3 Mapping to Discrete Symbols

After abstracting the raw acceleration data to linear segments, a discretization step is used to obtain a symbolic string representation of the original time series. This abstraction step is first and foremost required to enable fast discovery of motifs, but also for finding matches between motifs.

First, we evaluated two degrees of freedom per segment (length and slope), and mapping the segments onto symbols in a similar way as was done with the SAX approach by Lin *et al.* (2003). Our approach also discretizes the feature value space based on the distribution of the values. The main difference to SAX is that the first abstraction step produces constant segments of fixed length, thus having only one degree of freedom, while SWAB produces linear segments with individual slope and length. With this approach, our initial test showed that very long segments became over-represented in the motif discovery. This is due to inherent properties of accelerometer data, with long segments with a slope close to zero being over-represented, particularly during the night time and sedentary tasks, where little or no changes are present in the signal.

Being interested in mainly short and characteristic gestures, focus went to represent peak patterns of neighboring segments. Thus, for one dimensional approximation, the slopes of two neighboring segments are considered, where-by we use the angular representation of the slope defined as $\theta = \arctan(m)$. In the case of multidimensional approximation, we compute the angle of neighboring segments in the 2D plane spanned by the segments.

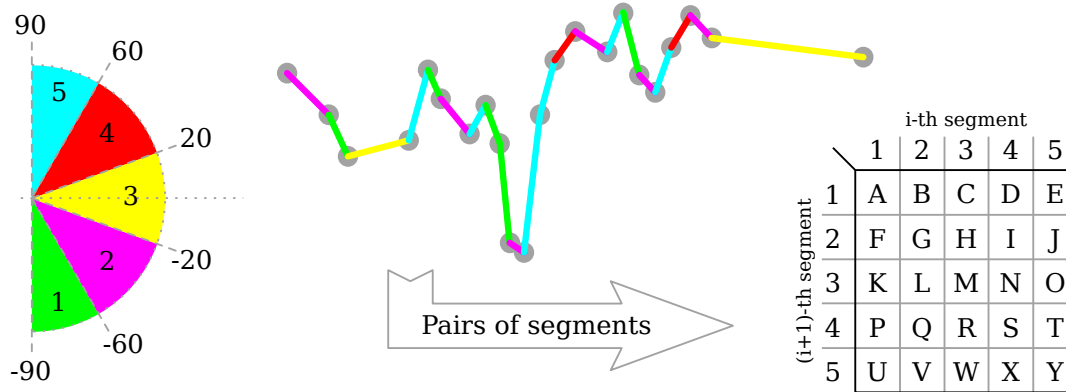


Figure 6.4: Mapping from linear segments to symbols: The slope range is divided into bins for a given number of separation points which are computed based on the training data segments' slopes histogram. The segments' corresponding bin numbers are then used as indices for the symbols matrix. Sliding through the segmented time series while considering two neighboring segments will thus result in a symbolic string.

To achieve discretization, the slope range from -90 to 90 degrees (or the planar angle range from 0 to 180 degrees, respectively) was divided into bins, whereby the borders (quantiles) are selected on the basis of representative data in a histogram during the training phase. To avoid over-representation of non-motion motifs, segments with a slope close or equal to zero were not considered. The rest, where we do not assume Gaussian distribution, is used to compute the quantiles for a given number of bins, which was found to produce the best results when set to 5.

Mapping the linear segments to discrete symbols is realized by sliding through the time series, considering the slopes of two neighboring segments (or the two neighboring angles between three segments) at a time, and converting them to one character (cf. Figure 6.4) using a 2 dimensional matrix. Converting an approximated time series using this approach will result in a long symbolic string, as shown in Figure 6.3, that can be parsed for motifs with the help of suffix trees. The advantages of this approach are two-fold: First, the length of a linear segment is not constrained to a fixed value, as it is the case with the SAX approach, and common errors where symbols afterwards would need to be merged are avoided. Secondly, by not taking the length of a segment into account when mapping these to symbols, more importance is placed on patterns in the data where strong peak sequences occur. With a symbolic representation of the time series now completed, the next section will discuss the method for the finding of motifs.

6.3.4 Extracting Motifs by Means of Suffix Trees

Having mapped the raw acceleration data to a symbol sequence, motif discovery can now be done by finding substrings that occur multiple times in the target class. This is above all an efficiency problem: searching for all occurrences of every motif in a long string in an exhaustive fashion will result in a slow discovery process that is not scalable, as large sets of motifs are expected to be present.

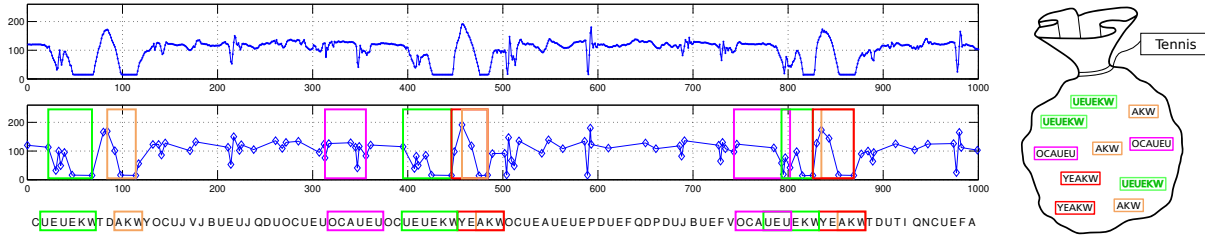


Figure 6.6: The Bag-of-Words classifier is built from the extracted motifs. Hereby, the motifs that also occur outside the target class (i.e. background data) are being retained.

6.3.5 Bag-of-Words Classification

Using the most discriminant motifs during a training phase as described in the previous paragraphs, classification is performed by local evidence of all motifs that support an activity (Figure 6.6). This straightforward bag-of-words classifier uses a sliding time window over the time series and accumulates local evidence by counting occurrences of motifs. As the activities tend to last at least 30 minutes and up to an hour and a half, a window size of 10 minutes was chosen.

6.4 Evaluation

The approach as described in the previous section is in this section tested under the conditions from our motivation scenario of psychiatric monitoring. After presenting the details on our experiment setup, the evaluation participants and the chosen activities, we compare the performance of our method with two commonly-used activity recognition techniques that have been chosen as a base line. After presenting the performance results of our and the other methods, we will use the findings to discuss the applicability of our technique for other activities.

6.4.1 Participants and Target Activities

The data used in the current experiment comes from 33 volunteers who have no known psychiatric disorders and for whom a physical activity was known before the recording phase, which they would do once each day, over the course of a whole working week. Thus, each participant was regularly performing a key physical activity, as it would be chosen by a psychiatrist. For most, this turned out to be a leisure activity or sports, for some a household related activity that was part of their daily schedule.

Table 6.1 gives an overview of all participants, specifying their gender, age and their personally chosen target activity which will be used for testing the detection accuracy of the chosen approaches. In addition to that, the amount of raw sensor data as well as the total number of segments used in the evaluation are given. This is to highlight the significant effect of the first data reduction step: The modified SWAB algorithm, executed with approximation threshold of 10 and buffer size of 80, on average results in more than fifteen times less data points.

Subject	Gender	Age	Target activity	3D samples	Segments
1	male	30	badminton	42 735 422	2 552 512
2	male	32	badminton	43 566 415	2 758 210
3	male	31	basketball	44 055 550	3 116 622
4	female	26	canoeing	35 361 647	2 281 800
5	male	32	cooking	43 553 117	2 923 277
6	male	35	cycling	42 841 897	2 259 414
7	male	30	dancing	43 430 128	2 708 621
8	female	14	dancing	43 203 531	2 491 817
9	female	16	dancing	43 496 339	2 461 287
10	male	20	drums	42 866 629	2 554 723
11	male	31	fishing	28 993 833	1 825 663
12	male	53	fishing	32 495 006	1 957 760
13	female	26	flamenco	43 101 537	2 980 562
14	male	27	guitar	43 313 553	2 259 521
15	female	27	guitar	43 230 164	2 825 311
16	male	23	guitar	43 693 805	2 705 336
17	male	28	gym	34 822 499	2 480 707
18	male	32	gym	43 459 162	2 738 924
19	male	30	gym	43 450 406	2 587 331
20	female	28	gym	43 380 386	2 925 873
21	male	31	ironing	49 274 551	3 267 584
22	female	27	keyboard	46 276 879	3 194 528
23	female	28	knitting	43 453 683	2 677 868
24	male	30	lunch	44 243 749	2 757 975
25	male	25	soccer	43 523 744	3 153 170
26	female	25	squash	42 181 448	3 042 767
27	male	27	squash	43 383 937	3 355 840
28	male	29	streetdance	46 605 778	2 859 799
29	female	30	streetdance	43 572 177	2 975 779
30	male	32	washing car	42 960 708	2 723 290
31	female	28	xbox	43 391 116	2 553 799
32	female	28	yoga	42 991 198	2 705 832
33	female	30	zumba	26 927 159	2 011 826
average				41 934 459	2 687 131

Table 6.1: Each of the participants wore our sensor day and night for about a week. For every participant, one physical target activity was chosen before the study to be performed once every day, leading to the 5 folds in the cross-validation. To illustrate the impact of early abstraction, the raw acceleration samples and the number of linear segments produced after the piecewise linear approximation step are listed to the right.

The dataset from each participant was split into separate blocks of about a full day (24 hours \pm 50 minutes) each to facilitate 5-fold cross validation. Each activity instance, depending on the actual activity, generally lasted between 30 and 90 minutes, except for fishing where the activity instances lasted 4 hours. The target activity on average holds \pm 5% of the entire fold, with the rest being other daily activities.

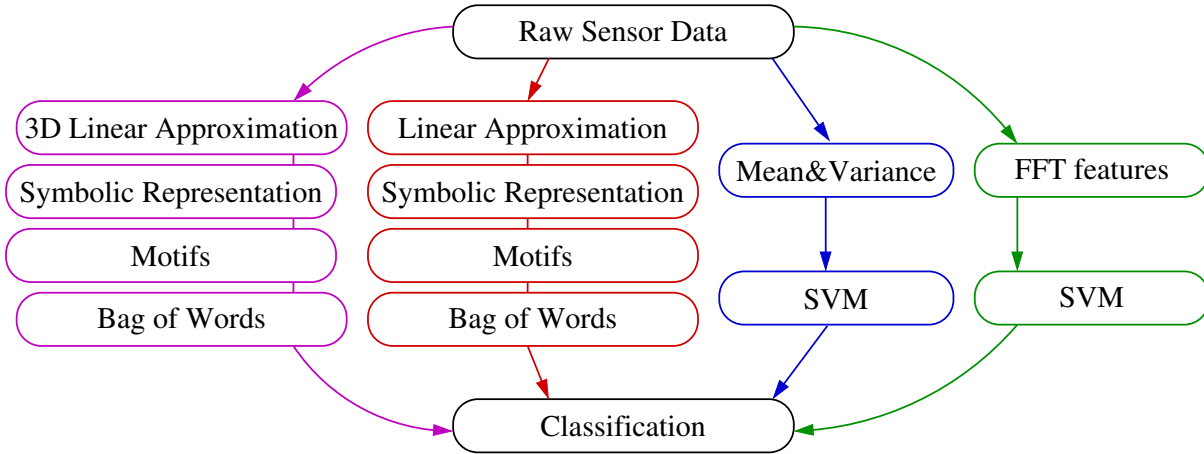


Figure 6.7: Overview of the detection evaluations: The dense motif classifier (red) and its 3D version (violet) are compared with two strong classifiers that rely on mean and variance features (blue) and FFT features (green) respectively.

6.4.2 Benchmarking the Performance

To evaluate the classification performance of our approach, a comparison to two standard activity recognition techniques was done. For the latter, several classifiers were identified, with the Support Vector Machine (SVM) as the best performing, as well as different feature sets to abstract the raw data. Due to their coverage in the activity recognition community, e.g. in (Huynh and Schiele, 2005) or (Zinnen *et al.*, 2009), mean and variance were identified as one combination. An additional set of features based on Fast Fourier Transform (FFT) coefficients were chosen as another: the 16 FFT features that have been suggested and evaluated in (Zinnen *et al.*, 2009) consist of the absolute, real valued FFT coefficients grouped into 4 logarithmic bands, 10 Cepstral Coefficients, the spectral entropy and energy of the signal.

One imbalance in this comparison is illustrated in Figure 6.7: Since the dense motif approach aims at extracting characteristic motion patterns for target activities from the symbolic representation of the original sensor data, more resources are spent on pre-processing the sensor data, and less on the classification. Although Figure 6.7 details just the required steps, and not their time complexity, it is clear that the approaches differ significantly in how the processing steps are weighted.

Figure 6.8 shows the average times gathered during the 5-fold cross validations with our dense motifs approach for one set of parameters (approximation threshold of 10 and buffer size of 80; symbols mapping with 5 bins). For one day worth of data the two abstraction steps (producing segments and converting them to symbols) require about 10 seconds. Depending on the activity, the time required for extracting the characteristic motifs from the training part of the dataset ranges from 1 up to 56 seconds. Obtaining motif occurrences for the classification on the fifth part of the dataset and computing the score needs from 5 up to 275 seconds (with an average of approximately 65 seconds and 70 seconds standard deviation), using a standard laptop setup and with the source code written in Python.

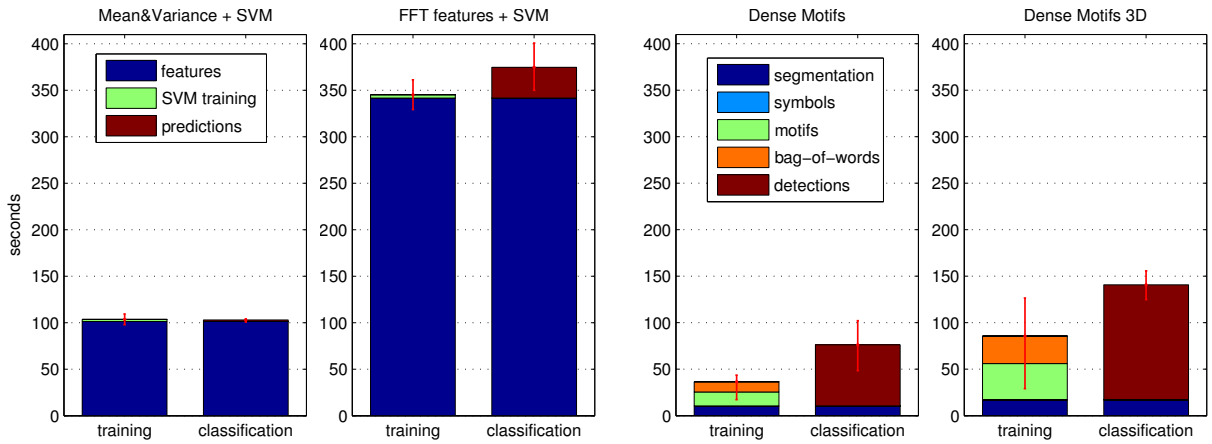


Figure 6.8: Mean execution times in seconds for training and detecting on 24 hours of data with the four approaches, with the upper and lower quartiles (red lines): The dense motif method is especially faster in training, with segmentation and discovery of motif occurrences taking up most of the time. Note that the symbolic mapping step is very fast and therefore not visible on this axis scale. For the SVM-based approaches, most of the time is spent on calculating the features on the sliding window, with the classification done in a few seconds. The same parameters were used as in the classification analysis.

Both mean and variance, as well as the FFT-based features, are computed on a sliding window over the raw data, with window sizes varying from 1 to 30 seconds. For classification, the `svmtrain` and `svmclassify` methods from the Matlab Bioinformatics Toolbox were used. The performance of the features with the SVM classifier was evaluated by the same 5-fold cross validation as for the dense motifs approach. The detections produced by the SVM classifier are smoothed by a sliding window of 10 minutes to filter out outliers (false detections), resulting in a score. At this stage, by evaluating the obtained classification versus the the ground truth annotations, precision and recall are computed for our dense motifs as well as for the features with SVM approaches. Figure 6.9 shows an example illustrating how the different classification techniques performed on the third day of the cycling dataset during the evaluation phase. The score plots below the raw data show the aggregated motif occurrences for the dense motif method, and the normalized results of the windowed filter after SVM classification.

6.4.3 Experiment Results and Discussion

This section presents the experiment results for the leave-one-day-out 5-fold cross validations: For every activity one day is left out for testing purposes, while training (obtaining the motifs that tend to represent the activity) on the other four days. Since the evaluation considered a wide range of possible parameter combinations (abstraction thresholds, buffer window sizes, number of bins for symbolic discretization, suffix tree traversal depth, window sizes, etc.), only a few prolific figures are shown to discuss the experiment results.

Figure 6.10 and 6.11 show a comparison of the best performing average precision and recall figures of our approach (including the two variants with single-axis or multi-

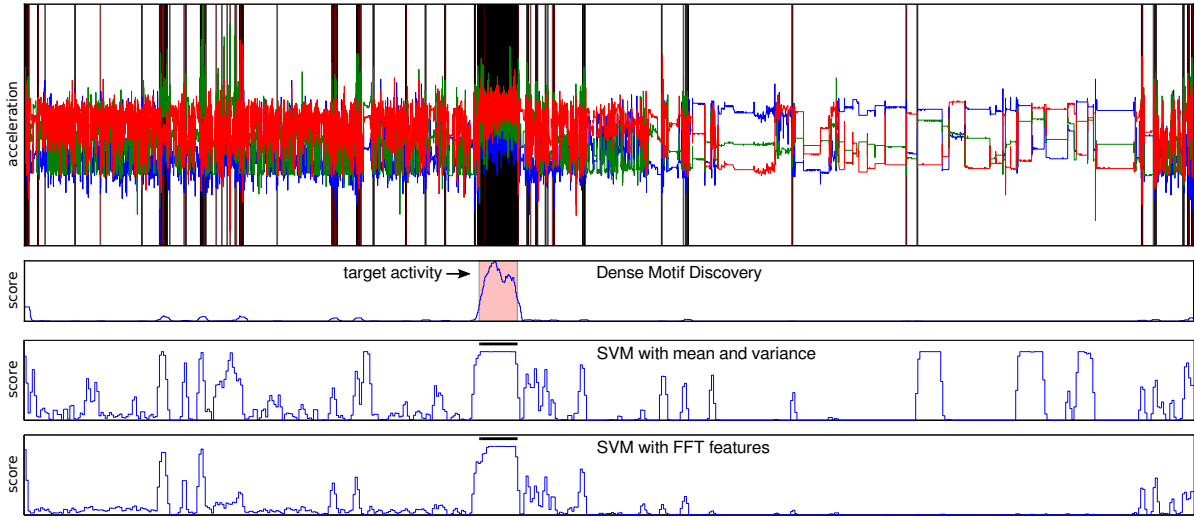


Figure 6.9: One day of the experiment data in which one of the participants cycled for about one hour. The topmost plot shows the original 3D sensor data, along with motif occurrences highlighted by black markers. The three plots below give the corresponding score plots produced by the different classification approaches during the evaluation: The first plot shows aggregated motif occurrences, while the two plots below show the smoothed SVM classification for mean and variance and FFT-based features respectively, with all three approaches using the same sliding window length. After combining all such results for all participants' data, the precision and recall figures show overall performance of the three approaches in Figure 6.10 and 6.11.

dimensional approximation) and the SVM classifier that has been trained with mean and variance, and the FFT features. Additionally, the performance of a random 'guessing' classifier is depicted for completeness. Precision and recall are averaged over the number of folds, while for each activity and classification method the choice of parameters with the best classification performance is chosen.

The SVM classifier trained with mean and variance features performs well on activities that involve a lot more motion, with especially the variance of the signal playing a significant role. Our evaluation with this large dataset proves this reliance, which can specifically be seen by comparing the activities badminton, drums, soccer, xbox playing, squash or zumba with gym, cycling, knitting, flamenco, guitar or dancing activities. While the first set activities exhibit very high accelerations due to sharp hand motions, the latter activities lack such high accelerations.

The dense motif approach is in many cases the best-performing, in some cases even by a significant margin. To illustrate its strengths, the performance on the badminton data is shown in Figure 6.12, including a zoomed-in region on a brief time span of 50 seconds with motif occurrences matching the underlying characteristic motion patterns. Motifs here often overlap, with their dense occurrences making the detection of the activity more reliable.

Overall, dense motifs with 3 dimensional approximation step can not match the recognition performance of the per-axis approximation. While the trajectory in space is abstracted well by linear segments, the symbolic mapping step relies upon the pla-

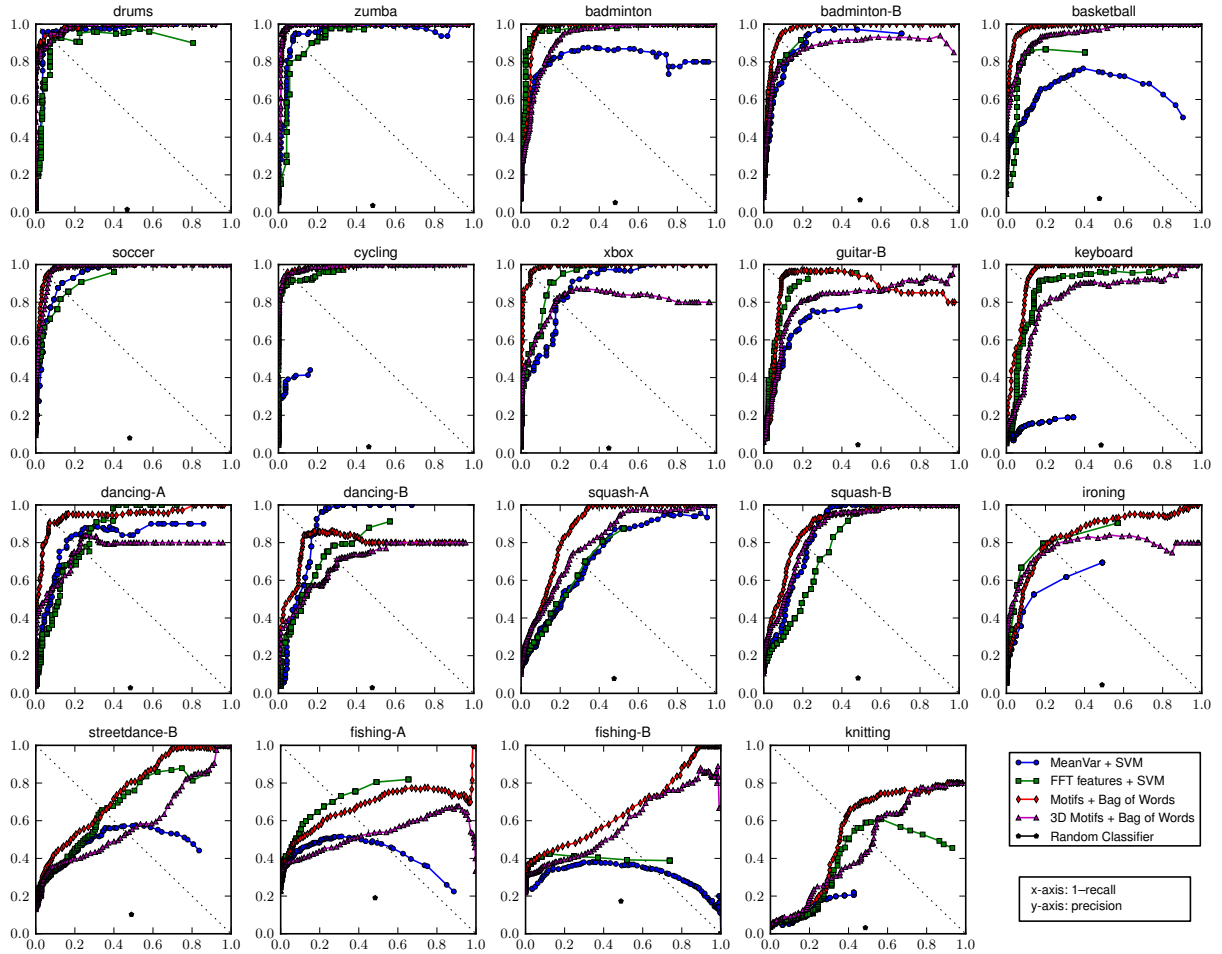


Figure 6.10: Precision and recall performance results on 19 out of 33 different activities obtained through the leave-one-day-out 5-fold cross validation, averaged over the number of folds. Presented here are activities where the different classification techniques performed mostly very well or similarly average. Very good results were achieved for activities with recurring characteristic motion patterns, which is mostly the case when characteristic acceleration peaks can be found in the sensor data. Unequal distribution of motif occurrences for a target activity leads to reduced recall performance.

nar angles between adjacent linear segments, which results in much more information about the 3D trajectory being lost. To preserve this relevant information during this abstraction step, future work should consider alternative mapping approaches that improve encoding of the 3D trajectory.

The precision and recall figures reveal that the detection performance of our approach is both activity and person specific. For example, dancing, street dancing or gym activities from different participants show a significant discrepancy in the performance both for the dense motifs and the traditional approaches (cf. Figures 6.10 and 6.11). With the large dataset containing many different activities, our goal is to generalize the performance results of the chosen approaches to be able to assess for which activities the dense motif technique suits most. When considering these obtained performance figures, we can observe four types of results:

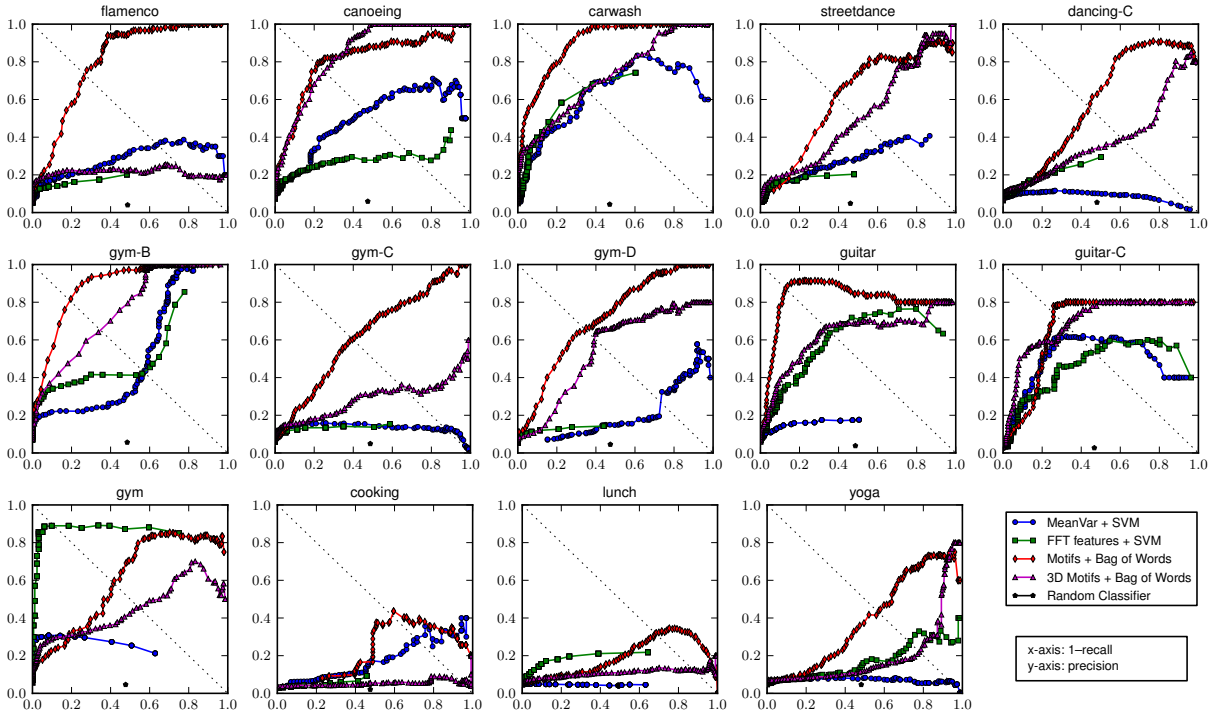


Figure 6.11: Precision and recall performance results on the remaining 14 activities obtained through the leave-one-day-out 5-fold cross validation, averaged over the number of folds. The activities presented here show a more diverse classification performance across the classifiers, with (a) dense motifs significantly outperforming the conventional approaches: flamenco, canoeing, dancing, majority of gym or guitar, (b) FFT being superior on one gym activity and (c) all approaches failing on cooking, lunch and yoga activities.

1. All classification techniques perform mostly “good” or similarly “average”
2. Dense motifs performs significantly better than SVM trained with features
3. SVM trained with FFT-based features outperforms dense motifs
4. All classification techniques perform poorly

Group 1: High-variance activities

The first group consists of activities that involve a lot more motion than all other activities throughout the day. This property can be captured well using both the variance of the signal as well as its frequency characteristics, which results in good performance by SVM trained with these features. Our shape-based approach for such activities is able to grasp the shape of the time series and by accumulating characteristic motifs therefore to perform on par and often better than the two other approaches.

Figure 6.10 contains performance figures for activities where the chosen approaches were able to perform similarly good (except cycling and keyboard, where mean& variance features performed poorly) or similarly average for streetdance-B and knitting.

When investigating the impact of the abstraction, we noticed that the parameters that control the first abstraction step (piecewise linear approximation) have little impact on the classification performance for activities that reach approximately 95% equal error rate (e.g., badminton, drums, zumba or soccer). This can be explained by the impor-

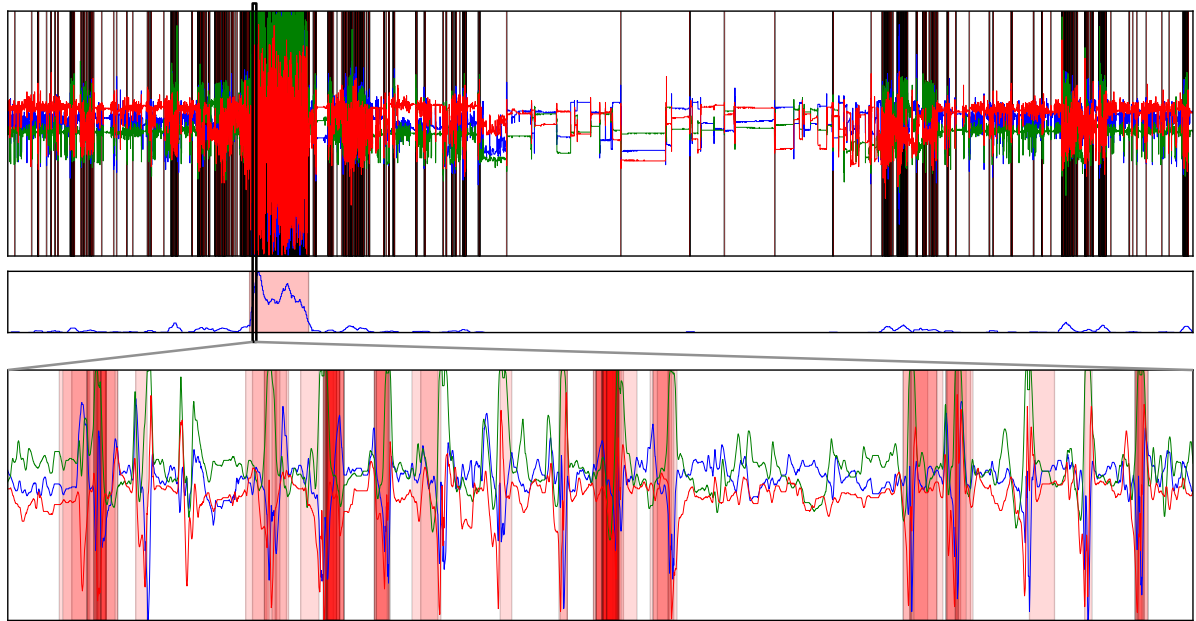


Figure 6.12: Dense motifs performance on one fold of the badminton data over one day (upper plot, with badminton activity highlighted in red), and a 50-seconds fraction thereof with motif occurrences (lower plot). Characteristic motions such as forehand, backhand, smashes, are often marked by motifs, while areas in between tend to be left out.

tance of high acceleration peaks in the signal that are preserved even with a coarse grained approximation.

For activities where a slightly lower classification performance of approx. 80-90% equal error rate was achieved (e.g. carwash, squash, ironing or dancing), the hand motions tend to be less sharp, leading to less high acceleration peaks. The motions become more fluent, but are still fast enough to produce characteristic peak patterns. With a more coarse grained parameters there is a risk of abstracting the sensor data too much, such that the essential shape of the signal is lost, resulting in less good classification. Figure 6.13 shows how varying the main approximation parameter impacts the quality of dense motif classification.

Group 2: Fluent gesture-based activities

This group consists of activities that tend to have less sharp but more fluent hand motions, showing a wide performance difference between the chosen approaches (cf. Figure 6.11). Here, classification performance is much more activity and person dependent, whereby the dense motif approach (single-axis approximation) performed significantly better than its 3D version and the two feature based approaches.

Overall, the classification results could not match the quality of the first group, at best reaching 85% equal error rate for one of the guitar activities. The classification performance ranged from average 55% for ballroom dancing (dancing-C) and 60% for the majority of gym weight lifting exercises, up to reasonable 80% for street dancing and playing guitar. Hereby, the dense motif approach outperformed other approaches

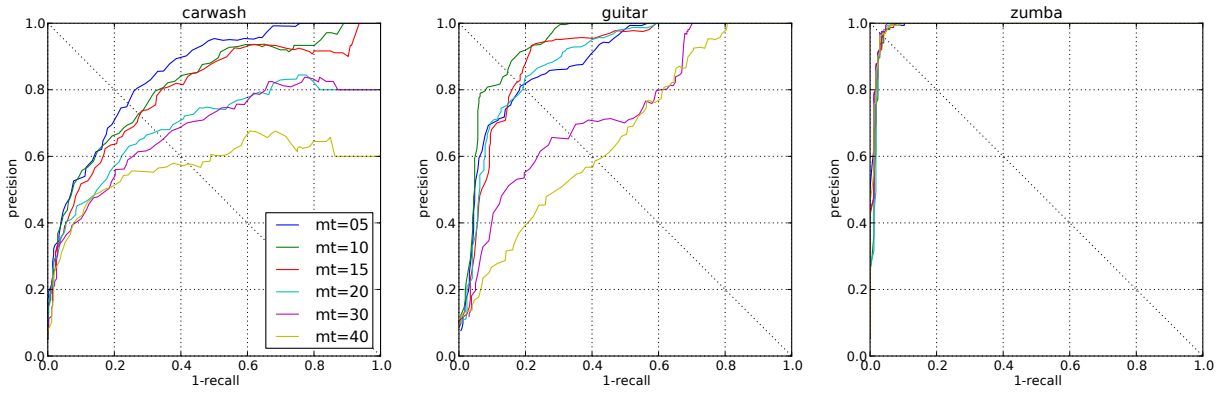


Figure 6.13: These precision-recall plots shows how varying the merging threshold parameter from the first abstraction step impacts the classification performance of dense motif discovery. While for activities with the high acceleration peaks (e.g. zumba) the merging threshold has almost no influence, a more coarse grained approximation for less sharp motions (e.g. carwash or guitar) often leads to characteristic peaks to be removed, thus reducing the classification performance.

by up to 20% in equal error rate, having the largest leap ahead on flamenco, guitar and most of the gym datasets.

For the majority of this groups' activities, the mean and variance features did not perform well. This is not surprising, since the variance in the accelerations is much less discriminant due to less sharp hand motions. This to a large extent also impacts the performance of FFT features, which, due to little periodic motions in flamenco or other dancing activities, are not characteristic enough to produce good classification.

The dense motifs approach benefits the most from the characteristically short motions, which are not equally distributed over the whole activity, such as during flamenco dancing (as shown in Figure 6.14). Similarly, playing guitar data turned out to be captured well with motifs, gaining a significant advantage over the traditional features. Different ways to play were observed, including hitting or plucking the strings depending on musical genre, as well as interruptions of the activity (e.g.: a bathroom break), result in varying density of motif occurrences. While dense motif approach still detected much of these activities, such uneven distribution of characteristic motions leads to a slight drop in the recall and thus in lower detection performance.

Since the motions tend to be less sharp, such that the acceleration peaks are not that characteristic, the choice of the first abstraction step parameters is much more severe. Thus, the multi-step abstraction of the dense motif approach is more reliant on the appropriate parameter settings.

Besides that, the motions during the target activity tend to be much harder to distinguish from background data, such that many potential good motifs are being retained during the training phase. Visual inspection of different folds during the evaluation and comparison to other activities shows that the initial number of motifs is not very high in the first place, and is heavily reduced as motifs that appear in the background data are discarded. When classifying with lower number of motifs, we have observed that the number of motif occurrences outside the target activity has a much higher impact on

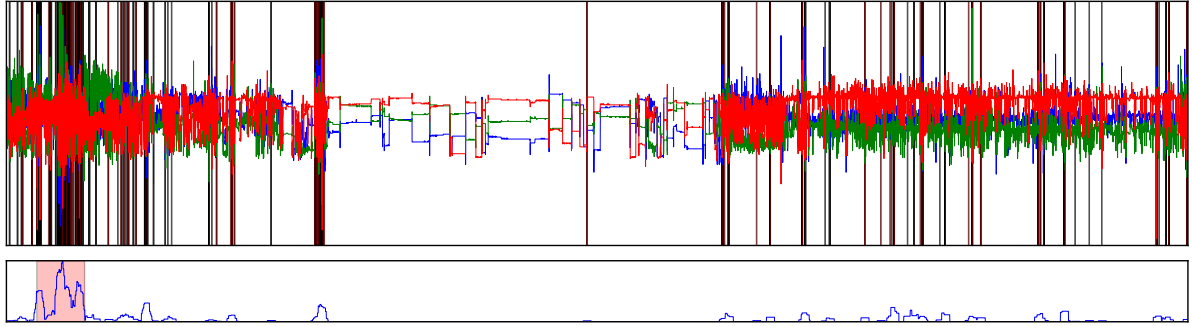


Figure 6.14: Example of dense motifs on a day with the flamenco activity (highlighted in red). The drops in the score for the target class, relate to not equally distributed characteristic motion patterns, resulting in lower classification performance. To improve on these results, additional activity models can be utilized.

the density score (higher density peaks outside the target class), and therefore reduce the overall detection performance.

Group 3: Activities with dominant frequency features

This group consists of activities as in the previous group, with the only difference that FFT features are able to significantly outperform the dense motif approach, reaching 80% equal error rate. Surprisingly, this happened for a single activity only, namely the first gym dataset. For the other three gym datasets, FFT features performed much worse, such that these datasets can be found in group 2.

This result did surprise us a bit, since we have expected a more coherent behavior of the three approaches on gym and similarly slow and periodic activities. Considering this result as a single case outlier would be too far fetched, though. The reason for such discrepancy in performance is most likely to be found in the individual style of carrying out the gym exercises, where the FFT features could profit from their ability in representing periodic parts in the acceleration data. Additionally, the fixed window size for feature extraction may play an additional role: for this activity, it seems, the window size allowed for extraction of feature values that the SVM was able to distinguish. The dense motif approach, on the other hand, is not bounded by a fixed window size, which allows it to perform similarly on all four gym data sets.

In this case, the gym activity consisted of different weightlifting exercises, with gestures being much slower and better characterized by the FFT features than the other three gym activities. Hereby, the exercises seem to be carried out with another frequency, resulting in more confusion with background data and therefore very low (failed) classification performance. Figure 6.15 shows the dense motif performance for this standing out gym dataset, on 24 hours and a sub-sequence lasting for about 2 minutes. The FFT-based features, on the other hand, computed on a window of 5 seconds, were able to profit from the frequency domain characteristics of the gym exercise activity: The SVM classifier trained with these features performs considerably good reaching over 80% in equal error rate.

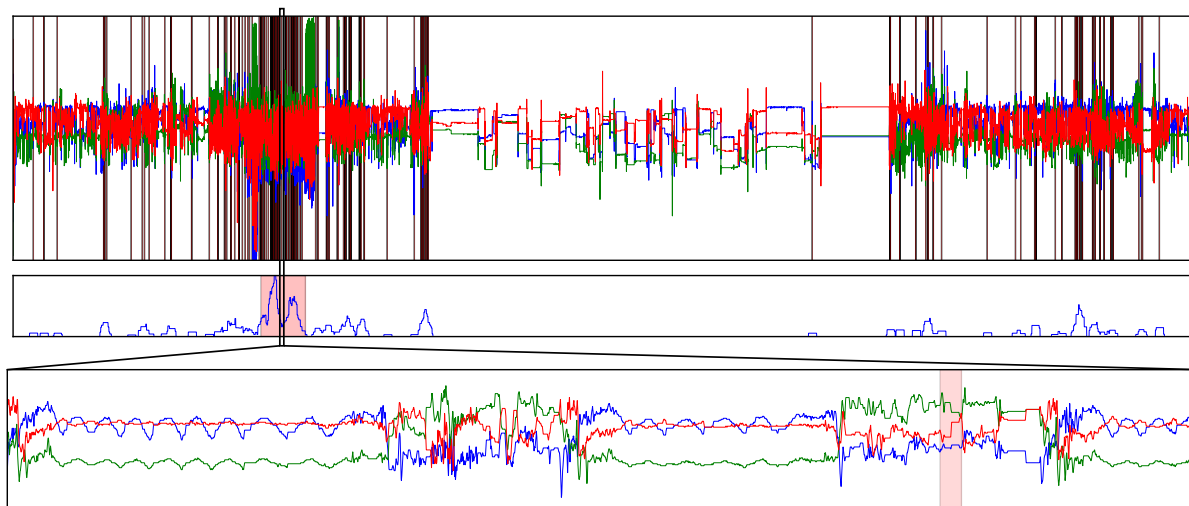


Figure 6.15: Dense motifs for a day of the gym data (upper plot, with gym activity highlighted in red) and a sub-sequence lasting 2 minutes (lower plot, with motifs marked in red). Two of the exercises can be recognized by periodically signals in the lower plot.

Group 4: Activities where more modalities are needed

Finally, the fourth group consists of activities where the chosen approaches were not able to produce meaningful classification. This is mostly because these activities tend to be complex activities, consisting of multiple sub-activities that also tend to be performed in other parts of the day, or activities lacking characteristic patterns.

For cooking, the sub-activities among others would be washing and cutting vegetables, fruit or meat, stirring, baking, roasting, doing the dishes. Besides that, the participants sometimes tend to interrupt cooking for other short-while tasks (e.g. answer a telephone call, or taking care of children). The lunch activity consists of walking to the cafeteria, getting the food, eating, going for a coffee and then walking back to the office. Yoga, on contrary, turned out to be a much more static activity, with very little motion and therefore lacking characteristic patterns required for motif discovery.

For all these activities, neither the variance in the accelerations, nor the frequency characteristics, nor the shape of time-series in the motif discovery were suitable to produce any meaningful classification in our experiments.

Thus, similar activities that do not exhibit characteristic peak patterns therefore remain a hard challenge for dense motif discovery. To assess, whether dense motifs are usable at all, a much more detailed evaluation is required which should evaluate appropriate parameters, specific features and modeling.

6.4.4 Applicability for Other Activities

Overall, we can conclude that our dense motif approach tends to detect well those activities that contain a high amount of characteristic motion patterns. From the performance figures presented earlier we have observed that dense motifs generally shows good results for sports activities, such as (a) racket sports, with characteristic swings of

the racket during the game, (b) ball sports, which include running, dribbling, kicking or throwing the ball, (c) high-paced fitness, such as zumba, running or jogging.

Besides sports activities, our technique was able to reach good classification performance for activities where the participants were playing music instruments. Here, the performance varies on how characteristic the peak patterns are and how reliable the dense motifs approach is able to capture these, depending on the music instrument itself (drums, guitar, keyboard, piano, violin etc.) and the way it was played. While dense motifs performed well on the drum or keyboard datasets (approximately 95% equal error rate), playing guitar has a higher variety (75-90%): By just moving the fingers (plucking the strings) instead of moving the arm (hitting the strings) will impact the recognition results, as the peak patterns in the time series will be significantly different. Nevertheless, the performance advantage compared to the two traditional techniques is higher for more challenging ways of playing (i.e.: guitar-A and guitar-B). One therefore can expect good results from other activities such as sawing or hammering, if the activities' durations are long enough and exhibit recurring characteristic patterns that will lead to motif discovery.

More unpredictable performance was witnessed from gym, dancing or activities with similar kind of motions. These activities, similarly to playing music instruments, are performed by the humans very individually, which directly impacts the classification performance. For gym exercises, the dense motifs approach on 3 out of 4 participants was able to achieve only average results (approximately 60% equal error rate). A much better classification performance on the 4th dataset (reaching 80% accuracy) was reached because of the participant's characteristic warm-up running phase of 30 minutes duration before the weight exercises. The dense motifs method was able to significantly outperform the two other approaches on 2 out of 4 datasets. For one dataset, though, the FFT-based features achieved a detection rate of more than 80% accuracy, most likely due to the way the individual performed his training plan. For the dancing activities, the detection performance increases significantly when the motion patterns tend to produce distinctive peaks in the time series: While the ballroom and Latin dances as well as street dancing datasets performed at a very poor 60%, much better results were achieved for flamenco (73%) or hip-hop and German "Gardatanz" (82-90%). For the first two types of dances, the dense motifs approach did outperform the others by a wide margin. Also, breaks during the activity (e.g., a bathroom break, or answering the phone for few minutes) has shown to impact the density of motifs. If these are annotated and trained as a part of the target activity, there naturally will be negative impact on the classification performance.

Household activities, such as cooking, eating, vacuum-cleaning, or ironing play an important role in activity recognition research that focuses on activities of daily living (ADL). In our experiments, we were able to detect some of these activities (i.e., ironing, knitting, washing the car, going for lunch, cooking), but with a varying accuracy. Hence, for a productive deployment, the medical staff involved would have to carefully chose the ADL of interest. From our observations (80% equal error rate), dense motifs will most likely perform well on the ADLs that exhibit characteristic acceleration patterns, such as ironing, vacuum-cleaning, swiping the floor, cleaning windows, or washing the car. Complex activities, on the other hand, consisting of multiple sub-activities which regularly occur multiple times a day (e.g., "lunch" that includes walking, sitting,

and eating gestures that occur frequently outside the activity) or tend to be short or inherently variable or often unstructured (e.g., “cooking” of different types of menus, different workflows) remain challenging to be detected accurately.

6.5 Conclusions

We presented an activity recognition method designed to spot physical activities in long-term continuous datasets of 24/7, day-and-night recordings from a wrist-worn inertial sensor unit. The approach uses motif discovery to recognize target activities, by means of the most typical gestures occurring in them. This allows the method to find possible matches efficiently in large amounts of data, to pinpoint where a particular activity might have occurred. Through the density of these motif occurrences, our system can thus predict the likely underlying human activity.

This design has been inspired from an on-going deployment in psychiatry monitoring and was evaluated on a considerable dataset under similar constraints. The dataset contains more than 160 days of 3D accelerometer data from 33 participants that wore a wrist-worn sensor unit for approximately a week non-stop. The sensor was recording raw inertial data at 100 samples per second, resulting in more than 1.383 billion 3D acceleration samples.

Our approach relies on early data abstraction as described in Chapter 4: The discretization of the abstracted time series to its symbolic representation is then performed in order to utilize efficient search-oriented data structures, such as suffix trees or suffix arrays and to extract recurring patterns (motifs) that are characteristic for specific activities. Based on these motifs, we are able to create a bag-of-words classifier that relies on the density of motif occurrences, instead of complex and computationally expensive modeling techniques.

Our experiments show that the approach is able to detect many physical activities, at an accuracy on par with standard feature-based recognition approaches, hereby reaching an equal error rate performance of more than 90% for 11 and more than 80% for additional 7 of 33 activities. Our approach was able to outperform these baseline approaches on 30 of 33 datasets, whereby performance advantage was achieved on 12 datasets, while being significantly inferior only on one. We demonstrated that the approach is able to work on large datasets with continuous inertial data, which thus allows scaling for true long-term deployments over months at a time.

The amount of participants and gathered data allowed for a detailed evaluation of the system’s performance potential with regard to physical activities. Experiments have shown that the presence of characteristic gestures is crucial for the system to produce usable results. For activities with slower movements, such as particular weight lifting at the gym or dancing, this reliance turns out to be a weakness, as these motions are often not picked as motifs. This result is specifically important for the psychiatric monitoring scenario, because the choice of an appropriate activity is crucial to gain a more detailed insight of patients’ behavior and the impact of the disorder on its quality.

The next chapter specifically considers this aforementioned weakness and proposes an additional sensing modality that allows to identify object interaction and thus improves activity recognition performance literary in hardware.

7 Detecting Interactions with Efficient Wrist-worn Sensors

Some events, such as a high variety of household activities, are too complex to be determined from just single 3D accelerometer. The fact that a human often uses or interacts with objects while performing activities yields an information gain. When considering object detections as characteristic evidence for activities (Figure 7.1), this information can significantly improve the quality of activity recognition from acceleration data, as presented in previous chapters. This approach is specifically interesting for a class of activities the detection of which solely from accelerometer data is hard, and addresses the problem much earlier on the hardware level by introducing appropriate sensing modalities.

The use of a wrist-worn sensor that is able to read nearby RFID tags and capture the wearer's gestures has been suggested as a way to both detect the objects we interact with as well as to identify the interaction. Making such a prototype feasible for longer-term deployments is far from solved, however, as plenty of challenges remain in hardware, embedded algorithms, and the overall design of such a device.

This chapter presents several of the challenges that emerged during the development of a functioning prototype that is able to sense interaction data. We focus in particular on RFID tag reading range optimization, efficient data logging methods and meaningful evaluation techniques for long-term deployments.

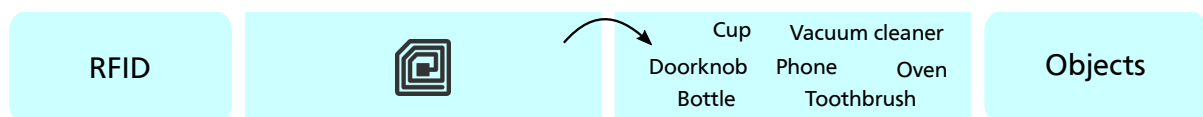


Figure 7.1: Detecting object interaction with wearable wrist-worn RFID readers yields significant information gain. With regard to long-term activity recognition, RFID tag detections provide evidence to be used to improve classification.

7.1 Motivation

With the introduction of small and inconspicuous RFID tags, wearable tag readers have been proposed as early as 2000 for the detection of objects we interact with (Schmidt *et al.*, 2000). A multitude of work, such as of Boronowsky *et al.* (2001); Philipose *et al.* (2004); Patterson *et al.* (2005); Hodges and Pollack (2007) and many others, evolved designs from glove-based prototypes to sleek bracelet-like designs such as the Intel iBracelet (Fishkin *et al.*, 2005; Logan *et al.*, 2007).

The combination of sensed RFID tags and inertial data for detection of what gestures are performed with the objects, has been mentioned and explored in a large body of

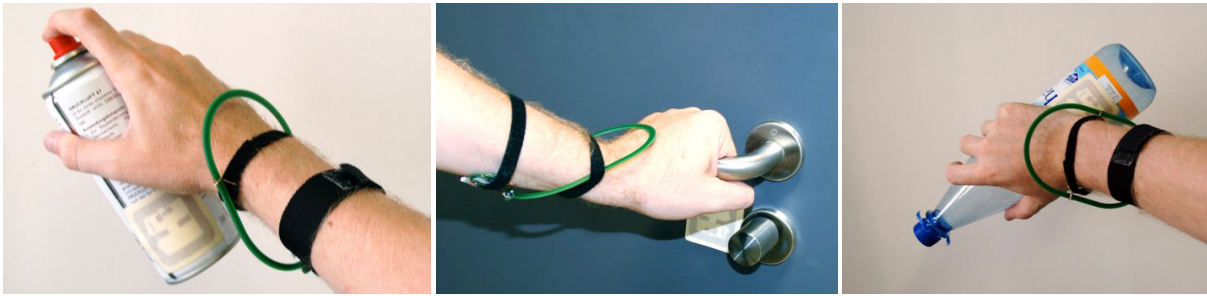


Figure 7.2: The wrist-worn prototype identifies grabbed objects and the physical interactions with them, using a combination of RFID and inertial sensors. The aim is to explore light-weight and power-efficient solutions in particular, to facilitate long-term deployment.

work in years after. In one of the earlier articles, Wang *et al.* (2007) showed that the characteristic motion patterns in the inertial data, combined with the knowledge of which tools or objects were grabbed by the user, gave in many cases very good results in the recognition of various daily activities such as “brushing teeth” or “making tea”. Other work such as MIT’s ReachMedia by Feldman (2004) combines inertial sensors and an RFID reader to detect particular on-the-move interactions from mobile users.

Applications suggested for such a wrist-worn sensor range from medical applications, e.g., detecting Activities of Daily Living to follow the routines of the elderly who are living independently (Fishkin *et al.*, 2005; Stikic *et al.*, 2008), to generic mobile interfaces that are designed to be more natural than existing mobile interfaces, e.g., an input device for wearable computers (Feldman *et al.*, 2005). One of the aspects of this work remains underdeveloped, even though it is a main requirement for the acceptance of wrist-worn RFID-accelerometer-sensors in these applications: a battery-driven device at that location needs to be both light-weight and power-efficient enough for long-term deployments.

Our work specifically focuses on the challenges that were encountered during the development of a bracelet that detects interactions performed with detected objects (Figure 7.2), while focusing on a low-power solution that is deployable over long stretches of time.

The main contributions of this work are as follows: First, we mention the technical procedure in designing and optimizing a wrist-worn RFID antenna. Second, a benchmark is presented that allows evaluation of different antenna configurations. We then report on experiments where this prototype was deployed in a gardening scenario for few hours as well as in a domestic setting for several days.

7.2 Combining RFID reading and Inertial Sensing

The basic principle of RFID-tag reading is that a reader is able to power nearby tags by induction, using relatively large antennas. Having the tag reader at the wrist means that grabbed objects and tools can be detected using just these tags, assuming the reader’s range is large enough to power and communicate with the object’s tag. Since the reader needs to be mounted at the wrist, this type of wrist-mounted RFID sensing comes with

a harsh energy constraint: the antenna and its circuitry need to be strong enough to detect hand-held tags, yet power-efficient enough to not drain the battery after a short while.

Accelerometer sensors have been widely used for the recognition of physical actions that have characteristic motions or postures. Zinnen *et al.* (2007) propose the characteristic wrist positions and motions to detect short actions in a car scenario such as “pulling the handbrake” or “opening the oil tank”. In our prototype, after detection of a *claw hammer* one might for instance expect the actions “hammering” or “pulling out nails”, depending on the gestures detected with the wrist-mounted accelerometer.

Since accelerometers are known to be power-efficient but their data to be not as rich, compared to IMUs containing also gyroscopes and magnetometers, the knowledge of what object the user has grasped can help here to distinguish between a limited set of interactions. Furthermore, object detections can significantly narrow down the amount of data in which an automated search for motion patterns is executed, thus speeding up data analysis.

To summarize, the two sensing technologies have proven their worth in earlier work where prototypes were used in feasibility studies. Several questions remain regarding their operation in applications which require functioning longer than a few hours. We hereby focus specifically on the RFID modality. In particular, the following questions emerge and are addressed:

- How can the working range of an RFID antenna be increased to reach from wrist to the object in the hand?
- How can we compare different types of antennas and decide which performs better?
- How dense should RFID samples be to detect grabbed objects, yet save energy?

The basis for this chapter’s wrist-worn sensor design is a custom data logger platform, with its current version presented in Chapter 3. The accelerometer-based module allowed power-efficient capturing of inertial data, enriched by light sensors, a temperature sensor, along with a precise time stamps. It is able to log the obtained samples on a small microSD card, or wirelessly transmit chunks of sensor data to a nearby station via Bluetooth.

For detecting and reading nearby RFID tags, the M1-mini from SkyeTek RFID reader was chosen. This exact module is used by other researchers (Philipose *et al.*, 2004; Feldman *et al.*, 2005; Wang *et al.*, 2007) as it is one of the smallest and power-efficient modules on the market. The M1-mini comes with a small on-board antenna which has a reach of about 3 centimeters. To have a reading range larger than that, an external antenna and a matching circuit are required. Figure 7.3 shows the current version of the bracelet design, with the oval-shaped PCB antenna and matching circuit, the M1-mini RFID reader, and the data logger on top of each other (without the battery).

7.3 Optimizing RFID Reading Range

One of the most time-demanding efforts in building a wrist-worn RFID sensor is the optimization of the bracelet’s built-in antenna. Although Want (2006) gives a very good

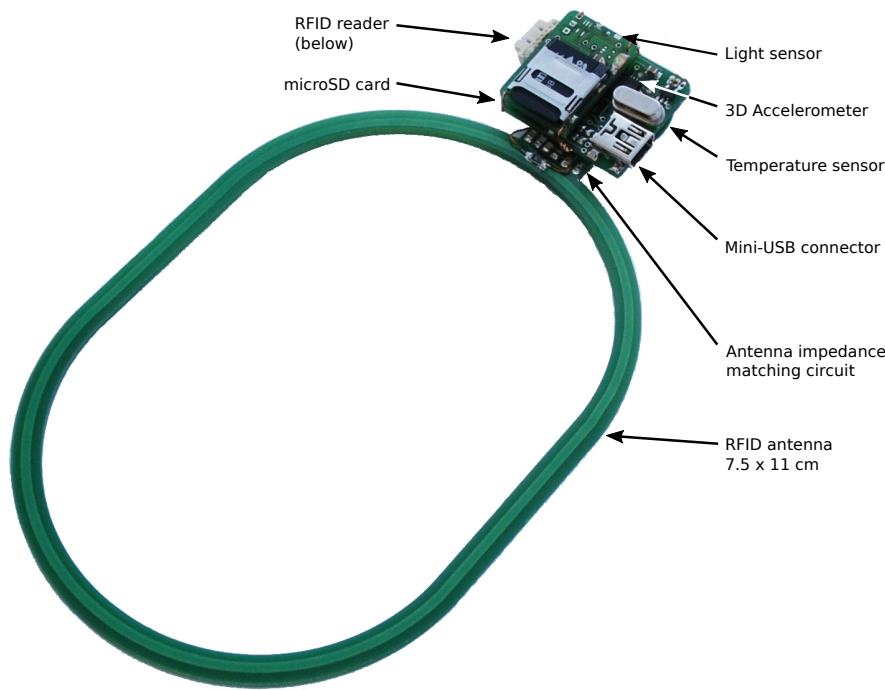


Figure 7.3: The data logger with the custom bracelet antenna and a small RFID reader.

introduction in the use of RFID tags and readers in pervasive computing applications, it requires a deeper knowledge to maximize the potential of a given antenna. Fishkin *et al.* (2005) describe a custom bracelet with integrated RFID reader and coil, but the authors provide no details about their engineering and tuning processes of their antenna. As the reading range for RFID tags is crucial in this type of work, this section is dedicated to a detailed description of the antenna design and tuning process.

The Q -value is a measure of “quality” for the antenna and is directly related to the reading range. Generally, increasing Q will result in a higher power output of the particular antenna, allowing higher reading distances. On the other hand, a too high Q will conflict with the band-pass characteristics of the RFID reader. Since a sufficient bandwidth for the wireless communication between the reader and the RFID tags has to be assured, the relation between the quality and the bandwidth is thus reciprocal, resulting in a trade-off between the quality of the antenna and the bandwidth.

The M1-mini SkyTek reader (Figure 7.4a) can be equipped with an external antenna, which in our case is a requirement as the built-in antenna does not reach further than a few centimeters. The driver impedance of the reader is typically set to $50\ \Omega$. Every antenna has its own impedance value, and to be able to use our customized single-loop coil with the M1-mini reader, the antenna impedance has to be matched to the reader’s impedance front end. There exist different matching approaches such as gamma, transformer or capacitance matching, whereby the latter was used in our work.

The matching circuit, as shown in Figure 7.4b, consists of a serial capacitor C_s , a parallel capacitor C_p and a parallel resistor R_p . The values chosen for those capacitors and the resistor have a great impact on the quality and the bandwidth of the antenna. Based on the experts work of Soffke (2007), the following approach is used to compute

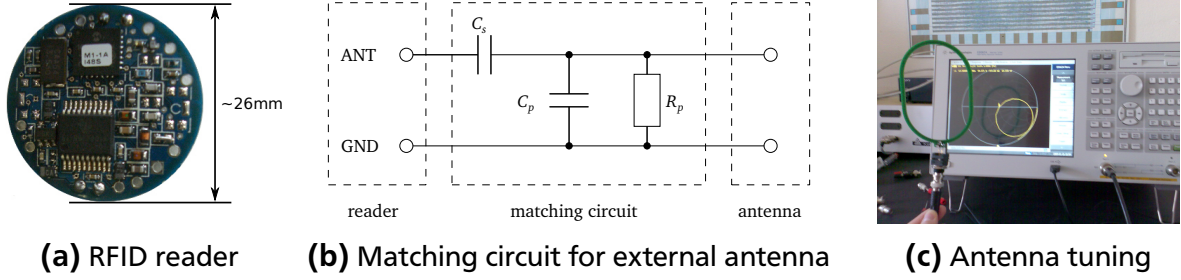


Figure 7.4: The Skyetek M1-mini RFID reader used in our setup has an internal antenna with a maximum reading range up to 4 cm. To increase the reading range, an external antenna can be attached. Since the external antenna has to match the 50Ω impedance level of the reader, a matching circuit (b) is required. The fine-tuning of the circuit was done using a network analyzer (c).

the values for the capacitors and the resistor, whereby the relevant variables are: the measured antenna inductance L , the target impedance $Z_0 = 50\Omega$, the Q-value of $Q = 30$, and the frequency of the field $f = 13.56\text{MHz}$.

First, we compute the parallel resistor value:

$$R_p = Q * \omega * L \quad \text{with} \quad \omega = 2 * \pi * f$$

Then, using the equations

$$G_L = \frac{1}{R_p} \quad \text{and} \quad B_L = \frac{-1}{\omega * L}$$

we compute the parallel and serial capacitor values

$$C_p = -B_L - \sqrt{\frac{G_L}{Z_0} - G_L^2} \quad \text{and} \quad C_s = \frac{1}{\omega * Z_0 * \sqrt{\frac{1}{Z_0 * G_L} - 1}}.$$

Using a network analyzer for our PCB antenna prototype an antenna inductance of $L = 322\text{nH}$ has been measured. Following the given approach the optimal values for the resistor and the capacitors have been found to be $R_p = 820.288\Omega$, $C_p = 371.663\text{pF}$ and $C_s = 59.807\text{pF}$. After utilizing the off-the-shelf available 820Ω resistor and assembling a combination of fixed and variable capacitors to meet the computed values, we used the network analyzer again, as shown in Figure 7.4c, to fine-tune the antenna to match 50Ω . Finally, a reading distance of up to 14 cm has been achieved, which slightly exceeds the reading ranges of the antennas mentioned by Fishkin *et al.* (2005) and Feldman *et al.* (2005).

Surprisingly, evaluating the antenna's reading range by increasing or decreasing the distance from tag to antenna, is not the best way to benchmark the accuracy in detecting tags. Starting with the RFID tags close by the antenna and increasing the distance gradually, will lead to more optimistic reading ranges, since the tags that are charged up by the reader are able to bridge some additional distance. The circumstances when wearing the bracelet and grabbing tagged objects tends to differ substantially from these lab tests. We propose a novel benchmark that will be presented in a later section in this work.

7.4 Optimizing RFID Reading Frequency

Every reading cycle, where the SkyeTek M1-mini RFID reader module seeks for the nearest tags, according to the datasheet tends to last from 20 in best case up to 68.4 milliseconds in worst case, depending on the tag type and whether the reading was successful. The reader draws approximately 60 mA in current while in active, 15 mA in idle and 60 μ A in sleep state. This means that a trade-off exists between detection speed and power consumption: More frequent searching for nearby tags means that objects are likely to be found faster yet at the cost of higher power consumption, whereas less frequent reading will make the battery last longer, but might result in missing tags or a slower detection.

Assuming that we want to search for nearby tags 16 times per second, we end up with a time slot of maximally 62.5 milliseconds for one reading. Since the reader also needs time to go to sleep as well as wake up, and additionally the worst case time for one seek even exceeds this time slot, no idle or sleep states are possible in this scenario. This means that the RFID reader will constantly consume power in active state, i.e., a draw of 60 mA. With a light-weight rechargeable battery with a capacity of 600 mAh, and assuming the rest of the bracelet consuming approximately 10 mA (in worst case), this will result in a runtime of about 8.5 hours.

Reducing the reading rate to 1 reading per second allows the RFID reader to go into the sleep mode to preserve power. In the worst case, it will take approximately 70 milliseconds to search for a tag and up to 100 milliseconds for switching modes, giving the RFID reader roughly 730 milliseconds for staying in the sleep state. Since in our approximated calculation we are able to neglect the power drain of 60 μ A during sleep, the runtime for the same battery and same configuration mentioned above will account to about 28 hours. This is more than thrice the runtime as in the high frequency reading case where the RFID reader's power efficient sleep mode can not be utilized.

In addition to the idle and sleep modes between readings, it is also possible to have the RFID reader change into sleep state and suspend the reading if no movement is registered by the bracelet's accelerometer. For example, no movement will occur for longer periods during the night when the participants wearing the bracelet is sleeping, or when the bracelet has been temporarily taken off the wrist. Combining the sleep states of the RFID reader and the main data logger in such situations will save battery power and result in a longer total runtime. Assuming the night phase to last about 7 hours and neglecting the power consumption during this period, the total runtime with the same battery will account to 35 hours. Utilizing a battery with slightly higher capacity will allow to deploy the bracelet for two full days without the need for recharging.

7.5 Evaluation of RFID Reading: The Box Test

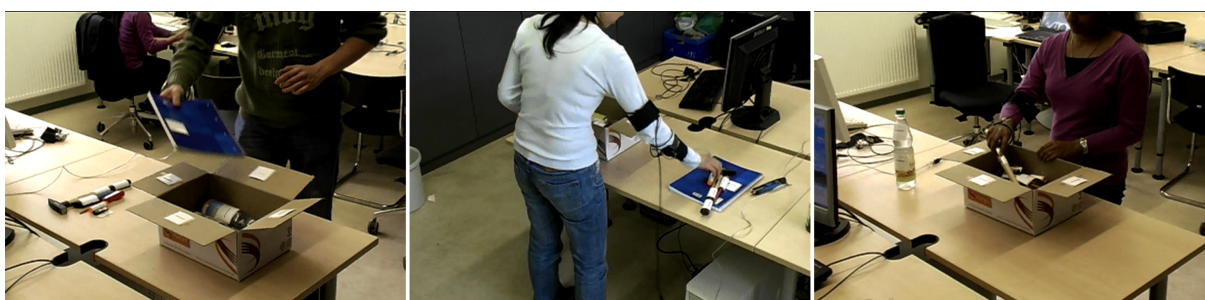
To obtain the optimal RFID parameters, both reading range and frequency, we designed a benchmark in which subjects wear the bracelet prototype and load a variety of tagged objects in and out a box, which is also tagged with several RFID tags. The advantage of this test over straightforward measuring increasing or decreasing distances between a test tag and antenna, is that the whole system is immediately tested under realistic



(a) Some of the tagged objects used in the test

objects
usb stick
glass bottle
hammer
utility knife
lipstick
plastic bottle L
plastic bottle S
screw driver

(b) All objects from the test



(c) Frames from a video recorded during the "box test".

Figure 7.5: Benchmarking the RFID antenna: Various subjects who participated in the box test had to load and unload a box with tagged objects. The objects were chosen to have a high variety in shape, weight and material.

circumstances. This test has the user's wrist and hand present in the middle of the antenna, and the amount of objects, as well as the speed at which objects are grabbed and released tends to be challenging enough. The subjects loaded and unloaded the box three times, and closed it each time it was loaded with all objects.

Furthermore, this test can employ a variety of objects with a wide range of properties that might impact later usage. They can be chosen to fit target applications, or they can be selected according to a variety of shapes and materials. Some of the objects are illustrated in Figure 7.5a, and the full list is given in Table 7.5b. Tags were mostly placed on the areas where people tend to grab the objects. Additionally, 8 tags were also placed on the flaps of the box to detect the closing and opening of the box. To be able to interpret sequences where things went wrong, all experiments were recorded on video (Figure 7.5c shows some frames from the video footage).

To evaluate how well an object was recognized in the box test, we count the number of hits, events when the RFID reader correctly found an object's tag as it was taken in the test subject's hand, and divide this over all occurrences when the tag should have been detected (excluding the tags affixed to the box). This measure will in the remainder of this chapter be referred to as *hit rate*.

The box test was performed for the shape of the antenna, the Q value, and RFID reading frequency, with the results being presented in the following paragraphs.

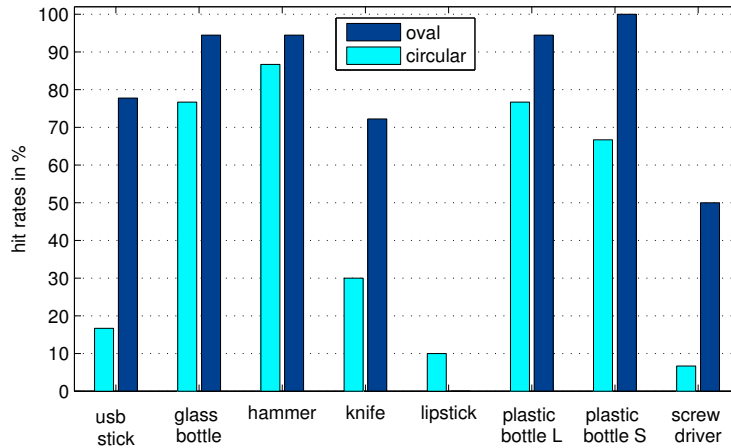


Figure 7.6: Hit rates for several objects during the box test, using two antennas: one circular and one oval-shaped, worn at an angle and tuned to $Q = 30$.

Antenna Shape

Figure 7.6 shows the hit rates between two types of antennas we tested in the early design phase: one PCB antenna which is round, as mentioned in (Wang *et al.*, 2007), and one of the new design that is oval in shape, whereby it is worn tilted at a downward angle, as shown in Figure 7.2. The latter is only slightly larger, but makes it easier to put on the antenna as a bracelet.

A third type of antenna, which uses flexible coils and snap-on metal buttons to close the antenna loops, such as presented by Feldman *et al.* (2005), is more promising for comfort reasons. Unfortunately, it is much harder to implement and unpredictable to design a matching circuit for, as its shape tends to change, and thus was not considered after initial experiments.

Using our oval-shaped antenna design over the circular one, the hit rates improved for all but one object, namely the lipstick. Substantial improvement can be observed for the USB flash drive, the utility knife and the screw driver.

Q-Value

The antennas were evaluated also regarding the quality parameter, by being configured with Q values up to 36. Figure 7.7 depicts the hit rate results for four different Q values for our oval-shaped antenna prototype.

Aiming at the maximization of the reading range to be able to detect tags on grasped objects, the box test benchmark revealed that a value of $Q = 30$ gives the best trade-off between the quality and the bandwidth of the antenna prototype.

Note that also in this test detecting the lipstick did not work well. One of the possible explanations would be that the RFID tag attached to the lipstick case (see Figure 7.5a) had to be bent. This negatively impacts the electrical charging of the tag through the inductive field generated by the RFID antenna. While the tag on the screw driver was bent also, but not to such extent, the effect is also visible.

RFID Reading Frequency

To find out at what frequency the RFID reader should wake up from its sleep state to seek for nearby tags, an experiment with a reading rate of 16 Hz was conducted.

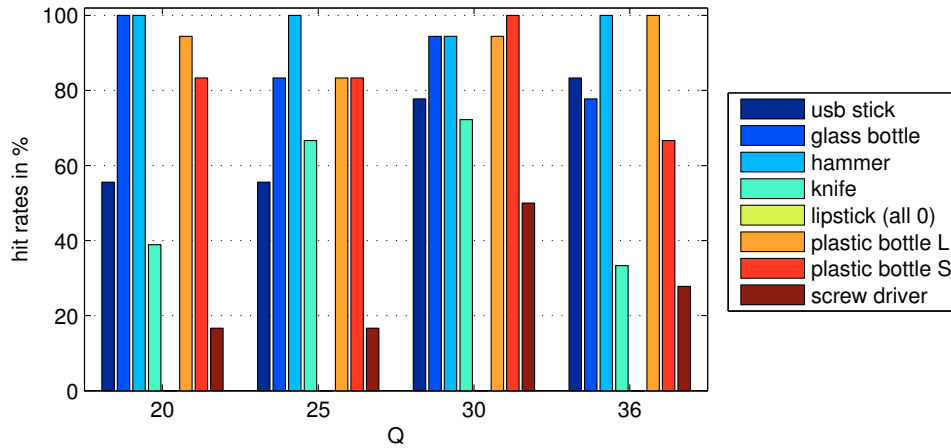


Figure 7.7: Hit rates for different values of Q , using the oval-shaped antenna. The hit rates for lipstick are 0% for all Q 's, and therefore are not visible in the plot.

To simulate less frequent readings, the data stored during that experiment was selectively analyzed with an increasing step factor. By doubling the step factor, and this way halving the reading rate, we obtained a monotonic decreasing number of captured tags. The average hit rate started at 100% for the factor 1 (16 Hz), and dropped to an acceptable level of 65% for the step factor values of 4, 8 and 16 (4 Hz, 2 Hz and 1 Hz respectively). Increasing the step factor even more resulted in the hit rate monotonously dropping further. From this we can conclude that a reading rate of 1 Hz provides a good balance between capturing the deployed tags on the one hand and saving a significant amount of battery power on the other. This frequency is on par with optimal frequencies found in related work by Philipose *et al.* (2004); Wang *et al.* (2007).

The value of the box test goes beyond a more realistic evaluation of wrist-worn RFID antennas. It offers insights in how well and how fast intended objects are found, while it places the prototype in an environment that is closer to its usage scenario, without demanding a costly experiment setup. For completeness, we also mention the distances reported in previous work as well as our measured maximum range in Table 7.1. Note however that not all of these measurements have been taken in the same situation and environment.

Project name	Maximum range	Amplification	Embedded in
iGlove (Philipose <i>et al.</i> , 2004)	3-5 cm	no	glove
SonMicro (Medynskiy <i>et al.</i> , 2007)	4-5 cm	no	glove
Phidgets (Hodges and Pollack, 2007)	10 cm	no	glove
ReachMedia (Feldman <i>et al.</i> , 2005)	10 cm	yes	wrist band
iBracelet (Wang <i>et al.</i> , 2007)	10 cm	no	bracelet
Oval-shaped design	14 cm	no	bracelet

Table 7.1: Several projects involving glove- and bracelet-based RFID readers, and their reported maximum reading ranges. We stress that a comparison between these distances is hard to do, and offer the box-test as a better estimation of how well a design fits the intended application.

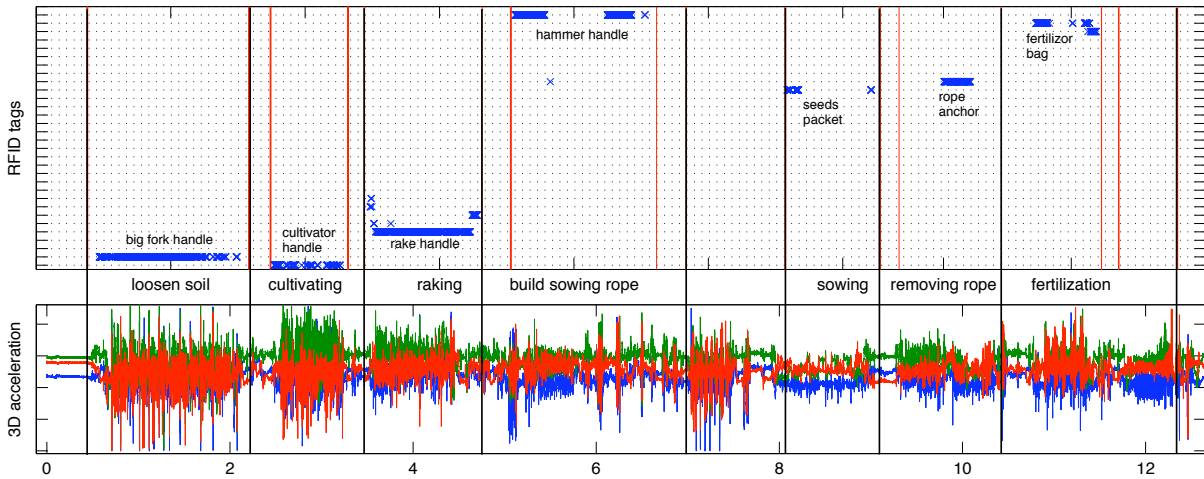


Figure 7.8: Raw data from both the RFID reader and accelerometer embedded into the bracelet during the sowing scenario from the gardening data. The top plot shows blue crosses whenever a tag was detected at a specific time (x axis) for a specific tag (y axis), while the bottom plot displays the accelerometer time series (approximately lasting 12 minutes). Note that one of the main object's tags was found for most interactions, and that at several occasions (e.g., raking) unrelated tags were detected.

7.6 Application Scenarios for Combining RFID and Inertial Sensing

This section considers multiple application scenarios as examples of activities where detecting object interactions can significantly improve activity recognition.

Gardening

To stress the interaction with tools and objects, a new data set was recorded which utilized both detected RFID tags, at 1 Hz, and raw accelerometer data at 100 Hz, where an hour-long gardening scenario was followed. This experiment was performed as realistic as possible, taken outdoors in a real garden while planting real flowers and sowing real seeds.

A gardening scenario is sensible to evaluate a range of interactions with objects, since many tools are typically required for the execution of a wide range of tasks. Tools and objects are also used in several ways, depending on the task at hand: the small spade for example was used both to “dig holes”, “create trenches” and to “firm the soil around plants”. A total of 16 different gardening-specific objects were tagged, including spades, rakes, shovels, flower tops, and buckets, with 36 tags being deployed in total. The tools and objects were positioned near each other throughout the recording, resulting in several instances in which a tagged object was moved out of the way and providing our bracelet with a ‘false hit’ (e.g., a *spade* was detected during sowing). The visualization in Figure 7.8 shows the raw data during the last gardening scenario. Annotation of the sensor data was based on the previously defined “gardening activity” script as well as video footage recorded during the experiment (cf. Figure 7.9a).

Three different scenarios were followed: one where a bed of flowers were planted in soil, a second where weed is removed and plants are watered, and a third in which

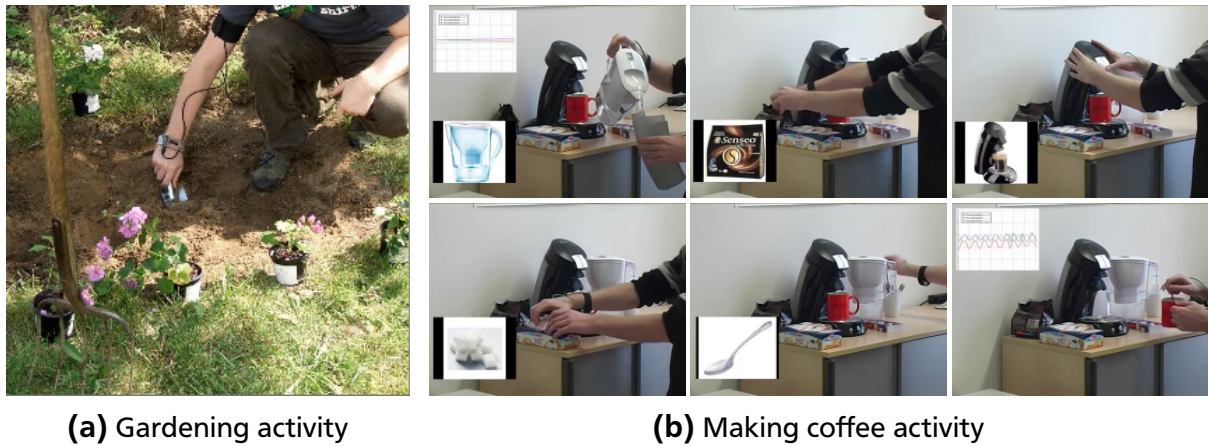


Figure 7.9: a) A photo taken during the gardening experiment, where the user has been wearing a bracelet with acceleration and RFID sensors. b) A set of frames from a demonstration video with a subject performing a “making coffee” activity that involved multiple objects shows detected objects and acceleration data (in overlays) in real-time. Combining acceleration data with detected objects gives more insight on the performed activities and can significantly improve activity recognition.

vegetable seeds are sown. Each of the three scenarios contained a series of sub-tasks, such as “loosening soil”, “digging a hole”, or “hammering in a sowing line”. Video footage was taken for post-annotation of the data.

Aiming at detecting patterns of interactions with a multitude of tools in this experiment, we rely upon piecewise linear approximations to reduce the accelerometer data and to preserve the essential shape of the signal. Hereby, the approximation performance of our mSWAB algorithm is also on this data set consistent with the evaluation results presented in Chapter 4.

Making Coffee

A multitude of short-lasting activities often involve interactions with a relatively large set of objects, whereby little characteristic motion patterns occur. Detecting such actions or activities solely from acceleration data can be considered to be very hard, especially when considering long-term continuous acceleration data.

The “making coffee” activity was recorded to demonstrate the additional information gain for activity recognition. Figure 7.9b reveals that detecting the objects water filter, coffee pack, coffee machine, sugar, spoon, and cup, yields an information foundation to consider underlying human motion. Hereby, only few characteristic motions were identified in the experiment, with “pouring water” in the beginning and “stirring” at the end being most prominent.

Long-Term Domestic Test

In order to characterize the performance of our system in a relatively long-running experiment, we deployed our sensor bracelet for an entire day, for three consecutive days. During this time, a test subject performed daily a set of 11 household activities such as “making the bed”, “polishing shoes”, “vacuuming”, and “sweeping the floor”. Fig-



Figure 7.10: The four photos on the left show some of the activities conducted during the long-term domestic study, here: watering the flowers, cleaning windows, ironing, and vacuum cleaning. The photo on the right shows the objects from the longer-term domestic setting study. Tagged objects and tools were chosen for their interactions, whereby 43 tags have been deployed, and scattered around the living environment of the test subject.

Figure 7.10 shows some more examples for activities and the objects that were interacted with during these three days. In total, 29 objects were tagged with 43 tags.

Our long-term logging target was met as the prototype was able to capture all data – both accelerometer data and RFID data – for this extended period, while the majority of the objects’ tags were detected during each interaction. Although the battery was left charged overnight, the longest continuous log still lasted 18 hours and the bracelet’s lightweight battery was never drawn to depletion. Our estimations are that on a lightweight battery of 600 mAh, the prototype would be able to log at least for two days continuously, assuming a highly active daily schedule.

The main obstacle for several days’ worth of *continuous* logging is currently the form factor of the sensor. Before and during the development of the bracelet we held short-term user trials which favored the oval design over the round one conform to (Wang *et al.*, 2007), mostly due to it being easier to put it on for subjects with bigger hands. Although the weight and size of the bracelet sensor were found to be acceptable by these user trials during initial development, the utilized straps were found to be hard to adjust and especially uncomfortable to wear during the night in the long-term study. For this reason, several designs are planned that will be evaluated over day-long studies that wrap the current prototype in resin casings with more suitable wrist-bands built in.

For such application scenarios, considering detections of used objects will substantially aid our dense motif approach (cf. Chapter 6). One of the possibilities is to use object detections in a similar way as user diaries, thus significantly reducing the search space for recurring motion patterns.

7.7 Conclusions

Inertial sensor data does not always provide enough information to detect complex events such as human activities.

This chapter presented a lightweight bracelet-like sensor that continuously detects *which* objects are handled, and *how*. It does this by combining a small-scale RFID reader and a 3D accelerometer: tags on the objects reveal what they are, while motion patterns performed while holding the objects characterize the type of interaction. Several prototypes of these bracelet-like sensor devices have been proposed previously, but we focus in this work especially on the aspects of making it deployable in real-world environments for longer time frames, balancing between sensor data richness and power consumption of the wearable sensor.

Apart from detailing the crucial hardware design choices to achieve both good detections and a long battery lifespan, we also suggest the use of a small practical benchmark study, the “box test”, to test various parameters used in the wrist-worn RFID antenna circuit. This benchmark allows to test various antenna designs in a limited location in relative short amount of time, with a wide range of objects and test subjects. We used this benchmark to evaluate different antenna shapes, various matching circuit parameters (via the Q value), and a range of RFID sampling frequencies. A technically optimal configuration was found to be an oval-shaped antenna, with a Q-value of 30, searching for nearby tags once every second.

Two data sets were recorded to test this prototype ‘in the field’. A the short-term gardening scenario focused on the use of lots of objects and various interactions per object. The making coffee demo of few minutes duration revealed that RFID tag detections yield a huge information gain in situations where very little characteristic acceleration patterns exist. Finally, a data set was recorded over several days to validate the long-term functioning of our light-weight bracelet in practice, identifying remaining challenges in the bracelet’s strap design but showing that continuous operation for longer periods is feasible.

With regard to fusing acceleration data with RFID tag detections, relying on the linear segmenting algorithm from chapter 4, much faster query-based pattern matching in acceleration data is possible. Furthermore, considering RFID tags as evidence for activities, our dense motifs approach from chapter 6 would be able to automatically use these detections to search ‘nearby’ for activity-related motion patterns, without user annotations.

The RFID sensing capability to detect object interactions as an additional modality to acceleration data reveals its particular advantage when considering a class of activities that are hard to predict solely from acceleration data. Such activities, for example household activities, often tend to be relatively short and, due to only few characteristic acceleration patterns within huge amounts of data obtained in long-term deployments, not captured well by our dense motifs approach. Detecting object interactions through RFID (as a complimentary modality to acceleration data) on the hardware level to a large extent will improve or ideally solve activity recognition for presented application scenarios.

These claims at the moment are left to be evaluated thoroughly in future work.



8 Conclusion

The deployment of sensors is still a challenging undertaking especially in human activity recognition or wireless sensor network scenarios. This mostly because these applications aim at long-term deployments. In contrast to lab experiments or short-term deployments, such deployments require more elaborate approaches, which must cover various aspects of the overall system, choosing or manufacturing sensors to capture physical phenomena of interest, identifying suitable features and utilizing efficient algorithms for dealing with the data, and finally performing classification of activities or events.

In this thesis, we specifically focused on the challenges of long-term scenarios with inertial sensors and addressed several aspects of the target system. The selection of appropriate hardware is particularly important considering the target applications' requirements and constraints. To this end we presented a miniature and power-efficient hardware device that can be worn to capture human motion or deployed in the environment to capture vibrations. Then we argued for using the shape of the acceleration signal as a feature and proposed for these scenarios different feature extraction, pattern matching and classification techniques. We conducted studies of significant size and have discussed the performance of our algorithms and techniques with regard to the targeted applications.

In the following we summarize the contributions of this thesis and conclude with an outlook for further research on potentially promising directions that emerged during our investigations.

8.1 Summary of Contributions

In Sections 1.1 - 1.4 we listed the main challenges of long-term application with wearable sensors for human activity recognition purposes in the medical domain, or the long-term deployments of vibration sensors in a wireless sensor network monitoring scenarios. With the combination of the high frequency sampling on the one hand, and the heavily constrained hardware with regard to the power supply and processing capabilities on the other, the challenge lies in creating a system that covers the whole chain, from a) sensor hardware to obtain the relevant sensor data, over b) efficient data abstraction algorithms or features for dimensionality reduction of the huge amounts of data, to c) efficient classification techniques for the two proposed application domains.

Custom efficient data logger

In Chapter 3 we presented a custom data logger, tailored for long-term deployment scenarios, specifically paying attention to power efficient design of the hardware as well as the logging routine. With this sensor device at hand we were able to perform the experiments and extended studies that constitute this thesis.

We have chosen an open-source design of the sensor hardware as well as the firmware, and made it publicly available from the projects website and code repository, thus making it more easier for other researchers to use our platform for own experiments, as well as to adjust it to personal needs without spending significant resources on “reinventing the wheel”.

Piecewise linear segments as shape features for query-by-example pattern matching

Considering the amount of sensor data acquired, the next step was the utilization of a suitable data representation that would allow faster pattern matching, event classification, and finally activity recognition. We argued for utilizing the “shape” of the signal as a feature, specifically aiming at recurring motion patterns that have characteristic signal shapes.

Chapter 4 presented a piecewise linear approximation algorithm tailored for human motion data, aiming at optimal representation while reducing computational complexity and thus speeding up the computation. The resulting approximation can also be easily visualized as a time series and used for faster query-based pattern matching.

On-line shape encoding for event classification in WSN applications

In Chapter 5 we proposed a WSN monitoring application that is targeting the detection of vibrations caused by passing trains to the rail, and suggested to use inertial sensors to capture and analyze these vibrations for railway infrastructure monitoring purposes. This scenario and the type of data are essentially different from our human activity recognition, with vibration events occurring sporadically over the course of time, demanding event detection and classification directly on the sensor node.

Encoding signal shape with symbols for dense motifs activity recognition

While Chapter 4 focused on using the abstraction time series for query-based pattern matching, in Chapter 6, motivated by a medical bi-polar monitoring scenario, we considered a common activity recognition domain challenge, namely the identification of unknown characteristic patterns that can be used to train a classifier to detect specific activities.

The huge amount of continuous motion data requires efficient and computationally fast techniques to identify characteristic recurring patterns. To this end, we proposed our dense motifs approach that consists of a multi-step procedure: First, the raw sensor data is abstracted to its piecewise linear representation, which is then consecutively converted to a symbolic representation. Utilizing the search-optimized suffix tree data structure, we are able to extract recurring string patterns that then are used in a bag-of-words classifier. The classifier itself predicts target activities based on the salience of motif occurrences.

Detecting human-object interaction and usage with accelerometers and RFID

Based on the early data abstraction and classification approaches, in Chapter 7 we proposed to extend the design by adding an additional modality that would allow to identify human-object interactions. We presented an extension of our data logger that consisted of a miniature RFID reader and a custom bracelet-like antenna, which allows to detect objects equipped with RFID tags. Thus, we obtain both human motion from

the wrist as acceleration data, as well as human-object interactions as RFID tag detections, which allows us to detect which objects the person was interacting with, and how exactly.

Summary

We have presented a system that covers all important aspects, ranging from the hardware to obtain the data, different scenario-oriented data abstraction and feature extractions techniques, as well as evaluations of the pattern matching and classification approaches. The hardware prototype allows for long-term applications in activity recognition and wireless sensor networks deployments. Efficient early data abstraction and feature extraction facilitate faster data analysis, both efficient off-line human motion pattern matching as well as detection of specific activities, or the on-line classification of vibration events for railway monitoring.

The hardware design, the firmware source code, data abstraction and feature extraction algorithms are made publicly available to support reproduction of experiments as well as to encourage other researchers to open their data sets to the community.

8.2 Conclusions

The presented contributions were evaluated in multiple extensive studies, showing the capabilities and advantages of our approach.

Customizing the data logger has a considerable impact on efficiency

Our experiments confirmed that storing the sensor data to the local memory is the most expensive task, besides the operation of the OLED display. With regard to the inexpensive consumer flash cards, these need to be chosen carefully based on their actual current drain footprint. The peak consumption from technical data sheets reveals nothing about the actual consumption of the card.

The current drain for the write operation range from as little as 0.05 mAs for the much more expensive industry class flash memory card, and from 0.15 up to 0.28 mAs for consumer flash cards, depending on their size and manufacturer. For long-term application scenario, the industry flash memory is definitely worth consideration, since it allows much longer runtime on a single battery charge.

The efficient compression of raw sensor data with run-length encoding has been identified as an essential key to achieve run-times of more than two weeks on a single battery charge (using best performing consumer microSD flash card). While for actigraphy-oriented scenarios reducing the sampling rate to 50 or 25 Hz is possible, high fidelity data is required when considering the identification of subtle motions in activities, such as weight lifting at the gym, dancing, or playing music instruments, or when using the sensor to capture vibrations.

For the desired long-term deployments of the wearable sensor, the runtime on a single battery charge was confirmed to be more than 14 days.

Linear segments lend themselves well for query-by-example pattern matching

The comparative studies revealed that our mSWAB is almost two times faster than the original SWAB algorithm, at the cost of more segments and only slightly increased approximation error.

The quality of approximations, varied by setting the approximation threshold, was evaluated through query-by-example pattern matching. Hereby, we compared the widely used dynamic time warping technique, and introduced a competitive novel approach that computes the similarity of patterns based on their K most characteristic segments. On a challenging data set, both techniques reached for optimal parameter settings a detection accuracy of more than 85%, whereby K longest segments was more than twice as fast than DTW.

Furthermore, mSWAB was found computationally efficient enough to be implemented directly on a sensor node even with harsh hardware constraints. Compared against raw sensor data dissemination or run-length encoding on 6 different types of data, the additional computational effort allowed better data compression during logging on 5 out of 6 data sets. This study confirmed that data abstraction directly on the sensor node can be performed in an on-line fashion, while reducing expensive operations (writing to flash memory or wireless communication) and preserving limited power resources of the sensor device.

On-line shape encoding facilitates on-line event classification in WSN applications

The shape of the vibration footprints caused by the trains are visually characteristic enough and can thus be used for event classification. The two studies focused on identifying suitable shape features that can be efficiently computed directly on the sensor node.

With these features at hand, a generic SVM classifier was trained to discern the events, which resulted in good train type classification of the six classes, reaching 97% in accuracy. Implementing the classifier directly on a sensor node can be done by converting the SVM class planes into a straight-forward decision tree. Counting the amount of axle impacts for train length estimation has reached a total mean-squared error of approximately 4.

The experiments have also revealed that some of the extreme acceleration peaks detected in the signal have been caused by impacts of potentially damaged train wheels. This nicely showcases how a low-cost acceleration sensor can obtain many relevant information for a large-scale infrastructure monitoring system. Deploying a sensor network with communication capabilities allows to obtain further information, such train speeds or their acceleration.

Symbolic representation facilitates efficient activity recognition with dense motifs

We evaluated our dense motifs approach in two studies, whereby the larger study contains more than 3800 hours of inertial data from 33 participants and a total of 22 different activities. The study results showed the efficiency of our approach on data sets with target activities that contain characteristic acceleration patterns in the signal. We compared the performance of dense motifs to that of an SVM classifier trained with statistical and frequency domain features, and discussed its suitability for different classes of activities.

Our study results revealed that dense motifs reach better or equal classification results with 90% equal error rate for 11 and more than 80% for additional 7 of the 33 activities. Our approach was able to outperform the SVM trained with features approaches on 30 of 33 datasets, whereby significant performance advantage was achieved on 12 datasets, and being significantly inferior only on one.

Furthermore, dense motifs allow faster processing of sensor data, with less than one minute for training process, which includes the mSWAB abstraction and symbolic mapping step, as well as motif extraction and training of the bag-of words classifier, which is more than twice and 7 times faster than with mean/variance and FFT based features. Performing the classification with the bag-of words classifier required approximately 50 to 100 seconds (this includes the data abstraction step and accumulating motif occurrences), depending on the amount of motifs accumulated. Also here, dense motifs performed slightly faster than mean/variance features, and more than 3.5 times faster than FFT based features.

Detecting object interaction with RFID can improve activity recognition in hardware

The evaluation study focused specifically on the hardware development of the RFID antenna, since its custom design aimed for accurate object detection. We presented a novel “box test” benchmark to compare and evaluate the quality of such bracelet antennas.

Besides an increased maximum reading range of 14 cm (against 10 cm of the Intel bracelet), the box test revealed that our oval-shaped antenna design performed better than the circular antenna: The hit rate of tag detections improved for almost all devices used in the test, with 70-98% for 6 out of 8 objects and 50% for the screw driver. Only the very small lipstick turned out to be hard to detect, for both types of antenna.

Since the RFID reader consumes substantial amount of energy, reducing the sampling frequency and putting the reader to a power-efficient idle mode is very important for long-term deployments. The optimal sampling frequency of the RFID sensor was found to be 1 Hz. With this frequency, the sensor device is able to last for approximately 35 days on a battery with a capacity of 600 mAh.

Combining acceleration data and object interaction yields huge potential in improving activity recognition, as motivated by the three scenarios: gardening, making coffee and domestic deployment. Specifically, the use of an RFID bracelet is preferable in scenarios where activity recognition solely from acceleration data is hard, since the detected objects tend to provide good evidence for underlying activities.

8.3 Outlook

With the achieved results at hand, the question now is, ‘Where do we go from here?’. Obviously, there are a multitude of possible directions that can be taken.

First and foremost, the modular characteristic of the current human activity recognition system allows for improvements and even complete replacements of these modules, opening various opportunities to evaluate alternative techniques. For example, the first segmentation step (mSWAB) can be replaced by a segmentation technique that combines the run-length encoding algorithm for flat signal periods and mSWAB when motion is detected, which would additionally reduce computational complexity and

thus yield positive impact for embedded execution directly on a sensor node. Alternatively, mSWAB can be completely substituted with another algorithms that produce piecewise linear segmentation differently or a piecewise polynomial representation. After this step, query-based pattern matching can be performed, or the representation may be mapped to symbols for motif extraction purposes. Converting the abstracted time series to its symbolic representation can then be achieved based on the polynomial degree and its constants. Furthermore, for application domains without the need for a shape-preserving representation, the sensor data can be converted to symbols, discarding previous information.

The motif extraction step, required for the creation of the bag-of-words classifier, is specifically efficient due to the linear complexity of the suffix tree creation. On the other hand, our current approach considers all accumulated motifs as equal candidates, if these do not occur in background data. This strict filtering can be relaxed such that it will also consider the frequency of motifs within the class as well as in the background data and based on this decide whether to remove or to keep the motif for the classifier. For example, motifs that occur only few times in the background data can be still preserved and used for the dense motif search.

Also, our dense motif approach currently works with strict substring matching, which can be replaced by approximate string search. This might specifically improve the matching results when considering similar patterns that have been abstracted by a different number of linear segments and therefor differ by a limited number of characters.

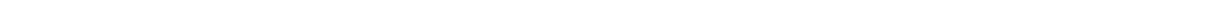
The bag-of-words classifier is currently based on the density of motif occurrences and can be improved through introduction of further features, such as statistical features from the raw data computed from a sliding window, or even other information modalities such as time and date, previous and upcoming activities from a calendar, location, etc. Furthermore, building activity models based on motif occurrences statistics, such as their order or the probability of multiple motifs to appear together or excluding each other, as well as other features can improve performance results. Beyond the scope of the thesis lies the option of using modeling techniques, such as HMMs, that can be used to model such dependencies.

Obviously, the dense motifs approach works specifically well for activities that exhibit characteristic acceleration peaks, and experiences problems when such peaks are not available. For activities where with characteristic cyclic motions, such as during gym exercises, DFT features performed better than dense motifs. For such activities, one possibility to build motifs such that these incorporate frequency domain features. Such considerations however require the design of other abstraction techniques, and are left as future work.

In our WSN railway monitoring scenario, the vibration patterns are currently abstracted based on a fixed width buffer window, which is, given a fixed sampling rate, is not speed invariant. Depending on the speed of the train and the highly variable impact of the wheels, the vibration footprint often varies to such an extent that axle peaks are missed. This leads to high deviations when estimating train lengths. To tackle this problem, a speed invariant approach is required that is able to minimize this estimation error.

Our investigations did not cover the distributed aspect of the sensor network, which can provide relevant information, such as the speed or acceleration estimation of the trains, or the refinement of the train type predictions as well as the wagon counts. Furthermore, the detection of potentially damaged wheels through extreme peak amplitudes should be investigated thoroughly.

The major challenge for this scenario is an actual sensor network deployment for a longer time period in order to evaluate the proposed system under real conditions.



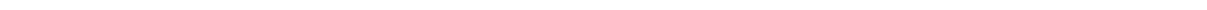
List of Figures

1.1	Capturing physical phenomena with accelerometers producing time series.	3
1.2	Capturing acceleration or vibrations with MEMS accelerometers.	7
1.3	Shape-preserving approximation with piecewise linear segments.	8
1.4	Encoding shape of the signal with efficient features.	9
1.5	Shape-preserving abstraction through symbolic representation.	10
1.6	Object interaction detection through RFID.	10
2.1	Time series representation tree.	15
2.2	Bit-level representation.	16
2.3	Segmented mean and variance representation.	17
2.4	Discrete Fourier Transform.	18
2.5	Piecewise Linear Approximation.	20
2.6	Polynomial approximation of sensor data.	21
2.7	Symbolic aggregate approximation	22
2.8	Symbolic abstraction of 3D time series.	23
2.9	An actogram as used in psychiatric monitoring.	27
2.10	Motivation for event classification in WSN.	30
3.1	Capturing human motion with MEMS accelerometers.	33
3.2	The custom data logger for long-term deployments.	34
3.3	Schematics to measure current consumption.	36
3.4	BioRob robot was used to perform a repetitive motion.	37
3.5	Determining battery capacity via drain resistor.	38
3.6	Current consumption: PIC polling vs. FIFO buffer.	40
3.7	Sampling and writing data to flash memory.	41
3.8	Current consumption: comparing different microSD cards.	42
3.9	Current consumption: the impact of the OLED display.	43
4.1	Linear segments as shape-preserving features.	47
4.2	Query-by-example prototype.	49
4.3	Example for long-term activity data from a wrist-worn sensor.	50
4.4	Modification of the original SWAB algorithm.	51
4.5	Example of multi-dimensional approximation with mSWAB.	53
4.6	Approximation of 3D acceleration time series with different methods. . . .	55
4.7	SWAB vs. mSWAB approximation error and number of segments.	56
4.8	Comparison of SWAB and mSWAB regarding error and execution time. . .	57
4.9	Using Dynamic Time Warping for pattern matching.	58
4.10	Using K longest segments for pattern matching.	59
4.11	Photos from the Walk8 data set.	60
4.12	Walk8 data set classification results.	61

4.13	Basic functionality of embedded mSWAB on a sensor node.	63
4.14	Example for the computation of the approximation cost.	63
4.15	Data sets used for the evaluation of the embedded mSWAB.	65
4.16	Target platform: a TelosB wireless sensor node.	66
4.17	Power consumption estimation with Energest: RLE vs emSWAB.	68
4.18	Comparison of energest results for six data sets.	69
5.1	Encoding the shape of vibrations through features.	73
5.2	Our system's concept: A sensor network deployed along railway tracks. . .	75
5.3	Sensor nodes at railway tracks capture the vibrations caused by trains. . .	76
5.4	Photos of trains recorded and classified during the first evaluation.	79
5.5	Part of the data set, showing approximately 35 hours of sensor data. . . .	80
5.6	Snapshots from video footage used for post-deployment annotations. . . .	81
5.7	Inspection of examples of train events detected in the sensor data.	83
5.8	Summary of the kNN and SVM classification results as confusion matrices. .	85
5.9	Confusion matrices for the SVM classification results.	87
5.10	Estimating length of the trains through extracted variance peaks.	88
5.11	Train speed estimation through time delay between sensor nodes.	90
5.12	Detection of defect wheels that require maintenance.	91
6.1	Abstracting the acceleration time series to its symbolic representation. . .	93
6.2	Motivation for dense motif discovery for long-term deployments.	94
6.3	Method overview explaining the dense motif approach.	96
6.4	Second approximation step: mapping to symbolic representation.	98
6.5	Extraction of motifs by means of a Suffix Tree.	99
6.6	Bag-of-Words classifier from extracted motifs.	100
6.7	Extended step-by-step overview.	102
6.8	Extended runtime comparison on 24 hours of data.	103
6.9	Dense motifs score compared to filtered SVM classification.	104
6.10	Precision and recall performance on 19 activities.	105
6.11	Precision and recall performance on remaining 14 activities.	106
6.12	Dense motifs performance on badminton data.	107
6.13	Impact of the mSWAB parameters on the classification performance. . . .	108
6.14	Example of dense motifs performance for flamenco data.	109
6.15	Example of dense motifs performance for the gym data.	110
7.1	Detecting object interaction through RFID.	113
7.2	Motivation for a bracelet with RFID and inertial sensors.	114
7.3	The bracelet antenna, the RFID reader and the inertial sensor.	116
7.4	SkyeTek M1-mini RFID reader and required matching circuit.	117
7.5	Benchmarking the RFID antenna with the "box test".	119
7.6	Comparing the hit rates of different RFID antenna designs.	120
7.7	Impact of the Q value on the detection of different objects.	121
7.8	Raw data from both the RFID and accelerometer sensors.	122
7.9	mSWAB performance on the gardening data set.	123
7.10	Long-term domestic study with the new bracelet prototype.	124

List of Tables

3.1	Test cases to evaluate the data logger hardware and software design. . . .	39
3.2	Electric charge consumption comparison for different operation modes. . .	44
4.1	Run-times of different approximation methods on 3D acceleration data. .	54
4.2	Comparing accuracy and execution time of DTW and KLS	60
5.1	List of features considered for train type classification.	78
5.2	Different train types from the second experiment.	81
5.3	Mean-squared error of the estimated train lengths.	89
6.1	List of participants from the extended study.	101
7.1	Several projects involving glove- and bracelet-based RFID readers.	121



Bibliography

- E. Aboelela, W. Edberg, C. Papakonstantinou, and V. Vokkarane (2006). Wireless sensor network based model for secure railway operations, in *Proc. of the 25th IEEE International Performance, Computing, and Communications Conference (IPCCC)*. 2006.
- R. Agrawal, C. Faloutsos, and A. Swami (1993). Efficient similarity search in sequence databases, in D. B. Lomet (ed.), *Foundations of Data Organization and Algorithms 1993*, no. 730 in Lecture Notes in Computer Science, pp. 69–84, Springer Berlin Heidelberg.
- O. Amft, H. Junker, and G. Tröster (2005). Detection of Eating and Drinking Arm Gestures Using Inertial Body-Worn Sensors, in *Proc. of the Ninth IEEE International Symposium on Wearable Computers (ISWC)*. 2005.
- O. Amft and G. Tröster (2008). Recognition of dietary activity events using on-body sensors, *Artificial Intelligence in Medicine*, vol. 42(2), pp. 121–136.
- L. Angrisani, D. Grillo, R. Moriello, and G. Filo (2010). Automatic detection of train arrival through an accelerometer, in *Proc. of the 2010 IEEE Instrumentation and Measurement Technology Conference (I2MTC) 2010*.
- R. A. Baeza-Yates and G. H. Gonnet (1996). Fast Text Searching for Regular Expressions or Automaton Searching on Tries, *J. ACM*, vol. 43(6), pp. 915–936.
- A. Bagnall, C. A. Ratanamahatana, E. Keogh, S. Lonardi, and G. Janacek (2006). A Bit Level Representation for Time Series Data Mining with Shape Based Similarity, *Data Mining and Knowledge Discovery*, vol. 13(1), pp. 11–40.
- L. Bao and S. S. Intille (2004). Activity Recognition from User-Annotated Acceleration Data, in A. Ferscha and F. Mattern (eds.), *Pervasive Computing 2004*, no. 3001 in Lecture Notes in Computer Science, pp. 1–17, Springer Berlin Heidelberg.
- K. Barr and K. Asanović (2003). Energy aware lossless data compression, in *Proc. of the 1st International Conference on Mobile Systems, Applications and Services (MobiSys)*. 2003.
- M. Barsky, U. Stege, A. Thomo, and C. Upton (2009). Suffix Trees for Very Large Genomic Sequences, in *Proc. of the 18th ACM Conference on Information and Knowledge Management 2009*.
- R. Bellman (1957). *Dynamic Programming*, Princeton University Press, Princeton, NJ, USA, 1st edn.
- E. Berlin, J. Liu, K. van Laerhoven, and B. Schiele (2010). Coming To Grips With The Objects We Grasp: Detecting Interactions With Efficient Wrist-Worn Sensors, in *Proc. of the Fourth International Conference on Tangible, Embedded, and Ambodied Interaction (TEI)*. 2010.

-
- E. Berlin and K. Van Laerhoven (2010). An on-line piecewise linear approximation technique for wireless sensor networks, in *Proc. of the 2010 IEEE 35th Conference on Local Computer Networks (LCN)*. 2010.
- E. Berlin and K. Van Laerhoven (2012a). Detecting Leisure Activities with Dense Motif Discovery, in *Proc. of the 2012 ACM Conference on Ubiquitous Computing (UbiComp)*. 2012.
- E. Berlin and K. Van Laerhoven (2012b). Trainspotting: Combining fast features to enable detection on resource-constrained sensing devices, in *Proc. of the 2012 Ninth International Conference on Networked Sensing Systems (INSS)*. 2012.
- E. Berlin and K. Van Laerhoven (2013). Sensor Networks for Railway Monitoring: Detecting Trains from their Distributed Vibration Footprints, in *Proc. of the 2013 IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS)*. 2013.
- U. Blanke, B. Schiele, M. Kreil, P. Lukowicz, B. Sick, and T. Gruber (2010). All for one or one for all? Combining heterogeneous features for activity spotting, in *Proc. of the 2010 8th IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*. 2010.
- H. R. Bogen, M. Herbst, J. A. Huisman, U. Rosenbaum, A. Weuthen, and H. Vereecken (2010). Potential of Wireless Sensor Networks for Measuring Soil Water Content Variability, *Vadose Zone Journal*, vol. 9(4), pp. 1002–1013.
- M. Borazio, E. Berlin, N. Kücüküydiz, P. M. Scholl, and K. Van Laerhoven (2014). Towards Benchmarked Sleep Detection with Inertial Wrist-worn Sensing Units, in *Proc. of the 2014 IEEE International Conference on Healthcare Informatics (ICHI 2014)*. 2014.
- M. Boronowsky, T. Nicolai, C. Schlieder, and A. Schmidt (2001). Winspect: a case study for wearable computing-supported inspection tasks, in *Fifth International Symposium on Wearable Computers (ISWC)*. 2001.
- A. Braun, H. Heggen, and R. Wichert (2012). CapFloor – A Flexible Capacitive Indoor Localization System, in S. Chessa and S. Knauth (eds.), *Evaluating AAL Systems Through Competitive Benchmarking. Indoor Localization and Tracking 2012*, no. 309 in Communications in Computer and Information Science, pp. 26–35, Springer Berlin Heidelberg.
- C. Burton, B. McKinstry, A. Szentagotai Tătar, A. Serrano-Blanco, C. Pagliari, and M. Wolters (2013). Activity monitoring in patients with depression: A systematic review, *Journal of Affective Disorders*, vol. 145(1), pp. 21–28.
- S. H. Cameron (1966). Piece-wise linear approximations, Technical report CSTN-106, IIT Research Institute.
- E. Capo-Chichi, H. Guyennet, and J.-M. Friedt (2009). K-RLE: A New Data Compression Algorithm for Wireless Sensor Network, in *Proc. of the Third International Conference on Sensor Technologies and Applications*. 2009.

-
- K. Chakrabarti, E. Keogh, S. Mehrotra, and M. Pazzani (2002). Locally Adaptive Dimensionality Reduction for Indexing Large Time Series Databases, *ACM Trans. Database Syst.*, vol. 27(2), p. 188–228.
- J. Cheng, O. Amft, G. Bahle, and P. Lukowicz (2013). Designing Sensitive Wearable Capacitive Sensors for Activity Recognition, *IEEE Sensors Journal*, vol. 13(10), pp. 3935–3947.
- T. Choudhury, S. Consolvo, B. Harrison, J. Hightower, A. LaMarca, L. Legrand, A. Rahimi, A. Rea, G. Bordello, B. Hemingway, P. Klasnja, K. Koscher, J. Landay, J. Lester, D. Wyatt, and D. Haehnel (2008). The Mobile Sensing Platform: An Embedded Activity Recognition System, *IEEE Pervasive Computing*, vol. 7(2), pp. 32–41.
- P. Corkum, R. Tannock, H. Moldofsky, S. Hogg-Johnson, and T. Humphries (2001). Actigraphy and parental ratings of sleep in children with attention-deficit/hyperactivity disorder (ADHD), *Sleep*, vol. 24(3), pp. 303–312.
- D.-J. Dijk, D. F. Neri, J. K. Wyatt, J. M. Ronda, E. Riel, A. R.-D. Cecco, R. J. Hughes, A. R. Elliott, G. K. Prisk, J. B. West, and C. A. Czeisler (2001). Sleep, performance, circadian rhythms, and light-dark cycles during two space shuttle flights, *American Journal of Physiology - Regulatory, Integrative and Comparative Physiology*, vol. 281(5), pp. R1647–R1664.
- P. Donato, J. Urena, M. Mazo, J. Garcia, and F. Alvarez (2004). Electromagnetic sensor array for train wheel detection, in *Proc. of the Sensor Array and Multichannel Signal Processing Workshop Proceedings, 2004* 2004.
- P. Duhamel and M. Vetterli (1990). Fast fourier transforms: A tutorial review and a state of the art, *Signal Processing*, vol. 19(4), pp. 259–299.
- A. Dunkels, B. Gronvall, and T. Voigt (2004). Contiki - a lightweight and flexible operating system for tiny networked sensors, in *Proc. of the 29th Annual IEEE International Conference on Local Computer Networks, 2004* 2004.
- A. Dunkels, F. Osterlind, N. Tsiftes, and Z. He (2007). Software-based on-line energy estimation for sensor nodes, in *Proc. of the 4th Workshop on Embedded Networked Sensors (EmNets). 2007*.
- P. Dutta, M. Grimmer, A. Arora, S. Bibyk, and D. Culler (2005). Design of a wireless sensor network platform for detecting rare, random, and ephemeral events, in *Proc. of the Fourth International Symposium on Information Processing in Sensor Networks (IPSN). 2005*.
- C. Faloutsos, H. V. Jagadish, A. O. Mendelzon, and T. Milo (1997). A Signature Technique for Similarity-Based Queries (Extended Abstract), in *Proceedings of Compression and Complexity of Sequences (SEQUENCES97) 1997*.
- A. Feldman (2004). *ReachMedia: On-the-move Interaction with Everyday Objects*, Ph.D. thesis, Master of Science in Media Arts and Sciences at the Massachusetts Institute of Technology.

-
- A. Feldman, E. Tapia, S. Sadi, P. Maes, and C. Schmandt (2005). ReachMedia: On-the-move interaction with everyday objects, in *Proc. of the Ninth IEEE International Symposium on Wearable Computers (ISWC)*. 2005.
- K. Fishkin, M. Philipose, and A. Rea (2005). Hands-On RFID: Wireless Wearables for Detecting Use of Objects, in *Proc. of the Ninth IEEE International Symposium on Wearable Computers (ISWC)*. 2005.
- F. Foerster, M. Smeja, and J. Fahrenberg (1999). Detection of posture and motion by accelerometry: a validation study in ambulatory monitoring, *Computers in Human Behavior*, vol. 15(5), pp. 571–583.
- E. Fuchs, T. Gruber, J. Nitschke, and B. Sick (2009). On-line motif detection in time series with SwiftMotif, *Pattern Recognition*, vol. 42(11), pp. 3015–3031.
- E. Fuchs, T. Gruber, J. Nitschke, and B. Sick (2010). Online Segmentation of Time Series Based on Polynomial Least-Squares Approximations, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32(12), pp. 2232–2245.
- T. Gao, T. Massey, L. Selavo, D. Crawford, B.-r. Chen, K. Lorincz, V. Shnayder, L. Hauenstein, F. Dabiri, J. Jeng, A. Chanmugam, D. White, M. Sarrafzadeh, and M. Welsh (2007). The Advanced Health and Disaster Aid Network: A Light-Weight Wireless Medical System for Triage, *IEEE Transactions on Biomedical Circuits and Systems*, vol. 1(3), pp. 203–216.
- T. Gao, C. Pesto, L. Selavo, Y. Chen, J. Ko, J. H. Lim, A. Terzis, A. Watt, J. Jeng, B.-r. Chen, K. Lorincz, and M. Welsh (2008). Wireless Medical Sensor Networks in Emergency Response: Implementation and Pilot Results, in *Proc. of the 2008 IEEE Conference on Technologies for Homeland Security 2008*.
- H. P. Godfrey and R. G. Knight (1984). The validity of actometer and speech activity measures in the assessment of depressed patients., *The British Journal of Psychiatry*, vol. 145(2), pp. 159–163.
- S. Golomb (1966). Run-length encodings (Corresp.), *IEEE Transactions on Information Theory*, vol. 12(3), pp. 399–401.
- S. Gouravajhala, D. Wang, L. Khuon, and F. Bao (2012). EpSMART: Epileptic seizure monitoring with alerts in real time: A tablet-based Android application for a real-time multi-modal seizure detection system, in *Proc. of the 2012 IEEE International Conference on Bioinformatics and Biomedicine Workshops (BIBMW)*. 2012.
- T. Grosse-Puppenthal, E. Berlin, and M. Borazio (2012). Enhancing Accelerometer-Based Activity Recognition with Capacitive Proximity Sensing, in *Proc. of the European Conference on Ambient Intelligence (AmI)*. 2012.
- S. Gupte, O. Masoud, R. Martin, and N. Papanikolopoulos (2002). Detection and classification of vehicles, *IEEE Transactions on Intelligent Transportation Systems*, vol. 3(1), pp. 37–47.
-

-
- R. Hamid, S. Maddi, A. Bobick, and I. Essa (2007). Structure from Statistics - Unsupervised Activity Analysis using Suffix Trees, in *Proc. of the IEEE 11th International Conference on Computer Vision (ICCV)*. 2007.
- M. R. Hodges and M. E. Pollack (2007). An 'Object-Use Fingerprint': The Use of Electronic Sensors for Human Identification, in *Proc. of the 9th International Conference on Ubiquitous Computing (UbiComp)*. 2007.
- J. Huang, H. Ogai, C. Shao, J. Zheng, I. Maruyama, S. Nagata, and H. Inujima (2010). On vibration signal analysis in Bridge Health Monitoring System by using Independent Component Analysis, in *Proc. of the SICE Annual Conference 2010* 2010.
- N. Q. V. Hung and D. T. Anh (2007). Combining SAX and Piecewise Linear Approximation to Improve Similarity Search on Financial Time Series, in *Proc. of the International Symposium on Information Technology Convergence (ISITC)*. 2007.
- D. T. G. Huynh (2008). *Human Activity Recognition with Wearable Sensors*, Ph.D. thesis, TU Darmstadt.
- T. Huynh, U. Blanke, and B. Schiele (2007). Scalable Recognition of Daily Activities with Wearable Sensors, in *Location- and Context-Awareness 2007*.
- T. Huynh, M. Fritz, and B. Schiele (2008). Discovery of activity patterns using topic models, in *10th International Conference on Ubiquitous Computing (UbiComp)*. 2008.
- T. Huynh and B. Schiele (2005). Analyzing features for activity recognition, in *Proc. of the 2005 Joint Conference on Smart Objects and Ambient Intelligence: Innovative Context-aware Services: Usages and Technologies (sOc-EUSAI)*. 2005.
- S. S. Intille, K. Larson, E. M. Tapia, J. S. Beaudin, P. Kaushik, J. Nawyn, and R. Rockinson (2006). Using a Live-In Laboratory for Ubiquitous Computing Research, in *Proc. of the 4th International Conference on Pervasive Computing (PERVASIVE)*. 2006.
- F. Itakura (1975). Minimum prediction residual principle applied to speech recognition, *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 23(1), pp. 67–72.
- P. Jallon, S. Bonnet, M. Antonakios, and R. Guillemaud (2009). Detection system of motor epileptic seizures through motion analysis with 3D accelerometers, in *Proc. of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. 2009.
- L. Jankelowitz, K. J. Reid, L. Wolfe, J. Cullina, P. C. Zee, and M. Jain (2005). CYstic fibrosis patients have poor sleep quality despite normal sleep latency and efficiency*, *Chest*, vol. 127(5), pp. 1593–1599.
- H. Junker (2005). *Human Activity and Gesture Spotting with Body-Worn Sensors*, Ph.D. thesis, ETH Zürich.
- S. Keawkamnerd, J. Chinrungrueng, and C. Jaruchart (2008). Vehicle Classification with low computation magnetic sensor, in *Proc. of the 8th International Conference on ITS Telecommunications (ITST)*. 2008.

-
- E. Keogh, K. Chakrabarti, M. Pazzani, and S. Mehrotra (2001a). Dimensionality Reduction for Fast Similarity Search in Large Time Series Databases, *Knowledge and Information Systems*, vol. 3(3), pp. 263–286.
- E. Keogh, S. Chu, D. Hart, and M. Pazzani (2001b). An online algorithm for segmenting time series, in *Proc. of the IEEE International Conference on Data Mining (ICDM)*. 2001.
- E. Keogh, Q. Zhu, B. Hu, Y. Hao, X. Xi, L. Wei, and C. A. Ratanamahatana (2011). The UCR Time Series Classification/Clustering Homepage: www.cs.ucr.edu/~eamonn/time_series_data/.
- D. K. Kim, J. S. Sim, H. Park, and K. Park (2005). Constructing suffix arrays in linear time, *Journal of Discrete Algorithms*, vol. 3(2–4), pp. 126–142.
- S. Kim, S. Pakzad, D. Culler, J. Demmel, G. Fenves, S. Glaser, and M. Turon (2007). Health Monitoring of Civil Infrastructures Using Wireless Sensor Networks, in *Proc. of the 6th Int’l Symposium on Information Processing in Sensor Networks (IPSN)*. 2007.
- M. H. Ko, G. West, S. Venkatesh, and M. Kumar (2005). Online Context Recognition in Multisensor Systems using Dynamic Time Warping, in *Proc. of the 2005 International Conference on Intelligent Sensors, Sensor Networks and Information Processing*. 2005.
- Y.-A. Le Borgne, S. Santini, and G. Bontempi (2007). Adaptive model selection for time series prediction in wireless sensor networks, *Signal Processing*, vol. 87(12), pp. 3010–3020.
- T. R. Leffingwell, N. J. Cooney, J. G. Murphy, S. Luczak, G. Rosen, D. M. Dougherty, and N. P. Barnett (2013). Continuous objective monitoring of alcohol use: twenty-first century measurement using transdermal sensors, *Alcoholism, Clinical and Experimental Research*, vol. 37(1), pp. 16–22.
- J. Lei, X. Ren, and D. Fox (2012). Fine-grained kitchen activity recognition using RGB-D, in *Proc. of the 2012 ACM Conference on Ubiquitous Computing (UbiComp)*. 2012.
- D. Lemire (2009). Faster retrieval with a two-pass dynamic-time-warping lower bound, *Pattern Recognition*, vol. 42(9), pp. 2169–2180.
- J. Lin, E. Keogh, S. Lonardi, and B. Chiu (2003). A symbolic representation of time series, with implications for streaming algorithms, in *Proc. of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery (DMKD)*. 2003.
- J. Lin, E. Keogh, L. Wei, and S. Lonardi (2007). Experiencing SAX: a Novel Symbolic Representation of Time Series, *Data Mining and Knowledge Discovery*, vol. 15(2), pp. 107–144.
- B. Logan, J. Healey, M. Philipose, E. M. Tapia, and S. Intille (2007). A Long-Term Evaluation of Sensing Modalities for Activity Recognition, in *UbiComp 2007: Ubiquitous Computing 2007*.
- K. Lorincz, B.-r. Chen, G. W. Challen, A. R. Chowdhury, S. Patel, P. Bonato, and M. Welsh (2009). Mercury: a wearable sensor network platform for high-fidelity motion analysis, in *Proc. of the 7th ACM Conference on Embedded Networked Sensor Systems (SenSys)*. 2009.

-
- G. P. Mazarakis and J. N. Avaritsiotis (2007). Vehicle classification in Sensor Networks using time-domain signal processing and Neural Networks, *Microprocess. Microsyst.*, vol. 31(6), pp. 381–392.
- Y. Medynskiy, S. Gov, A. Mazalek, and D. Minnen (2007). Wearable RFID for Play, in *Tangible Play Workshop, 2007 Conference on Intelligent User Interfaces 2007*.
- D. Minnen, T. Starner, I. Essa, and C. Isbell (2006). Discovering Characteristic Actions from On-Body Sensor Data, in *Proc. of the 2006 10th IEEE International Symposium on Wearable Computers (ISWC). 2006*.
- Y. Morinaka, M. Yoshikawa, T. Amagasa, and S. Uemura (2001). The L-index: An indexing structure for efficient subsequence matching in time sequence databases, in *Proc. of the 5th PacificAisa Conference on Knowledge Discovery and Data Mining. 2001*.
- C. Narayanaswami, T. Inoue, T. Cipolla, J. Sanford, E. Schlig, and Vishal (2002). *IBM's Linux Watch: The Challenge of Miniaturization*.
- G. Ogris, T. Stiefmeier, P. Lukowicz, and G. Tröster (2008). Using a Complex Multi-Modal On-Body Sensor System for Activity Spotting, in *Proc. of the 12th IEEE International Symposium on Wearable Computers (ISWC). 2008*.
- F. Osterlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt (2006). Cross-Level Sensor Network Simulation with COOJA, in *Proc. of the 2006 31st IEEE Conference on Local Computer Networks 2006*.
- J. Palmer (2012). The need for train detection, in *IET Professional Development Course on Railway Signalling and Control Systems (RSCS 2012). 2012*.
- S. Patel, K. Lorincz, R. Hughes, N. Huggins, J. Growdon, D. Standaert, M. Akay, J. Dy, M. Welsh, and P. Bonato (2009). Monitoring Motor Fluctuations in Patients With Parkinson's Disease Using Wearable Sensors, *IEEE Transactions on Information Technology in Biomedicine*, vol. 13(6), pp. 864–873.
- D. Patterson, D. Fox, H. Kautz, and M. Philipose (2005). Fine-Grained Activity Recognition by Aggregating Abstract Object Usage, in *Proc. of the Ninth IEEE International Symposium on Wearable Computers (ISWC). 2005*.
- C. Pham, T. Plötz, and P. Olivier (2010). A Dynamic Time Warping Approach to Real-Time Activity Recognition for Food Preparation, in B. Ruyter, R. Wichert, D. Keyson, P. Markopoulos, N. Streitz, M. Divitini, N. Georgantas, and A. Mana Gomez (eds.), *Ambient Intelligence 2010*, vol. 6439 of *Lecture Notes in Computer Science*, pp. 21–30, Springer Berlin Heidelberg.
- M. Philipose, K. Fishkin, M. Perkowitz, D. Patterson, D. Fox, H. Kautz, and D. Hahnel (2004). Inferring Activities from Interactions with Objects, *IEEE Pervasive Computing*, vol. 3(4), pp. 50–57.
- G. Phillips (1968). Algorithms for piecewise straight line approximations, *The Computer Journal*, vol. 11(2), pp. 211–212.

-
- N. Rajpoot and K. Masood (2005). Human Gait Recognition with 3-D Wavelets and Kernel based Subspace Projections, in *Proc. of the Workshop on Human Activity Recognition and Modeling (HAREM 2005)* 2005.
- M. Ramesh, S. Kumar, P. Rangan, and A. Vidyapeetham (2009). Wireless Sensor Network for Landslide Detection, in *Proc. of the International Conference on Wireless Networks (ICWN)*. 2009.
- C. Ratanamahatana and E. Keogh (2004). Making Time-series Classification More Accurate Using Learned Constraints, in *Proc. of the SIAM International Conference on Data Mining 2004*.
- A. Reinhardt, D. Christin, M. Hollick, and R. Steinmetz (2009). On the energy efficiency of lossless data compression in wireless sensor networks, in *Proc. of the IEEE 34th Conference on Local Computer Networks (LCN)*. 2009.
- S. Royant-Parola, A. A. Borbely, I. Tobler, O. Benoit, and D. Widlocher (1986). Monitoring of long-term motor activity in depressed patients., *The British Journal of Psychiatry*, vol. 149(3), pp. 288–293.
- H. Sakoe and S. Chiba (1971). A dynamic programming approach to continuous speech recognition, in *Seventh International Congress on Acoustics 1971*.
- A. Schmidt, H.-W. Gellersen, and C. Merz (2000). Enabling implicit human computer interaction: a wearable RFID-tag reader, in *Proc. of the The Fourth International Symposium on Wearable Computers (ISWC)*. 2000.
- P. M. Scholl, N. Kücüküydiz, and K. V. Laerhoven (2013). When Do You Light a Fire?: Capturing Tobacco Use with Situated, Wearable Sensors, in *Proc. of the 2013 ACM Conference on Pervasive and Ubiquitous Computing Adjunct Publication 2013*.
- J. Shieh and E. Keogh (2008). iSAX: Indexing and Mining Terabyte Sized Time Series, in *Proc. of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*. 2008.
- K. O. Soffke (2007). *Modellierung, Simulation und Entwurf induktiv gekoppelter Transpondersysteme*, Ph.D. thesis, TU Darmstadt.
- T. Stiefmeier, D. Roggen, and G. Tröster (2007). Gestures are strings: efficient online gesture spotting and classification using string matching, in *ICST 2nd International Conference on Body Area Networks (BodyNets)*. 2007.
- M. Stikic (2010). *Towards Less Supervision for Scalable Recognition of Daily Activities*, phd, TU Darmstadt.
- M. Stikic, T. Huynh, K. Van Laerhoven, and B. Schiele (2008). ADL recognition based on the combination of RFID and accelerometer sensing, in *Proc. of the 2nd Int'l Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth)*. 2008.
- H. Stone (1961). Approximation of curves by line segments, *Mathematics of Computation*, pp. 40–47.

-
- F.-T. Sun, C. Kuo, and M. Griss (2011). PEAR: Power efficiency through activity recognition (for ECG-based sensing), in *Proc. of the 2011 5th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth)*. 2011.
- M. Tahmasian, H. Khazaie, S. Golshani, and K. Avis (2013). Clinical Application of Actigraphy in Psychotic Disorders: A Systematic Review, *Current Psychiatry Reports*, vol. 15(6).
- E. M. Tapia, S. S. Intille, and K. Larson (2004). Activity Recognition in the Home Using Simple and Ubiquitous Sensors, in A. Ferscha and F. Mattern (eds.), *Pervasive Computing 2004*, no. 3001 in Lecture Notes in Computer Science, pp. 158–175, Springer Berlin Heidelberg.
- M. H. Teicher (1995). Actigraphy and motion analysis: new tools for psychiatry, *Harvard review of psychiatry*, vol. 3(1), pp. 18–35.
- E. Ukkonen (1992). Constructing Suffix Trees On-Line in Linear Time, in *Proc. of the IFIP 12th World Computer Congress on Algorithms, Software, Architecture - Information Processing '92, Volume 1 - Volume I 1992*.
- E. Ukkonen (1995). On-line construction of suffix trees, *Algorithmica*, vol. 14(3), pp. 249–260.
- G. Valenza, M. Nardelli, A. Lanata, C. Gentili, G. Bertschy, R. Paradiso, and E. Scilingo (2013). Wearable Monitoring for Mood Recognition in Bipolar Disorder based on History-Dependent Long-Term Heart Rate Variability Analysis, *IEEE Journal of Biomedical and Health Informatics*, vol. Early Access Online.
- K. Van Laerhoven and E. Berlin (2009). When Else Did This Happen? Efficient Subsequence Representation and Matching for Wearable Activity Data, in *Proc. of the International Symposium on Wearable Computers (ISWC)*. 2009.
- K. Van Laerhoven, E. Berlin, and B. Schiele (2009). Enabling Efficient Time Series Analysis for Wearable Activity Data, in *Proc. of the International Conference on Machine Learning and Applications (ICMLA)*. 2009.
- K. Van Laerhoven, D. Kilian, and B. Schiele (2008). Using Rhythm Awareness in Long-Term Activity Recognition, in *Proc. of the 12th IEEE International Symposium on Wearable Computers (ISWC)*. 2008.
- J. Virkkala (2012). Using Accelerometers as Actigraphs, *Poster at the 21st Congress of the European Sleep Research Society (ESRS 2012)*.
- C. Wang, Q. Xiao, H. Liang, X. Chen, X. Cai, and Y. Liu (2006). On-Line Vibration Source Detection of Running Trains Based on Acceleration Measurement, in *Proc. of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems 2006*.
- S. Wang, W. Pentney, A.-M. Popescu, T. Choudhury, and M. Philipose (2007). Common Sense Based Joint Training of Human Activity Recognizers, in *Proc. of the 20th International Joint Conference on Artificial Intelligence (IJCAI)*. 2007.

-
- R. Want (2006). An introduction to RFID technology, *IEEE Pervasive Computing*, vol. 5(1), pp. 25–33.
- J. Ward, P. Lukowicz, G. Tröster, and T. Starner (2006). Activity Recognition of Assembly Tasks Using Body-Worn Microphones and Accelerometers, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28(10), pp. 1553–1567.
- G. Werner-Allen, J. Johnson, M. Ruiz, J. Lees, and M. Welsh (2005). Monitoring volcanic eruptions with a wireless sensor network, in *Proceedings of the 2nd European Workshop on Wireless Sensor Networks, 2005* 2005.
- G. Werner-Allen, K. Lorincz, M. Ruiz, O. Marcillo, J. Johnson, J. Lees, and M. Welsh (2006). Deploying a wireless sensor network on an active volcano, *IEEE Internet Computing*, vol. 10(2), pp. 18–25.
- F. H. Wilhelm, M. C. Pfaltz, and P. Grossman (2006). Continuous electronic data capture of physiology, behavior and experience in real life: towards ecological momentary assessment of emotion, *Interacting with Computers*, vol. 18(2), pp. 171–186.
- P. E. William and M. Hoffman (2011). Classification of Military Ground Vehicles Using Time Domain Harmonics’ Amplitudes, *IEEE Transactions on Instrumentation and Measurement*, vol. 60(11), pp. 3720–3731.
- B.-K. Yi and C. Faloutsos (2000). Fast Time Sequence Indexing for Arbitrary Lp Norms, in *Proc. of the 26th International Conference on Very Large Data Bases 2000*.
- A. Zinnen, U. Blanke, and B. Schiele (2009). An Analysis of Sensor-Oriented vs. Model-Based Activity Recognition, in *Proc. of the International Symposium on Wearable Computers (ISWC). 2009*.
- A. Zinnen, K. Van Laerhoven, and B. Schiele (2007). Toward Recognition of Short and Non-repetitive Activities from Wearable Sensors, in *Proc. of the European Conference on Ambient Intelligence (AmI) 2007*.

Curriculum Vitæ

Eugen Berlin

Education:	2010–2014	PhD student in computer science <i>TU Darmstadt, Germany</i> Topic: Early Abstraction of Inertial Sensor Data for Long-Term Deployments
	2007–2009	M.Sc. in computer science <i>TU Darmstadt, Germany</i> Minor: IT management
	2004–2007	B.Sc. in computer science <i>TU Darmstadt, Germany</i>

Employment:	2010–2014	Research assistant <i>Embedded Sensing Systems</i> <i>TU Darmstadt, Germany</i>
	2007–2009	Student research assistant <i>Multimodal Interactive Systems</i> <i>TU Darmstadt, Germany</i>
	2005–2007	Student research assistant <i>Fraunhofer IPSI and Fraunhofer SIT institutes</i> <i>Darmstadt, Germany</i>
