Activity Recognition With Instrumented and Wearable Artifacts

Dissertation

zur Erlangung des Grades des Doktors der
Ingenieurwissenschaften
eingereicht bei der Technischen Fakultät der
Albert-Ludwigs-Universität Freiburg
von

Dipl. Inf. Philipp M. Scholl

Erstgutachter: Prof. Dr. Kristof Van Laerhoven
Zweitgutachter: Prof. Dr. Bernd Becker
Prüfungstermin: 27. März 2018

Ich versichere, dass ich die Arbeit selbständig und nur mit den angegebenen Quellen und Hilfsmitteln angefertigt und dass ich alle Stellen, die aus anderen Werken (auch aus dem Internet) dem Wortlaut oder dem Sinne nach entnommen sind, kenntlich gemacht habe.

Ort, Datum

Unterschrift

ACKNOWLEDGEMENTS

First and foremost, I express my sincere gratitude to Professor Kristof Van Laerhoven, my thesis supervisor, for his constant guidance, motivation, novel ideas, new perspectives on this thesis, and for the support without which this thesis would not have been possible. I also thank my committee member Professor Bernd Becker, for supporting my time at the University of Freiburg.

I would also like to thank my colleagues, office mates and friends for many fruitful discussion, and various fun to be had during breaks. Particularly the people at the University of Darmstadt: Vinay Sachidanda, Martin Berchtold, Nagihan Küzücyildiz, Eugen Berlin, Marko Borazio, Martin Dietrich, Agha Muhammad, Sofia Nikitaki, Christian Seeger, Rodrigo Do Carmo, Iliya Gurov, Robert Langenberg, Martina Brachmann, Paul Baumann, Stefan Kohlbrecher, Johannes Meyer, and Tobias Große-Puppendahl. My former colleagues at Karlsruhe: Matthias Budde, Till Riedel, Dawud Gordon, Matthias Berning, Markus Scholz and Andreas Hermann. As well as my office mates and friends at Freiburg, in particular Marc Pfeifer, Sebastian Böttcher, Benjamin Völker and Jessica 'Pia' Beckert.

I am also grateful for the brilliant researchers that I met along the way, which have all in one way or another inspired this work: Fahim Kawsar, Albrecht Schmidt, Thad Starner, Hans Gellersen, Andreas Bulling, Oliver Amft, Jamie Ward, Ulf Blanke, Daniel Roggen, Kai Kunze, Michael Beigl, Silvia Santini, Tilman Dingler, Markus Funk, Nan Zhao and Tobias Schubert. Also for the various thoughts that found their way into this thesis through the collaboration with students, particularly: Martin Philipp, Brahim El Majoub, Nils Schwabe, Martin Jänsch, Tobias Shultes and others. Also I extend my gratitude for the support received through the Google Glass Research Award, and the subsequent collaboration with Mathias Wille, as well as Ulrich Hold and Nina Schelter for running the experiment trials (sorry for asking participants to cut onions for several weeks in your basement).

Finally, I would like to thank my family and friends for their help and support. My mother and father, Marianne and Erich Scholl, for their support and faith in me to finish this thesis. My sister Daniela Stricker, my brother-in-law Daniel Stricker, and my nieces Annika and Rebbeca, with whom I spent too little time. My closest friend Lukas

Rauch, and Linda, who supported and inspired me throughout this thesis.

ABSTRACT

The recognition of human activities from (body-worn) sensor data promises novel applications and seamlessly blending the interaction with computers into everyday live. Up to the point where no explicit interaction with computing system is required, and relevant information is directly drawn from such sensor data.

Detecting activities from sensor data faces several challenges, both in technical and scientific areas, which are addressed in this thesis. From a technical perspective, storage and subsequent processing of large amounts of data can easily get cumbersome. An approach based on a multi-media container format, that compresses and stores multiple sensor data stream, including video, audio, motion, other sensor and ground truth data, in a single streamable file is proposed. Furthermore an Activity Recognition framework based on Unix processes, that is flexible, language-agnostic and parallelizable is investigated as well.

The major scientific challenge is to sample naturalistic datasets encompassing particular human activities. Core to this thesis is the idea of *fully* automating this data collection process by first trivialising the detection of the activity. Following the assumption that tools are used for particular tasks that are indicative for an activity, these artifacts can be instrumented to detect activities. This requires significant design effort for such tools, which can later be replaced with more general sensors. This idea is illuminated with two applications: detecting smoking from wrist motion and detecting process steps in wetlabs as navigational cues for video recordings and displaying task lists.

For smoking detection a dataset encompassing 351 smoking instances was collected with an ensemble system of Smartwatch, Smartphone and an instrumented lighter. A machine-learning recognizer, a trivial detector, and questionnaire elicitation of smoking behaviour are compared. In the wetlab, a systematic study including 22 participants for an entry-level experiment, a deployment of Google's Glass for task guidance in a university-level course, and a recording system at a research lab is presented.

ZUSAMMENFASSUNG

Die Erkennung menschlicher Aktivitäten anhand von (am Körper getragener) Sensorik ermöglicht neue Applikationen und eine nahtlose Einbettung von Computer-Interaktionen in unser tägliches Leben. Durch die Ableitung momentan relevanter Informationen aus Sensordaten, ist keine explizite Interaktion mehr notwendig.

Die Detektion von Aktivitäten aus Sensordaten besteht aus mehrere Herausforderungen, einerseits technischer und andererseits wissenschaftlicher Natur. Diese werden in der vorliegender Arbeit präsentiert. Von technischer Seite kann die Speicherung und Verarbeitung großer Sensordaten schnell unhandlich werden. Hier wird ein Ansatz vorgestellt, der auf der Speicherung mithilfe eines Multi-Media Formats basiert. In diesem Format lassen sich Video, Audio, Bewegungsdaten, weitere Sensordaten und Ground Truth Daten, in einer einzigen, streaming-fähigen Datei ablegen. Weiterhin wird ein auf Unix-Prozessen basierendes Aktivitäts-Erkennungs Framework, das flexibel, programmiersprachenunabhängig und parallelisierbar ist, vorgestellt.

Die wissenschaftliche Herausforderung besteht in der Aufzeichnung lebensnaher Daten, die menschliche Aktivitäten beinhalten. Eines der Ziele dieser Arbeit ist diesen Datenaufzeichungprozess *vollständig* zu automatisieren. Dies betrifft insbesondere die Aufzeichnung von Ground Truth Daten. Folgt man der Annahme, das Werkzeuge nur für einen bestimmten Zweck eingesetzt werden, lassen sich diese instrumentieren und dienen damit der Erkennung von Aktivitäten. Dies erfordert ein hohen Entwicklungsaufwand, der sich allerdings nach der Aufzeichnung möglicherweise durch die Erkennung mit einem allgemeineren Sensor ersetzen lässt. Diese Idee wird anhand zweier Beispiele beleuchtet: dem Erkennen von Rauchergesten mithilfe von Bewegungssensorik und der Erkennung von Prozessschritten in Biologie-Laboren, die zur Navigation von Videoaufzeichnung und Abarbeitung von Aufgabenlisten dient.

Zur Detektion von Rauchergesten wurde ein Datensatz mit 351 Instanzen mithilfe eines Ensembles aus Smartphone, Smartwatch und instrumentiertem Feuerzeug aufgezeichnet. Eine Erkennung basierend auf einem maschinellem Lernverfahren, eine triviale Erkennung, und die Erfassung des Rauchverhaltens mithilfe eines Fragebogens wurden verglichen. Im Biologie-Labor wurde eine systematische Studie mit

22 Teilnehmern, ein Einsatz von Google Glass als Anleitung in einem Praktikum auf Hochschulniveau, und als Aufnahmesystem in einem Forschungslabor, durchgeführt.

CONTENTS

INTRODUCTION

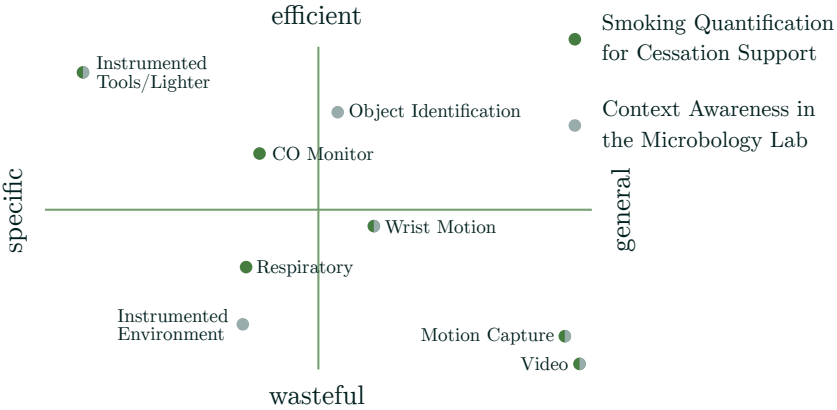


Figure 1.1: The design space of wearable and instrumented artifacts for smoking behaviour detection and context awareness in the microbiology lab. Simple sensors, which only allow to track very specific activities, can be made very power- and computation-efficient. More general sensors, employing wrist motion or motion capture can detect more activities albeit requiring more resources.

The development of tools was the first step in the evolution of mankind, enabling and augmenting tasks the human body on its own is not capable of. Integrating computation- and communication-enabled artifacts into our lives is the extension of this process for the human mind. Computers in their various current forms are realizations thereof, as computing excels in storing and finding information, and as such augment human memory capabilities - very much like Bush's original vision of the MEmory EXtender (MEMEX) [1] device. Today, about 80 years later, it is technically feasible to build such systems. Systems, which extract cues from body-worn and tool-embedded sensors to quantify behaviour and manual tasks, index their recordings and provide their user with novel remembrance capabilities.

Both conceptual and technical challenges in building such systems remain though, and a selection of these are addressed in this thesis. Conceptually, the context of a user needs to be recorded in a way that allows for useful cues. To be useful, such cues need to evoke past memories or contain recordings which allow to extract minute details. This can include video/audio recordings of executed actions, or simply counting events in a reliable manner. In the course of this thesis

two particular applications are investigated: semi-fixed procedures for manual tasks executed in wetlab environments, and the quantification of smoking behaviour in day-to-day life.

On first glance these two applications seem quite different. However, they share a surprising number of technical challenges. Memories can be evoked, and details extracted by browsing video/audio/motion data recordings by summarizing the observed human actions, e.g. which experiment was executing last? Was I executing the pipetting step in my last experiment correctly? How many cigarettes did I have past week? When am I smoking most often? The context of the user in such cases is recorded with multiple sensors, body-worn or tool-embedded, and presents the first technical challenge: (i) recording an ensemble of distributed sensors in an *energy-efficient*, *reliable* and *synchronized* way. The second challenge concerns the (ii) automatic interpretation of the recorded data, to extract high-level descriptions of observed human activities, often referred to as *Activity Recognition*. Due to the complexity of observed activities, machine learning algorithms which map data to these *manually created* high-level descriptions are often required. The challenge is to (iii) efficiently *parameterize* or *train*, *build*, and *deploy* such machine learning systems. Specifically, this requires large amounts of training data recored in *naturalistic settings*, which presents the final challenge. Traditional approaches require at least one human to close the so called (iv) *semantic gap*, to connect a data recording with a high-level description. This is cumbersome, costly and unreliable. Building system which directly integrate *ground-truth* collection can greatly mitigate this challenge.

These challenges are addressed by first presenting the Unix Filter approach to Activity Recognition that enables different Activity Recognition approaches to be quickly tested, as well as a data format that provides enough flexibility for both application sets. This appraoch is validated by its application to the detection of smoking from wearable sensors and from a Smartlighter. Addtionally it is tested in a wetlab environment for a distributed recording setup and subsequent detection of activities.

1.1 UNIX FILTERS FOR ACTIVITY RECOGNITION

The Activity Recognition approach was created by following the Unix philosophy of designing programs to do one thing, and to do it well. With this in mind, Unix processes that handle each step in an Activity

Recognition chain were built. These programs encapsulate one particular task, which can be individually parameterized or even replaced. Only the exchange data format between each step needs to be specified, which in the proposed approach is either a text or binary format. The former can be easily interpreted and modified, even by humans, while the latter provides lower overhead and the cost of less transparency and increased implementation effort.

Core to this approach, is the definition of a Unix command that handles the overall learning and classification task from the command line. Hyper-parameters, like the learning algorithm, feature set, or segmentation approach are supplied through command line argument. This in turn allows for process-level parallelization, especially on computing clusters, of grid-search and cross-validation - two of the core tasks when evaluating a machine learning model for Activity Recognition.

Additionally, a multi-media container format that encompasses video, audio, motion, sensor, ground-truth, and classification data in a single file is investigated. This provides the foundation of the remembrance support for wetlab workers, as well as a curation format for the long-term storage of smoking motion datasets.

1.2 QUANTIFICATION OF SMOKING FOR CESSATION

Smoking is called the number one reason of premature, preventable death worldwide. However, scalable and effective smoking cessation support is still elusive. One reason for this could be that efficacy measurements of cessation are currently limited to questionnaires, telephone interviews, small scale observational studies and ecological momentary assessment with Smartphones. Body-worn sensors and instrumented artifacts could provide novel, and objective insights into the behaviour of nicotine-addicted people.

Two approaches are compared in this thesis: (i) detecting smoking from wrist motion recorded with sensors in Smartwatches and Fitness tracker bands, and (ii) detecting lighter usage events with a Bluetooth-enabled lighter. Design decisions for the Smartlighter are presented, and the lighter was used for different field studies. It's main purpose was to collect ground-truth data for the motion detection approach, and was tested with a total of 17 smokers over the course of several days. Based on this collected dataset, a symbolic detection approach that provides a baseline recognition approach for *trivial* smoking gestures was evaluated. With this approach it is possible to show that non-trivial

smoking gestures were recorded. A machine-learning approach to smoking was subsequently developed, which detects these non-trivial gestures. The instrumented artifact approach requires less energy than a continuous motion recognition, albeit being a less general solution. The energy requirements of both solutions are presented in chapter 4 as well.

To show the benefit of an objective smoking measurement, the results of a questionnaire study are compared to the results of a measurements taken with the instrumented lighter. Evidence that smokers tend to overestimate their actual consumption, and evidence for a strong difference between asking smokers for their most common time of day spent smoking and what is actually measured with body-worn sensors or instrumented artifacts, was found. Eleven participants, recruited at the Universtiy of Darmstadt, participated in this study using one of the early prototypes of the Smartlighter. The mean participation time was eleven days.

1.3 MEMORY AUGMENTATION FOR LIFE SCIENTISTS

Experimental work executed in laboratories has been the target of multiple research projects already. The common goal is to augment manual work done by researchers in chemistry and biology laboratory, where the possibilities of manual interaction with computing machinery is naturally limited. Due to wearing protective garment, and to having to stop the task at hand for interacting with computers, other means of interaction would be helpful. The repeatability of experiment is the main goal when conducting these experiments, as only then valid conclusion can be drawn. Hence, technology which improves the documentation capabilities of an experimenter, and which improves access to information, is sought.

Similar to life-logging applications, we follow the idea of using body-worn sensors, particularly video, audio, and motion, to record everything while conducting a wet lab experiment. Such recordings in themselves are hardly useful, as the only cue to navigate them is the time-of-day. However, due to recording motion, executed activities can be detected and used as additional navigation cues. If, furthermore used objects were recorded, for example by radio frequency identification (RFID), more specific queries to a database of such recordings can be supported. For example, asking for all recordings where a particular compound was pipetted, to check whether this pipetting step was the

reason for an experiment being unrepeatable. This can be called offline querying and requires an Activity Recognition module to map motions into a queryable space. Once this module was established, it can also be used for online querying, i.e. retrieving information about the task at hand without the need for an explicit query. For example, details about the current workflow process can be retrieved by activating a head-worn display, which already retrieved information about the step at hand without any further input.

For this application, the applicability of multi-media containers for storing video/audio/motion and subtitle data is shown. The latter contains the results of an Activity Recognition, and provides a query space that can be efficiently searched. Sensor data is recored by augmenting the scientists instead of the environment, to gather deployment flexibility. For this, a study to detect steps of an entry-level wetlab experiment is presented, a deployment during a teaching lab course for eliciting the requirements for task guidance, and a study on a novel segmentation approach for such recordings is presented.

1.4 CONTRIBUTIONS AND THESIS OUTLINE

The thesis provides the following contributions to the current state-of-the-art:

Concept for Multi-Modal Activity Recognition orchestrating the recording and recognition of simultaneous sensor streams can easily become cumbersome. To overcome this issue, multiple simplifications are presented as well as a container format for such recordings, which is both *efficient* and *general*. It includes support for video, audio, and other continuous sensor data, as well as annotations, recognition results, and sparse sensor data in a common format. Furthermore, the performance and applicability of Unix filters for Activity Recognition are investigated.

Two Case Studies of the proposed Activity Recognition concept A smoking detection system based on a Smartlighter and wrist-worn inertial sensors is presented. Furthermore, a more general system which allows to quickly navigate point-of-view (POV) videos based on instrumented wet lab tools and wrist motion. To record this data, a common data format is presented and a recording infra-structure encompassing Android, Linux and other networked devices is shown. Furthermore, the challenges in recording, and conducting these

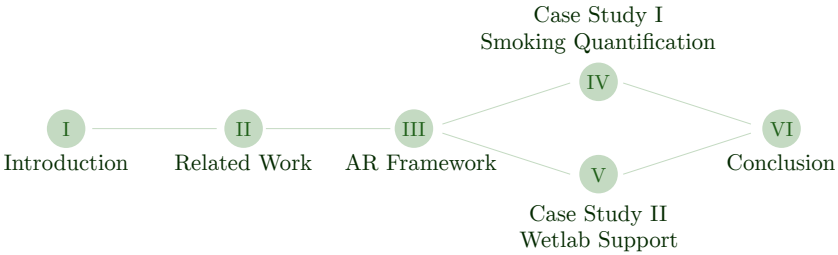


Figure 1.2: Outline of this thesis. After an introduction to the Activity Recognition (AR) framework, two case studies are presented. Both highlight the features of the framework.

studies are highlighted, as well as indications on the applicability for the chosen recognition approaches are extracted.

This thesis is split into four parts (see Fig. 1.2). First, work related to Activity Recognition, wearable smoking detection and support systems for wetlab work are described. The second part highlights the Unix filter approach to building *recognition pipelines*. This allows for combining several machine learning frameworks, switching to production and parallelizing various steps with minimal effort. In this chapter binary and textual data formats, as often used in related work, are compared and a novel format is suggested. This allows for multi-modal recognition approaches, including various sensors and switching between different datasets easily. In part three, the first case study on recognizing smoking from motion and from an instrumented artifact is described. A Smartlighter is combined with a wrist-worn motion sensor to quantize smoking sessions. Such data can be used for evaluating interventions as well as enabling just-in-time interventions. Part four describes the application of a multi-modal recognition approach as a memory augmentation tool in wetlab environments. The researcher, instead of the environment, is outfitted with various sensors. Cues are automatically extracted from the sensor recordings, to provide a browsable recording of manual wetlab work. Combined with an egocentric (or other) video and audio recording this enables a memory augmentation system. The final chapter concludes the thesis, and gives an outlook on further possible research scenarios.

RELATED WORK

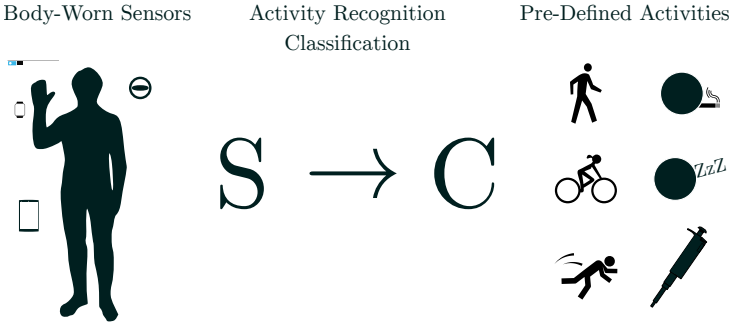


Figure 2.1: Overview of Activity Recognition systems. Body-Worn sensors record application-specific sensor data. A *classifier* maps (continuous) sensor data (S) to categorical labels of activities (C). Designing, building, deploying, and evaluating such wearable systems are the major challenges when creating novel systems.

Activity Recognition, the problem of detecting activities from (body-worn) sensor data, has been researched quite extensively over the past years (see Figure 2.1). The software components and their interaction, is well known by now and boils down to a processing *pipeline* where frames of sensor data are fed in and labels for each data frame are returned. How to best gather *reliable* sensor data in a scalable fashion, which sensors to apply for a given application, how to built learning pipelines, how to evaluate them and how to tune the parameters for each step are still open issues. Deep Learning, which combines learning, parameter tuning, and evaluation into a single pipeline is also applied in the field of Activity Recognition and shows promising results. An overview of commonly used approaches over the past 20 years is given here, highlighting the different steps involved in predicting categorical labels from unseen sensor data and different application areas. To this end the following questions will be investigated in this section:

- Which application/activities were investigated?
- What are the major challenges for wearable Activity Recognition?
- Which sensors were used, how were they attached?
- How was the system evaluated?
- Is there a difference between in-lab and in-the-wild studies?

- Which system exists to quantify smoking from body-worn sensors?
- Which manual task support system are described in the literature, specifically for wetlab environments?

Being directly related to the contributions of this thesis, the latter two question are systematically reviewed. The remaining questions are addressed based on survey works and with the goal of covering a large area of works. Generally all reviewed works follow an architecture as depicted in Fig. 2.1. A physical phenomenon (correlated to an activity) is captured with a sensor and mapped to a pre-defined set of labels designating activities. The works are set apart by experimental design, methodology, application, sensors, and evaluation strategy. [2, 3] provide a recent overview of application for HAR, while [4] provides an introduction to the technical challenges.

For a fundamental understanding of work involving any machine learning approach is Peter Norvig's formula of artificial intelligence [5]:

$$act^* = \underset{a \in actions}{\operatorname{argmax}} E(Utility(Result(a, s))) \tag{2.1}$$

which states how a computing system / artificial intelligence will chose the next action (act^*) from a set of possibles actions. s corresponds to the state of the environment and user, in the following also called the *context* of the user, E to some kind of (probabilistic) reasoning and *argmax* to a search algorithm. $Result()$ predicts how a particular action will change the world state s, and $Utility()$ encodes the goals of the system. Each part of this equation has its own set of challenges, however Activity Recognition is limited to detecting the state s from sensor data recorded with wearable or instrumented artifacts.

2.1 APPLICATIONS OF ACTIVITY RECOGNITION

Wearable systems, and particularly detecting activities from sensor data, promise to digitize otherwise hardly detectable action of humans - to augment capabilities and senses. A continuously worn computer system can improve one's intellect, memory, creativity and communication, and also one's auditory, visual, olfactory, gustatory and tactile senses [84, 85]. Such systems balance power requirements, heat dissipation, computational power, network connectivity, wearability, obtrusiveness, task-specific interfaces, and particularly *robust recognition* of its wearer intentions and actions. This recognition minimizes the amount of explicit interactions with the system to seamlessly blend into the wearer's

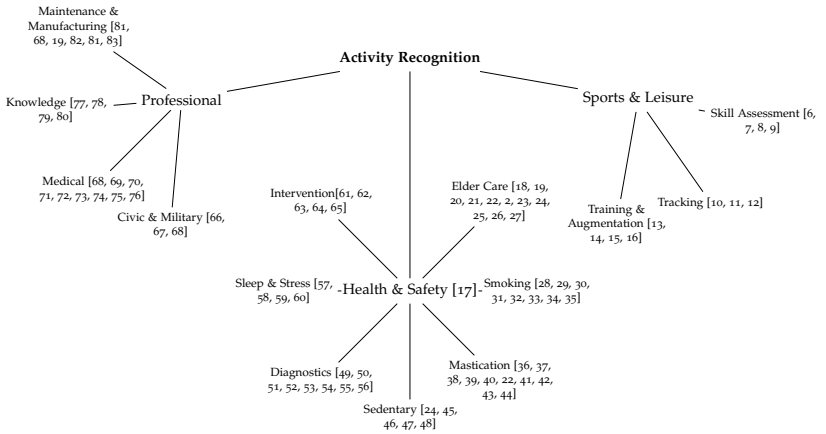


Figure 2.2: Applications of Activity Recognition, in which body-worn or instrumented artifacts render indications on the task at hand.

normal life and the task at hand [86, 87, 88]. Or, as Starner [89] defines Wearable Computing where

> . . . the interface becomes a natural extension of its user.

Besides this abstract idea of Wearable Computing, and in particular Activity Recognition, there are a number of concrete applications that clearly benefit from detecting the user's context from body-worn sensors. The largest of those areas is Health & Safety [4, 90].

Declining mortality and fertility rates create an ever-increasing elder population [91]. Wearable technology is sought to reduce the burden of care-givers and to enable elders to live independently for longer periods of time. *Activities of daily living (ADL)* and potentially life-threating situations like falling are prime targets for detections and have received a lot of attention [18, 19, 20, 21, 22, 2, 23, 26, 27]. Early detection and interventions that could result in chronic diseases like sedentary lifestyles [24, 45, 46, 47, 48, 3], smoking [28, 29, 30, 31, 32, 33, 34, 35], obesity and unhealthy diets [36, 37, 38, 39, 40, 22, 41, 30] are often targeted as well. By quantifying and continuously monitoring [55, 56], suggestions for healthy lifestyle changes and other interventions are the goal [62, 63, 64, 65]. Playing a rhythmic audio signal was shown to assist during freeze-of-gait for Parkinson's disease (PD) patients [61]. Diagnostics, for example from gait monitoring for the onset of PD [49], or mental health issues [56, 55] are further examples.

Sport & Leisure applications are probably the most well-known deployments of Activity Recognition Systems. These relieve their users of manually noting down details about a workout, for example the track length and duration of a bike ride. Promoting healthy lifestyles [10, 11, 25] through feedback on physical activity is motivated by the lack of physical activity being a major risk factor for Non-Communicable chronic Diseases (NCD). These NCD are believed to cause 60% of worldwide deaths [92], including cardiovascular diseases and certain cancer types. In combination with mobile phones, activity recognition can personalize training plans and assess the skill of its wearer, for fitness [6], weightlifting [7], martial arts [9], swimming [15], or climbing [8] to name a few. Wearable sensors also estimate the energy expenditure of physical activity in different measures [93]. An early study on the UbiFit Garden system [12] has shown the efficacy of such physical activity feedback. Example Leisure applications include Stochasticks [13] which augments the Billiard experience, augmented reality games [16] and instrumented shoes to increase dancing skills [14].

In between the last two areas is the longitudinal assessment of treatment outcomes, mental disorders, and tracking rehabilitation progress [53], for example for stroke, Parkinson's, and multiple sclerosis patients [51]. These patients suffer from impaired motor skills, which can be picked up with wearable sensors. This is also coined mobile health (mHealth) in this scope, which allows for remote monitoring, home-based therapies, efficient daily care, but also for improved clinical trials when used as an assessment tool. Particularly measuring long-term effects with body-worn sensors allows for novel diagnostic insights [50] in the patient's real-life environment. Demonstrated for example by the AMON device [94], which monitors multiple cardiovascular parameters. Wearable assistants, which intervene just-in-time have been demonstrated for freeze-of-gait condition for PD patients [95], intervening for COPD [96], or acute epilepsy seizures [54].

In the last example sensors pick up the *context* of their users, whether this is a (longitudinal) medical condition or their current intent. Context Aware computing focuses on the idea of building systems which utilize *context*, detected with body-worn or environmental sensors, to *augment* its user's capabilities [88]. Having a clearly defined idea of the intention or information needs of a user is a challenge, hence the application of Activity Recognition to support activities of professions, i.e. *professional* activities or "everyday" life needs. People with memory impairments, e.g. dementia patients, but also knowledge workers can benefit from

life logging applications [97, 98]. Wearing a camera (and other sensor modalities) that continuously records the user's experience indexed by their activities augments normal human memory. However, to move "beyond total capture" [99] requires further insights into how memory is retrieved and organized. This is often task-specific: a remembrance system [79] can index and answer queries to such an "external" database [100]. Different usage scenarios like conference assistants [77], meeting assistants [78], but also providing in-situ conversation assistance [101] or public speech helpers [80] were investigated. Commonly a head-worn display provides information to the users, while context is detected with body-worn sensors. Soldiers [67, 102] are supported in their training, providing real-time feedback on their physiology, team positions, and tactical overviews. First responders, like firemen, benefit from similar systems [68, 66].

Due to the possibility of structuring professional activities in a way that facilitates recognition with sensors, these activities are targets for recognition as well. Particularly the commercial availability of head-mounted displays (HMDs), like Google Glass, has increased the interest into the detection of process steps for manual activities. The idea is to provide guidance for the task at hand [103] (also sometimes called proactive Documentation), to increase task safety [104, 66], for remote support [105] and automated quality checks [106]. Wearable computers are particular useful in the following areas [107]:

Maintenance & Manufacturing Aircraft maintenance was one of the earliest targets for assistant systems, where also instrumented artifacts [108] contributed to the detection task. Task guidance systems for aircraft inspection [109], armored vehicles [110], workshop machines [111], or car manufacturing [81, 112] are examples. This also includes hands-free access to operation manuals and remote assistance [113].

Civic & Military Bio-Hazard handling professionals [66] are one group of professional, where protective clothing and the task restricts the possible amount of manual interaction with a computer system - hence novel ways of explicit interaction need to be found. For example, thick gloves impair its users ability to type or use touch interfaces. One solution is the yo-yo interface [114], a single hand tethered device controlled for gesture input. Tracking the movement of an intervention group, would allow to get a tactical overview [115] and plan emergency response during disaster recovery [68, 116]. Similarly for soldiers [67], whose physiology may also be tracked, as well as astronauts [117, 118] during EVA/IVA missions.

Knowledge Work Particularly the monitoring of manual workflows, but also supporting data collection for mobile workers, can be enhanced with the help of an activity recognizing wearable systems. Detecting the stage of a workflow decreases the amount of explicit interaction. Switching recipe steps automatically [119] instead of having to verbally instruct a computer system to switch to the next step is one example. But also checking safety-related steps (was the bio-hazard container closed again?), or providing steps as navigation cues for video material (a YouTube-tutorial which jumps directly to the segment of relevant information) are possibilities. Wood workshop and bike maintenance activities were detected with body-worn accelerometers and microphones [120]. Assembling IKEA furniture [121], and cooking [21] are further examples of detecting steps in procedural knowledge to provide in-situ information. Supporting field work by providing recording support, for example in biology, was also demonstrated [122, 123]. Another string of work is engaged with managing attention and interruptibility [124], trying to build contextually-aware systems, which delay notifications to a "fitting" time.

Medical professionals, like doctors and nurses, are also thought to benefit from wearable technology and activity recognition. For example, accessing patient records, triage support, note taking in forensics and other fields, self-reflection and improving education of surgeons are possible applications [68, 69, 70, 71, 72, 73, 74, 75, 76, 125, 126].

2.2 CHALLENGES AND ECOLOGICAL VALIDITY

The challenges of designing wearable computing system are well documented [84, 85] and still exist for current system. The foremost challenge is to provide *power* to miniaturized computing systems, particularly since the energy density of batteries improves at a much lower rate than computing power. Energy scavenging from the human body might be a solution for wearable computing though [127]. *Heat dissipation*, *networking*, *privacy*, *interface design*, and the *creation of intellectual tools* are further examples. Beside the technical challenges, the *robust* detection of the user's current context from sensor data remains to be an open question, which is one of the goals of this thesis.

Particularly the *ecological validity* of correlating sensor signals to a recognized human activities is of importance for its practical use. This validity refers to how well the setup of a study (free-living condition) reflects the real-world of the subjects under study [128]. This definition

of ecological validity is however criticized in [129], which offers the alternative term *representative (study) design* to ask study authors for justifications of generalizing results beyond their study design. These definitions are originally from the field of psychology, but also applicable for Human-Computer Interaction and Activity Recognition, which typically involve user studies, the value of *naturalistic studies* was argued for [130, 131, 132, 133] in these contexts.

To what extent a user study can reflect the actual phenomena under study is mostly dictated by *ethical* considerations. For example, the famous Stanford prison experiment, which has lead to the introduction of institutional ethical review boards, is an unethical study where participants were lastingly harmed during the experiment. A famous example for Activity Recognition are setups for fall detection from sensor data. Such studies can only be of limited ecological validity, since participants cannot be hurt when falling in an ethical way. But also *privacy* when recording body-worn sensor data is an issue, and participant's data should be limited to the specific detection task as well as deletion of personal data must be possible.

Not only the study setup but also the evaluation methodologies involved in testing the performance of a machine learned classifier is a further challenge when creating valid, generalizable results. As pointed out in [134] the type of cross-validation can introduce a bias which leads to an overestimated generalization error. However, by limiting the amount of possible correlations when segmenting a time-series this can be avoided. Still, the cross-validation methodology, whether leave-one-participant-out, k-Fold or random subsampling, needs to be chosen to support the actual detection hypothesis. [135] describes several evaluation methods for wearable computers, as well as some of the pitfalls endangering the validity of studies, and [136] surveys methodologies in use in HCI. One outstanding challenge is the *robust collection of ground truth* data, as exhibited in [137]. In this study, in-lab and free-living data for eating recognition is combined. One finding is that in-lab data, i.e. controlled for confounding variables, allows generalization to free-living conditions. However, learning from data in the free-living condition leads to worse performs - this is probably due to low-quality ground truth data. Class imbalance, or encoding unwanted priors due to the structure of the dataset is another shortcoming to watch our for during evaluation. Common solutions include the up- or down-sampling of classes [138].

Performance Metrics which allow for detailed interpretation of

continuous activity recognition systems are proposed in [139, 140]. Instead of the prevalent scoring methodology in machine learning [141], which assumes that each classification result is statistically independent, these metrics take the implicit correlation of time-ordered results into account. Instead of scoring single frames, event identification are scored. Not only mismatches between classification and ground truth of a single frames but also the surrounding frames are taken into account for scoring, providing more insight into a classifier's performance.

Human Activity recognition is furthermore complicated due to its problem definitions [18, 4, 19]. *Concurrent activities* or the ability of humans to do multiple activities at once, is challenging to recognize since usually only singular activities are sampled. Superpositions of multiple activities can not be easily calculated from these samples. *Confounding activities* or also the ambiguity of interpreting a particular activity label is problematic. For example, *smoking* could refer to consuming a cigarette or consuming a cigar, for which different assumption can be made. But also eating or drinking might look similar to smoking when looking at arm motions. These two challenges exist for almost any Activity Recognition system.

2.3 ACTIVITY RECOGNITION APPROACHES

Activity Recognition approaches are rather widespread. Here, we follow the definition from [4, 142] of the Activity Recognition Chain (ARC), which splits these systems into the following steps: *pre-processing*, *segmentation*, *feature extraction*, *training and classification*, and *post-processing*. Each step will be discussed separately, and while not exhaustively discussing all approaches, an overview will be given. The overall goal of an ARC is to encode the *similarity* of sensor data time-series, which belong to executing the same activity

Pre-processing typically involves sensor-specific processing of analog-digital-conversion (ADC) results. For example, gyroscope data might require baseline removal, and low-pass filtering to remove sensor noise. Calibration is another common operation. For some applications, specific frequency bands can be filtered out. Normalization, unit conversion and re-sampling are further operations. Re-sampling is often done for storing sparse or non-periodically sampled sensor data. For example, GPS data might only be sampled when movement is detected, which is subsequently re-sampled to a fixed interval to simplify processing. Pre-processing is therefore involved in increasing the data quality by

removing artifacts that can be safely assumed to be not correlated with the pattern to detect. Quantization, i.e. reducing the resolution of data samples, can furthermore reduce sensor noise and decrease computational complexity at later stages. In Activity Recognition, the results of such a quantization step are called *motifs* [140] and often expressed as strings. The overall output of the pre-processing step are several vectors $s_i = (d_0, d_1, \dots d_{k_i})$, where i enumerates the specific sensor and k the sample at a specific time step. The k_i-th sample can differ for each sensor, as each sensor might be sampling at a different rate, i.e. each sensor runs with its own clock. When recording multiple sensors in parallel, clock synchronization becomes an important technical challenge [119, 143, 144].

For time-series classification, an interval over which to classify needs to be chosen. Such intervals are called segments and accordingly this processing step is called *segmentation*. The choice of segmentation strategy is an important parameter of an ARC. Two simple approaches are segment via ground-truth, i.e. to to assume a "perfect" segmentation exists, or to apply a sliding-window of pre-defined duration with or without overlapping the segments. Segmentation can also be formulated as detecting change points in time-series data, also termed piece-wise approximation (PA) [145] or spline interpolation. By assuming an error limit to such a functional representation, the segments correspond to the support points of these interpolations. Wavelet representations [146] lend themselves well for periodic signals, while spline approaches are better suited for non-periodic signals. Thresholding a running signal property, like the signal's energy or variance, is a simplified form, which is commonly used. Also, unsupervised clustering algorithms can be applied to identify segments with similar signal properties. For segmenting multi-sensor series, usually only a single series is segmented, and the extracted segments used for the remaining series as well. This however requires that multi-rate recordings can be split accordingly, for example requiring a minimum segment duration that equals the lowest sampling rate, or by re-sampling to a common rate beforehand. The output of this step is then multiple vectors $s_{i,j} = (d_0, d_1, \dots d_{k_i})$, where j designates the j-th segment, and k_i a sensor sample on the sensor's local clock.

Prior to learning patterns, which are indicative for a particular class $c \in C$ from segmented sensor data $s_{i,j} \in S$ (cf. Figure 2.1), the sensor data is *extracted* into a *feature vector*. This serves two purposes: first to minimize the *Curse of Dimensionality* by limiting the amount of

possible data combinations that represent the problem. This means, the larger the feature vector, the more collected data is required to capture the space of classes. Second to remove the continuous nature of the recorded sensor data, i.e. to create a sensor- and rate-independent representation of each sensor vector called the feature vector $f_j = s_{i,j}$. This can be achieved in a multitude of ways and is roughly classified into three categories: signal-, model-, and domain-based.

Signal-based methods are the simplest form of features that can be calculated. These involve the calculation of statistical moments (mean, variance, skew...) , median, range of values, root-mean square, number of peaks, energy and zero-crossing rate to name the most common. Beside these *time-domain* features, features in the *frequency-domain* are also commonly found. After applying a Fourier or other type of transform, the first n components (frequency and magnitude), spectral energy, n-largest coefficients can be extracted. The large number of possible features exhibits a full treatment here, however a comprehensive survey [147] describes them in a larger number, and proposes a method to automatically select the most representative for a given dataset. Compared to Computer Vision, where standard descriptors like SIFT are common, there is no such standard feature set for Activity Recognition, which is probably due to the heterogeneity of the used sensors.

The second class of features are model-based and are related to the previously mentioned segmentation step. A mathematical model with a *fixed* set of parameters is matched onto the current segment, and the parameters make up the extracted features. This can involve (piece-wise) linear, polynomial or trigonometric models, or fitting the parameters of statistical distributions. Due to modelling the time-behaviour of a continuous signal this class of features is particularly suited when parts of the sensor signal can be predicted from a certain point in time.

Domain-based features in contrast, take knowledge and assumptions into account that are known from the process at hand or from the recording setup. [148] measured multiple points on the human body, and estimated the position of each point by a kinematic chain. Sequences of these positions, called primitives, were then used for classification. Expressing activities in terms of threshold on a human skeleton model for weight lifting exercises is demonstrated in [7]. For eye-tracking, features as the number of saccades, fixations and blinks can be extracted that are particular for this application. For raw data of inertial motion sensors, such domain-derived features depend on the targeted application.

When selecting a feature set for a particular recognition task, domain- knowledge derived features are preferred. This, however, is only possible if enough facts about the recognition tasks are known. The alternative strategy of calculating and selecting a minimum set of signal- or model-based features can inform such domain-features. Model-based features lend themselves well for detecting repetitive or predictable time-series segments. The output of this *feature extraction* step are feature vectors f_j for each segment j.

Once *feature vectors* are extracted and a machine learned model is in place, the segments be classified. This means mapping from the feature space F to the class space C. The output of this step is then used in the specific application, e.g. the persons is walking, jogging, or running right now. For this, the model M carries a parameter set ϕ that contains the choice of pre-processing, segmentation, feature extraction, and the choice of hyper-parameters for each step. The output of this classification often allows for a probabilistic interpretation:

$$p(c|s_j, \phi) = M(\phi, s_j) \quad \forall c \in C \tag{2.2}$$

, i.e. p designates the probability that a particular segment j belongs to class c. A point estimate, for example the class c with highest probability, is then chosen as the "prediction" of the classification model M. Other *post-processing* steps also take the timely-correlated previous or following predictions into account. This is called smoothing and often applied to remove spurious mis-classifications. Not all classification models estimate the probability of a sample belonging to a class. A common technique to estimate this probability from such models is known as "Platt Scaling" [149].

By estimating the probability of each class, a NULL-rejection scheme can be implemented, also known as the activity spotting challenge [150]. This refers to the complication that not all *irrelevant* activities can be ecologically sampled. Imagine a longitudinal sensor recording, where the activity of interest occurs only sporadically. When applying a model to spot this activity, it might be spuriously detected on irrelevant segments. Those spurious detection can be sometimes rejected by a threshold on the probability of this classification. Furthermore, the model should only be trained on the activities of interest, and not on the irrelevant segments (which we label with NULL). The combination of learning only the relevant activities, and suppressing irrelevant segment by a threshold on the class probability, is a more general solution than

explicitly learning the NULL class from a limited sample set [151].

For example, imagine an accelerometer dataset where the NULL class consists only of samples of a single value. The activity of interest consists of actual movement. Training a machine learning model on the NULL class explicitly would be a trivial solution, as only a comparison to the single value is required. When this value is not seen during classification, the prediction would be the activity of interest. This model would give perfect classification scores, however when exposed to new data during deployment would result in erratic predictions. Such a scenario is not that uncommon, since this happens when no "background" data and only the activity of interest is recorded. In such cases, only a NULL-rejection scheme would result in a realistic performance estimation.

The question which model to train, and how to train these models is an ongoing challenge. For activity recognition, two distinction of classification algorithms can be made: whether an algorithm is unsupervised or supervised, and if its input vector needs to be of static or dynamic size. For example, the simplest feature extraction scheme is to use the raw time-series of sensor data and apply only a specific vector norm (see Fig. 2.3 for examples). *Unsupervised* algorithms require the number of classes (but not their distinctive labels) and a vector norm $\|.\|$ as parameters. Clusters of feature vectors are then calculated by equalizing the distance calculated by the chosen norm $\|.\|$. The similarity of feature vectors is encoded in this norm. Depending on the choice of the norm $\|.\|$ also dynamically sized feature vectors, i.e. vectors of varying dimension can be used. Dynamic time warping (DTW) [152] and cross-correlation are examples for dynamic norms, while the euclidean distance requires the compared vectors to be of equal size. A common algorithm choice is the k-means algorithm, but also the DBSCAN algorithm was used successfully for activity recognition [153].

The ability of unsupervised approaches to recognize activities rests on the choice of distance metric $\|.\|$ on F, and whether the classes in C need to labeled. When such labeling is required and a distance metric can well separate the classes, a k nearest neighbours (kNN) algorithm will already provide good performance. However, in the most common case, that such a metric is not apparent, more sophisticated learning algorithm like Support Vector Machines (SVM), Random Forests (RF), or Neural Networks (NN) can be applied. These create more involved decision boundaries on the dimensions of F, but also require tuning of more hyper-parameters, for example the choice of SVM kernel. These

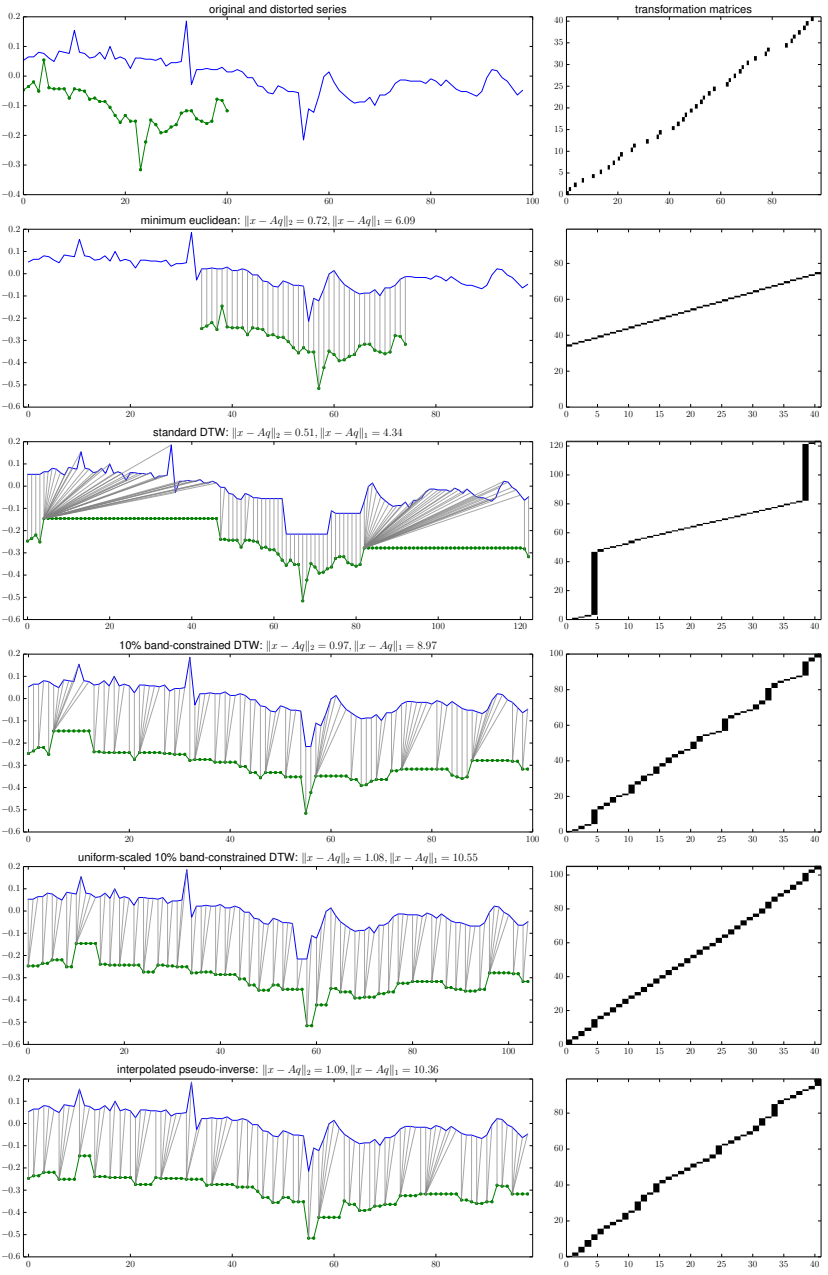


Figure 2.3: Different norms to define the similarity of time-series.

approaches work only on static feature vectors though, and domain-knowledge can only be encoded in the feature extraction step.

If, however, the extracted features are of dynamic size, time-dependent signal correlations are apparent, or if the duration of a signal pattern is dynamic, probabilistic methods can be applied. This includes Hidden Markov Model (HMM) or Conditional Random Fields (CRF) [121], which are both special cases of bayesian networks (BN) [154] or even Markov networks (MN) [155]. Besides the ability to handle dynamically sized inputs, these also allow to encode domain knowledge as dependencies between random variables. As such, these are powerful tools to model the dependency between body-worn sensor data and human activities.

No matter which machine learning model is chosen to represent the mapping from sensor data to activities, a particular set of parameters needs to be chosen, this step is called *training* the model. These parameters are derived from a *training dataset*. The performance of this trained model is estimated on a *validation dataset*, of which the mapping from sensor data to output labels is known. For this, ground truth labels are compared with the prediction of the just trained model. Typical machine learning metrics like Accuracy, F1, Prediction and Recall and specialized event [139] for activity recognition allow an interpretation of this performance.

Depending on the type of application, different cross-validation schemes need to be applied to the dataset. Cross-validation refers to the strategy on how the dataset is continuously split between training and validation data. A common choice is random sub-sampling of the dataset, which was shown to be overly optimistic [156]. This is usually applied to test the performance in a user-dependent way, i.e. how would a system perform for unseen data from the same user. A better alternative to is k-Fold cross-validation. Less commonly used, but allowing to get an idea on how well a recognition system would perform for unseen users, is to test all leave-k-participants-out (LKPO) splits, where k is typically set to one. Generally, the choice of cross-validation scheme depends on the application and the particular research question. Hence, when designing an ARC, the choice of cross-validation should be decided on first and carefully selected [139, 140].

Furthermore, the particular choice of ARC parameters, including the pre-processing, segmentation, feature extraction, machine learning model and respective hyper-parameters are application dependent. Hence, exhaustive empirical exploration of large parameter sets is

required for each application as there is no clear best approach [157]. Deep Learning, which optimizes this whole ARC chain on a particular performance metric is such an approach. In this thesis, a comparable approach is presented. Here, however, classic ARCs are parameterized and optimized, instead of limiting the model to Neural Networks.

2.4 WEARABLE SENSORS AND AUGMENTED OBJECTS

Finding the right sensors to measure application related signals from the human body or from application specific artifacts is an open challenge, however the body of research with novel sensing methods is constantly increasing in size. Here is an overview of the most commonly used sensor modalities and one paper to illustrate its use. Most well known are body-mounted inertial motion sensors which can be correlated with human activities [45]. Not every moving body part can be measured this way though, most notably eye-movement is tracked by electrooculography [4] or camera-based approaches [158]. A further sensing principle is the change of capacitance between two electrodes [159], which allows to measure deformations and also body-internal changes [160]. Muscle-based deformations can also be measured with force-resistive sensors (FSR) [161]. Body-coupled (e.g. via a stethoscope) and ambient audio signals are yet another modality, which however offers greater fidelity [162]. Also reading signals directly from the brain is under active investigation [163]. A recent overview of the over 15 physiological data points can be found in [164].

The mentioned sensor modalities usually have a single measurement point. Most of them can also be combined to form a sensor array that allows for higher spatial resolution. [165] presented a ultrasound tomograph for hand gesture detection, which offers greater resolution than previous electromyography (EMG) attempts [166], albeit at higher energy costs. EMG was also shown to benefit from an increased number of electrodes [167, 168]. Electrical impedance tomography is another form of signal array processing to detect fine-grained limb movements.

However, we argue that instrumenting activity-specific tools does not only trivialise the detection methodology, but is also more energy-efficient. The Mediacup [169], a coffee cup with wireless connectivity and temperature sensors, was one of the first instrumented artifacts. It recorded one's coffee consumption, as well as spontaneous meeting by detecting the density of collocated cups, allowing to derive new context information of its users. [170] shows that fill level estimation by

an RFID-tag attached to a normal glass cup is possible. This idea was continued in the Smart-ITs project [171], that investigated an active tag (an embedded computer) that could be used to instrument everyday objects. Pin&Play [172], similar in nature, provided an elegant solution to the problem of energy supply and communication by a dual-pole, conductive surface to connect multiple tags. A usage scenario for instrumenting boxes that contain tools or pills is investigated in [173]. This allows to track workflows, in a way similar to the proposal of [174], which instruments tools with RFID tags, which are detected by a wrist-worn reader. Situating displays at the positions where information is needed, e.g. displaying weather information at the wardrobe is another form of instrumenting an artifact (or providing augmented reality in this particular application). More recently [27] argued for augmenting a home with simple sensors for detecting purposes.

While the latter examples revolve around additional tags that are attached to artifacts, recent advances allow to integrate networking, detection and power supply directly into the artifacts themselves. [175] presented an e-cigarette augmented with a Bluetooth interface, which tracks its user's consumption and provides additional information like location, time and overall nicotine consumption. For normal cigarettes, the instrumented cigarette box [176] is one way of tracking the actual consumption of its user. In this thesis the design and implementation of a cigarette lighter to achieve the same goal is presented.

2.5 SMOKING DETECTION WITH WEARABLES

Detecting smoking from wearable sensors is mostly motivated by the possible impact an objective and longitudinal assessment of smoking can have on novel interventions, and insights into coping strategies for substance dependence [177]. Furthermore, developed detection techniques could be generalized to other activities that are *repetitive*, exhibit a *typical duration* and happen only *sporadically* throughout the course of a day. Based on this observation, different wearable sensor were investigated.

Inertial motion sensors, worn on the wrist, are most commonly used. Smartwatches, equipped with low-power MEMS motion sensors, would be a practical modality. The intuition is to detect repetitive *Hand-To-Mouth* gestures which signify a possible smoking session. The full set of magnetometer, gyroscope and acceleration measurement (9DOF) [29], gyroscope and acceleration (6DOF) [178, 31], and acceleration only

(3DOF) [32] were utilized. Rate-of-Turn (gyroscope) measurements are usually used for segmentation of the sensor data [29], while acceleration data estimates if the hand is at the mouth. Currently, no baseline dataset is publicly available, challenging direct comparison of these approaches. Recognition rates (F1) of larger than 80% were reported, pointing to the feasibility of detecting at least prototypical smoking instances - instances in which the smoker was standing, held the cigarette in the monitored hand and always moved his arm from waist to mouth.

Combinations with other sensors were also investigated. The location of a smoker is an indication for him smoking [33] as well. This can be picked up with GPS and WiFi scans. Another common modality are mobile inductance respiratory phlethysmographs (RIPs). These belt-like devices monitoring breathing rate and depth and are worn around the chest. Deep inhalations, common for a particular kind of smoking, can be measured and their repetitions is indicative for smoking [179]. Combining these with wrist motion was shown to increase classification scores [180]. Alternatively, and also one of the earliest wearable devices for monitoring smoking, the radio signal strength (RF) between a necklace antenna and a wrist-worn antenna [181] was investigated. Nowadays, the Bluetooth link strength between Smartwatch and Smartphone could provide a similar signal.

Recently, acoustic approaches were presented. A necklace with a body-coupled (stethoscope), far-field microphone and loudspeaker is used to detect hand-to-mouth gestures via the Doppler effect. Deep inhalations via stethoscope [182]. These detectors are however only started when the flick of a lighter was detected via the far-field microphone in oder to conserve energy. Another study presents the results of only using a neck-worn microphone to detect breathing patterns [183].

More exotic sensor deployments make up the fourth group of modalities. Dust sensors and electrodermal activity on the wrist [184], and smart cigarette cases [176] are examples thereof. These rely on the way the devices are used to detect cigarette smoking. Device-free recognition is enabled via Wifi signals, when a smoker influences the signal propagation between multiple access points [34]. The latter does not require any worn device, but a deployed infrastructure.

Current clinical devices for assessing smoking status can not be used for cigarette-level tracking. These just provide a limited estimate of the number of cigarettes smoked, and information about the smoking status [185]. Included in these methods are self-reports [186], serum-based methods with test strips [187], and breath analyzers [188, 189,

Study	Sensors	Annotation	n / k / t	Env
mPuff [179]	RIP	self	4 / 8.2 / 11h	Field
puffMarker [180]	RIP/ECG	self	33 / 1 / ?	Field
RiSQ [29]	9DOF	self	4 / 7.5 / 4h	Field
HLSDA [31]	6DOF	?	11 / 21 / 17h	Field
Tang [32]	3DOF	shadow	6 / 5.6 / 2h	Field
Smokey [34]	WiFi	video	? / 277 / ?	Field
Cui [182]	audio	self	2 / 18 / 24h	Field
PACT [181]	RF/RIP	?	20 / 2 / 1h	Lab
Raiff [178]	6DOF	video	6 / 6 / 3.5h	Lab
Dementyev [184]	dust/EDA	self	12 / 1 / 4h	Lab
Qin [33]	GPS/6DOF	self	3 / ? / ?	Lab

Table 2.1: Studies on wearable smoking detection. Only those studies where data was collected are included.(RIP = respiratory inductance phletysmography, RF = radio frequency signal strength, EDA = electrodermal activity, ECG = electrocardiography). A limited amount of studies attempted longitudinal recordings for longer than two days [31, 180, 34]. The n/k/t column, refers to the amount of participants n, the average amount of instances k per participant and the total duration of data recording per participant t. Only [34] did not report on n hence the total number of instances is reported.

190]. Usually, these methods are just used as an additional efficacy measure for interventions. The Breathalyzer however was tested as an unbiased feedback during an intervention program [188] and has shown a promising effect. In contrast to non-serum methods, post-hoc validation of the smoking status is possible. A different concept, which reminds its user when its time to have another cigarette, is implemented in the QuitKey [191]. This is thought to teach its user about their addiction.

Only few systems were characterized beyond the actual recognition performance. One commonality is that almost all require the application of machine-learning for signal analysis. Study design and data collection is of upmost importance, as this is the foundation for the generalizability of the gathered results. Table 2.1 provides an overview of concluded studies. Despite the largely different number of participants, lab and in-field study setups, a drawback of most modalities is the detection delay. For any continuous sensor, like acceleration, a time

window at least half of a typical cigarette consumption is necessary to distinguish from con-founding activities like eating or drinking. Two studies [182, 29] explicitly considered the energy requirements of a longitudinal deployment of the sensor system.

Challenges particular to wearable smoking detection were posed in [182, 32, 178]. Objective smoking data collection in-field was mentioned as one of the major challenge by almost all authors. *System complexity*, as well as the *unreliability of self-reports* are the main issues. For detection from sensor signals, *con-founding* activities, like coughing, scratching one's nose, eating or drinking [36], and gesticulating can lead to false detections. But also capturing combinations of different *smoking styles* and *concurrent* activities, for example walking, riding a car or bike, sitting and lying are not easily managed. The ambiguity introduced by different sensor wearing styles is addressed in [35] - sensor data is transformed according to how it is worn. [192] introduces a hierarchical probabilistic model to encode knowledge about smoking and other hand-to-mouth related activities.

This thesis focuses on two questions that have not been answered conclusively yet: (a) Is energy-efficient *long-term* recognition of smoking instances via IMU-measured wrist motion possible? And (b) what recognition performance for such detections is to be expected under unconstrained real-life conditions? The first question is addressed by benchmarking different classification modalities on their energy efficiency. The second question by using a novel ground-truth collection method: Instead of solely relying on human observers or self-reports by the smoking participants, an instrumented lighter tracks smoking incidents. This allows the gathering of a data set, in which the participants becomes less aware of being tracked as well as allowing for a study setup where participants can move freely, following their usual daily routine without interruptions by the recording system.

2.6 WEARABLE SUPPORT IN THE WETLAB

The notebook, in which experimenters record their thoughts, results, and plans serves as one of the core parts during a life scientist's research. As such it has been the target of many research efforts, as well as targeted by commercial endeavours. Electronic laboratory notebooks (ELNs) are often sought to replace or enhance their pen and paper counterparts. However, even offering clear advantages, like *searchability* of records, *multimedia* integration, *enhanced collaboration* [193, 194, 195],

edit-ability [196] and *capturing of instrument measurements* [197, 198], their spread is still limited. This is partially due to the legal requirement of the stored record, which form the basis to trace and claim inventions, to check the conformance to established protocols, and to handle liability and legal issues. For electronic records, only a limited legal framework is in place [199, Taylor2006; 200], complicating possible usage scenarios. More profound though is that the 'manipulation of digital artifacts' might not be flexible enough for scientific record keeping [201]. Or as Kanza et. al. [202] put it:

> Paper notebooks are considered easier to use, input data to, read, transport, inexpensive, readily available, 'turn on' instantly, have infinite battery life, are socially acceptable during meetings, and require no training and minimal IT support.

Pen and paper solutions are also preferred because of the flexibility and freedom over visual structure they provide [203]. This has lead to several efforts which try to combine physical and electronic notes. The *a-book* [204] combines a tablet and PDA to capture paper notebook writing. Based on a fiducial marker, entries can be augmented with additional media and easily shared. A system to support biologists in the field was presented in the *ButterflyNet* [123] project. Handwritten notes are captured (with an Anoto pen) and combined with visual and audio information for later access. This allowed the biologist to capture information in the field, and augment it with other sensory clues - a task that previously had to be done manually. The *Prism* [193] project reports on a study of biologists' work practices and presents a hybrid system using hand-written notes as well as digital content to capture, visualise and interact with so called activity streams in the laboratory. Forcing too much structure has been found to be too inflexible. An open design based on linking and searching information bits was then adopted, similar to the MyLifeBits design [205] but specific to the experimenters workflow.

Other type of systems are focussed on providing a more formal specification of single experimental workflows, on which user interfaces are built. These systems also augment the lab itself to provide these user interfaces. The *LabScape* [206] project was an early investigation in a ubiquitous computing platform to help scientists and students to access and capture information in the laboratory. It uses interactive flowchart diagrams to visualize and annotate ongoing procedures

that are accessed via touch-tablets, barcode scanners, RFID tags [207], numeric keypads and wireless keyboards.

During the *Combechem* project the idea of a digitized flowchart was enhanced to the Semantic Smart Laboratory [208], a system for supporting chemistry experiments focusing on providing a flexible ontology for describing experiments and storing them for later retrieval. A formal definition of chemical experiments is presented as part of the *Labtrove* project [209] as well. The *eLabBench* [210] and *Biotisch* [211] take the integration in the laboratory further by replacing the traditional workbench with a tabletop system that presents information on the bench's surface, also allowing interaction, sensing of augmented objects (e.g. racks of test tubes) and taking pictures of the whole setup with an overhead camera. The gathered digital information is stored in a wiki-like notebook for later retrieval.

In contrast, the approach proposed in this thesis, focuses on the largely unexplored area of supporting and augmenting laboratory tasks by means of a lightweight, exclusively wearable system. The setup requires little to no interference with the laboratory environment and its inventory, and offers hands-free operation. We argue that this approach of augmenting the *researchers* instead of the laboratory, has many advantages, not in the least the fact that every user in existing laboratories can still opt to keep on documenting their experiments by traditional methods.

As has been shown by some of the above research, a formal workflow tends to be valued by experimenters and can be exploited as a structure for information capture as well [206]. We investigate in particular whether a wrist-worn accelerometer unit can be used to capture such pre-defined structure, for instance to index associated video and audio recordings. In the case of the life science laboratory, these workflows tend to be frequently predefined, and actions extracted from available textual descriptions, facilitating a semi-supervised approach. Such monitored workflows enables easy documentation access by jumping to the currently required information, and can also assist in recording. Other work domains were investigated in the Wear@It Work project [68], for maintenance [212], manufacturing [81], inspection tasks [213] as well as the use of HMDs in wetlabs [214].

Workflow monitoring can be achieved with body-worn motion sensors, but also through the continuous detection of object use. In the microbiology lab this mostly involves samples of living organism and other compounds. The facility in which such samples are stored are

called Biobanks or Bio-repositories - these are charged with the preservation of patient (and other) samples, their documentation, retrieval and safe storage. Best practises for Bio-repositories workflows [215] include the barcoding of samples with a unique identifier, together with human-readable information. Electronic records can be connected via this identifier, and location tracking implemented via barcode scanning. A centralized database even allows for keeping track of shipment logs and cross-institutional information sharing. In contrast to such fiducial markers, RFID technology provides *non-line-of-sight reading*, *read-write* support, *fast parallel reading* capability, and the potential for location, temperature and motion sensing [216]. The latter three are important since they allow to mitigate common errors [217], like tracking of transportation failures, avoidance of unnecessary heating during identification, and to a certain extent the mis-labelling of samples. Location tracking is of special interest since a discrepancy to the electronic record can be automatically detected if samples are stored next to a reading unit. Smart tubes [218], RFID labels in repositories [219], and freezable tags [220] were reported. However, all reports deployed a fixed station to interact with the inventory system.

A wrist-worn RFID reading unit could remedy the disadvantages of a fixed station. Based on the ability of RFID for remote, non-contact identification of objects, specific tasks, e.g. using a hammer, and their accompanying activities can be derived in a reliable fashion [221]. However, antenna design is a major challenge: a trade-off between size, flexibility, robustness and its wearability has to be found. The placement on the human hand mainly dictates the possible choices. In first iterations the reader was placed on the back of the hand [222, 174, 221] which allowed for reading distances of $1-2$cm. Antennas looped around the wrist [223, 222, 224, 221] have replaced this design. However loop antennas need to be rigidified to keep their performance controllable, a $10-15$cm reading range with a common 5cm-patch RFID-tag has been reported. A flexible antenna placed between thumb and index finger [81] is challenged by sweat and by changing (antenna) parameters due to movement. While placing the antenna on the thumb achieves the best reading performance, especially for miniature tags, its attachment point also hinders the movement of the wearer's hand. In this work, a flexible antenna worn in the palm is compared to a rigid loop antenna worn around the wrist.

Simplifying and integrating the identification and labelling of samples has been argued for in other research as well. Boriello et. al. [225,

207, 206] do not only argue for tracking samples, but also the tools used in a micro-biology wet laboratory. This information can be used post-experiment for reconstruction purposes, or during conduction for error checking. Fiducial markers were registered with a camera below a tabletop by Tabard et. al. [210] as an alternative. This allowed for rack-based identification of samples. For tube-level identification RFID tags were added to all containers, a reader integrated into the rack and communicated to the system via an active fiducial marker. Both setups allowed for labelling and identification of multiple tubes in parallel.

Activity recognition from objects instrumented with RFID markers, and from wearables instrumented with motion sensors would then allow to index video, audio and other recordings to augment the life scientist's memory. This in line with research from a life-logging perspective with the goal of improving human memory [226, 100, 227]. To move "beyond total capture" [99], the retrieval and creation of useful cues is of importance, which is also in line with Vannervar Bush's 1945 vision of the MEMory EXtender (MEMEX) device [1]. In this work, we assume activities and used objects are useful cues.

2.7 CONCLUSION

This chapter presented the current approaches to detecting activities from body-worn sensors. Recording of sensor data, and particularly gathering ground truth data presents the first challenge in naturalistic settings when exploring novel applications. For explored applications publicly available datasets often exists. However, in both cases a data format that encompasses multi-modal, and multi-rate sensor, ground truth, and secondary evidence data remains challenging.

The second challenge is the parametrization of the Activity Recognition Chain (ARC), which requires careful cross-validation. Often large parameter spaces need to be searched, which requires software frameworks that can be run on a large number of cores in parallel, and a framework that is flexible enough to quickly test new processing ideas. Despite current deep learning approaches, which lock a developer into a particular framework, an alternative approach is proposed in the next chapter. This approach provides more flexibility, while retaining most benefits of a deep learning approach.

The presented framework is tested on two applications, of which related work was reviewed: detecting smoking from wrist motion, and detecting microbiology process steps from body-worn sensors. For

smoking recognition a multitude of approaches based on different sensors exists nowadays that could be combined into an ensemble of detection methods. For detecting process steps it is clear that only an ensemble of sensors will provide practical recognition, which is further challenged by the necessity to formally capture the information required during such experiments.

UNIX FILTERS FOR ACTIVITY RECOGNITION

Activity recognition from (body-worn) sensor data involves the application of machine learning. Hence, a framework or library is used for any type of detection. Choosing a model for a particular task always requires experimentation of different possibilities, and hyper-parameter tuning according to the problem at hand. The processing of sensor data is conceptualized as a sequence of steps that modify this data, where the input is raw sensor data, and classifications are the final output.

In this chapter an approach based on the classic *Unix filter* architecture [228] is proposed. In this architecture processing steps are encapsulated in Unix processes, and limited to a single specific task, following the Unix philosophy to "do one thing, and do it well". Each task accepts at least one standard input, and provides at least one standard output, through which communication with other tasks is enabled. These processes are then combined with pipes which connects these in- and outputs to form more complex processing chains. The specific processes required to build Activity Recognition Chains (ARCs) are accordingly mapped to single Unix processes.

An architecture based on Unix processes exhibits features, which are hard to achieve with typical library-based architectures. Unix processes can be built with any programming language an author might prefer, or a project demands, only the input and output data format needs to be adhered to. This simplifies testing new ideas and processing steps. Due to making the exchange of data explicit through the piping mechanism, process-level concurrency is facilitated since these processes can be run in parallel, and are activated only when new data from a previous step is available. Furthermore, the transport of data is transparent to the developer of each process. Data can originate from a network connection, as well as a local file, rendering integration into new computational environments easier. Processes are also encapsulated, a bug in one of the processes does not (directly) affect the rest of the processing chain. This renders the debugging of potential problems easier.

Specifically for Activity Recognition, which typically involves the recording of multi-modal datasets, a number of data recording and learning frameworks were proposed. One of the major challenges is the distributed recording, and curation, of multiple sensors involving video, audio, motion and other data sources. The Context Recognition Network (CRN) toolbox [229, 119] proposes the use of the Apache CouchDB for storage, and provides pattern recognition and labeling tools written in Java. This framework was also extended for recording

sensor from the Android OS [230]. The Gesture and Activity Recognition Toolkit (GART) [231] is a further example of a toolkit written specifically for Activity Recognition. It provides high-level abstraction for (live) data collection from sensors, and machine learning models in the Java programming language. All tools have in common that they are limited to their particular programming environments, and while this allows for modification, integration with other environments becomes harder. In the approach presented here, the only constraint is adhering to a specific data exchange format.

With such an architecture in place, more attention needs to be placed on the exchange format between processes. Traditionally this is a text-based format. However this incurs a large processing overhead, and is quantified in this chapter. Since data between each process is copied to improve encapsulation, this overhead is multiplied for each step in the processing chain. A binary format is more complicated to implement, and requires a description of the data, while a text-format is often self-descriptive. Both approaches are compared in this chapter.

Process-based parallelization is easily achieved with this architecture. This allows to make transparent use of multi-core machines, and cluster of Linux machines. Since data is explicitly copied between processes, exchanging this data via local memory or through a network connection is transparent when executing the chain. This way parallelization can be easily achieved on process-level.

Modifying and adapting chains to novel problems is mostly achieved due to a standardized input format, which also encodes meta-data about the data itself. This involves the actual sensor modality for a given sensor data stream, but also recording parameters that might change for different datasets. Due to this standardization and the transport transparency which enables parallelization, deploying trained machine learning systems is facilitated as well. The input of a machine learning system can be switched from a dataset loaded from disk, to a network stream which contains live-data recordings for example.

In this chapter the architecture of the proposal will be described, specifically how common processing steps are splits into several independent Unix utilities. Before that an Android-based recording tool, which (optionally) compresses and synchronizes multiple sensor streams is described. This tool also directly stores data in a multi-media container, which provides a binary self-describing exchange format. Particularly the issue of identifying sensor modalities, in the case this meta-data is lost, from sensor data alone is investigated. Afterwards pro-

cess steps are described, and the binary exchange format is compared to a character separated values (CSV) format according to the incurred performance overhead. Finally, scalability is shown by investigating the use of this framework on a multi-core, Linux cluster.

3.1 DISTRIBUTED RECORDING AND CURATION

The first component which follows the Unix filter design is the actual recording of sensor data from body-worn systems. Mobile devices running the Android Operating System (OS) are pervasive nowadays, and are able to run Linux software. This is the target of the presented distributed sensor data recording component presented in this chapter. Android already provides facilities to recording device-local sensors. Remote sensors, which do not run Android, can be integrated via a network connection through Bluetooth or WiFi. Cabled USB connection present another option for integrating additional self-build sensors. Despite Android devices, the recording component can be used on most embedded devices that run the Linux kernel.

Besides exchanging meta-data like sample formats, and sensor settings, synchronizing such a network of sensors becomes the major challenge. Each sensor runs with a different clock that might be offset in comparison to other clocks, or run at a slightly different speed, which causes so called clock drift. While clock synchronization is required to start recordings on multiple machines at the same time, clock drift is harder to fix. One way though is to assume constant sampling rates from each sensors and synchronize all sensor streams on a global clock by either marking missing samples or removing redundant samples. Another alternative is to store a timestamp on a device-local, and global clock with each sample, and meet these challenges during analysis of the sensor data. This however, unnecessarily complicates the analysis.

Figure 3.1 shows a conceptual overview of recording component. Components depicted there represent Linux processes, that are connected through pipes. Sensor processes read sensor data with specified parameters, most notably the sampling rate, and print each sample on its standard output. Multiple of such outputs need to be merged on a common timeline and stored in time-related blocks. This is also called *multiplexing*. Blocks of sensor data, which might be recorded at different rates are interleaved to optimize for time-based access in memory (cf. Figure 3.1). Here, we rely on the FFmpeg software suite to multiplex multiple sensor data streams into a single output file for

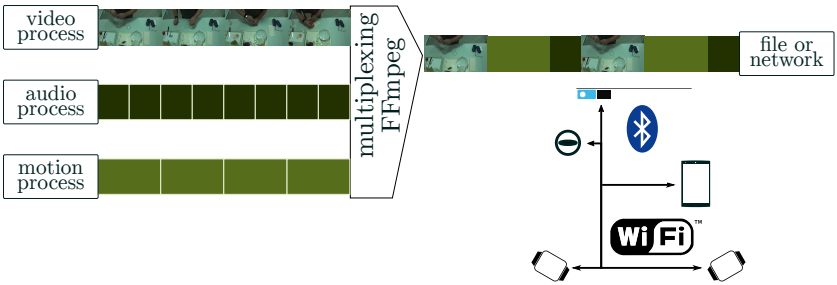


Figure 3.1: Architecture of the distributed recording infrastructure, depicting sensor recording and multiplexing processes. Also depicted is a possible device setup consisting of a single Smartphone, two Smartwatches, a Ricoh Theta S camera and Google Glass, all connected via Bluetooth and WiFi.

storage. This output file is in the Matroska [232] file format, which can contain multiple video, audio and subtitle streams. Sensor data is stored (and compressed) as an audio stream, and these files can also be streamed to other machines via network connections. Recording parameters are stored in the file as multi-media tags.

With this recording infrastructure, new sensor sources can be easily integrated and tested independently of the actual recording. Only a new Linux process that generates sensor data at a constant rate, either locally or from a remote source, is required. This process then outputs data on its standard output, which itself can be transported via a network connection if required. FFMpeg than finally multiplexes multiple sensor sources into a single file. This also provides a compressed, standardized format for video, audio, motion and subtitles which can be opened with already existing software. Hence, alleviating the need to write software specifically for reading and writing a newly recorded datasets. The Matroska multi-media container can also serve for curating datasets for longterm storage. Due to the optional compression storage space is saved, and the format also allows for live-streaming. Such live-streaming can be with facilities already integrated in the FFmpeg software suite, which is in use for a large portion of internet live streaming solutions.

The recording infrastructure is called CMotion and allows to select recording parameters, and starts the recording on multiple devices after clock synchronization. Data is stored locally on each device in a

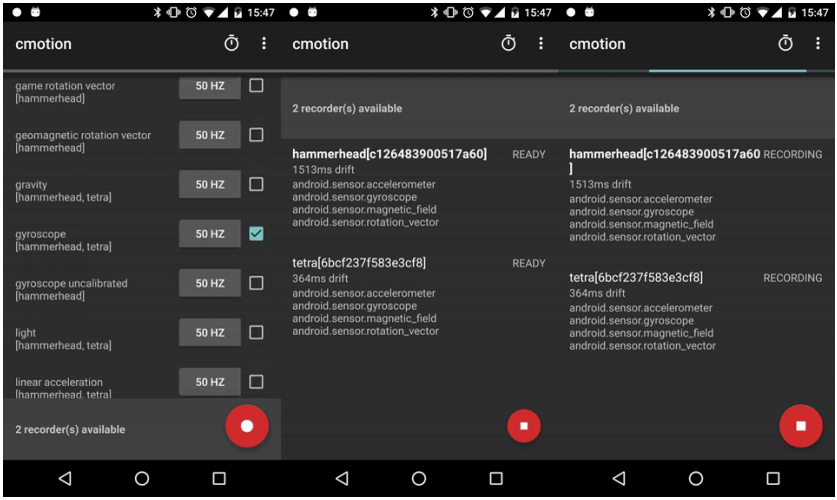


Figure 3.2: Screenshots of the Android application which controls a network of recording devices. The sensors that is to be recorded on each device is chosen, afterwards clocks are synchronized and the recording started on each device. The recording status is shown on each device

multi-media file. Android devices store data and recording parameters in a Matroska-file, which contains FFmpeg-multiplexed sensor data streams. The user interface of the Android application which controls the recording is shown in Figure 3.2. Devices like the WiFi-enabled Ricoh Theta S 360° camera store video files in a different video container, which can be converted and merged post-recording. All recorded data is aggregated offline on a PC post-recording. Streaming data live via the provided network connections is possible with this setup (for example via the UDP, or RTMP protocol that is provided through FFmpeg) but was not attempted due to higher energy requirements and possible data loss when not additionally storing data locally.

3.2 INERTIAL MOTION DATA COMPRESSION

At the heart of each Activity Recognition task is a dataset. This dataset might be formed from multiple media streams, like video, audio, motion and other sensor data. Recorded at different rates, sparsely or uniformly sampled, changing units and with different numerical ranges,

these streams are challenging to process and store. These parameters are usually documented in an additional file that resides next to the data [61, 233, 234, 235, 236]. The actual data is commonly stored in a CSV file, in a binary format for Matlab or NumPy, or in Machine Learning frameworks specific ones like ARFF [237] or libSVM [238]. For small, independent time-series this is a worthwhile approach, mostly due to its simplicity and universality. However, parsing CSV files incurs a large performance and storage overhead, compared to a binary format.

When observing with multiple independent sensors, synchronization quickly becomes a challenge [61, 143, 239]. Different rate recordings have to be resampled, time-coded files have to be merged. This issue is often hidden until the dataset is going to be used. Possible approaches range from offline recording with post-hoc synchronization on a global clock, to live streaming with a minimum delay assumption - for which all but the last one require some form of clock synchronization and careful preparation. Storing events with timestamps on a global clock is then one possible way to allow for post-recording synchronization, i.e. each event is stored as a tuple of <timestamp, event data>.

The subsequent step of merging such time-coded streams often requires to adapt their respective rates. Imagine, for example, a concurrent recording of GPS at 3Hz and acceleration at 100Hz. To merge both streams: will GPS be upsampled or acceleration downsampled, or both resampled to a common rate? Which strategy is used for this interpolation, is data simply repeated or can we assume some kind of dependency between samples? How is jitter and missing data handled? These questions need to be answered whenever *time-coded* sensor data is used. A file format which makes the choice of possible solutions explicit is the Matroska multi-media container.

The following Activity Recognition Datasets were published over the last few years and selected as examples of data encodings:

HASC Challenge [233] 540 subjects, time-coded CSV files. Mostly activities of daily living without secondary evidence like video recordings.

Opportunity [61] 12 subjects were recorded with 72 on- and off-body sensors in an Activities of Daily Living (ADL) setting. Multiple video cameras were used for post-hoc annotations. Data is published in synchronized, time-coded CSV files.

Freiburg Longitudinal [240] one sensor, one subject, four weeks of continuous recording. Data is stored in numpy's native format.

Sensor data is not different from low-rate audio. Common parameters are shared, and one-dimensional sensor data can be encoded with a lossless audio codec for compression. Rate, sample format and number of channels need to be specified for an audio track. The number of channels is equivalent to the number of axis an inertial sensor provides, as well as its sample rate. The sample format, i.e. how many bits are used to encode one measurement, is also required for such a sensor. Other typical parameters, like the range settings or conversion factor to SI units (if not encoded as such), can be stored as additional meta-data, as those are usually not required for an audio track.

Lossless compression, like FLAC [241] or WavPack [242], can be applied to such encoded data streams. This allows to trade additional processing for efficient storage. Several lossless schemes are evaluated. These include the general LZMA2 and ZIP compressors, and the FLAC [241] and WavPack [242] audio compressors. All but the first two can be easily included in multi-media container formats. To use audio streams, data needs to be sampled at a constant rate, i.e. the time between two consecutive samples is constant and only jitter smaller than this span is allowed. Put differently, the time between two consecutive data samples t_i and t_{i+1} at frame i must always be roughly equivalent to the sampling rate:

$$\forall i \in N : t_{i+1} - t_i = \frac{1}{r} - \epsilon \tag{3.1}$$

Compared to time-coded storage, the recording system has be designed to satisfy this constraint. Problems with a falsely assumed constant rate recording setup will therefore surface faster. Especially in distributed recording settings, where above mentioned constraints is checked only against a local clock which drifts from a global clock.

Sparsely sampled events can be encoded as subtitles. Here, each sample is recorded independently of its preceding event, i.e. the above mentioned constraint does not hold. Each event needs to be stored with a time-code and the actual event data. Depending on the chosen format, this can also include a position in the frame of an adjacent video stream or other information. For example to annotate objects in a video stream. A popular format is the Substation Alpha Subtitle (SSA [243]) encoding, which includes the just mentioned features. Since data is encoded as strings, it is suitable for encoding ground truth labels. To a limited extent, since no compression is available, it can be used for sensor events as well. For example, low rate binary sensors, like RFID readers can be encoded as a subtitle.

Encoded sensor and subtitle data can then be combined with audio and video streams in a multi-media container format. One such standard is the Matroska [232] format, that is also available in a downgraded version called WebM [244] for webbrowsers. Once the data streams are combined into one such file, this data can be "played" back in a synchronous manner. This means that streams recorded at different rates, and in different formats, need to be converted to a common rate and possibly common format. Meta-data that contains additional information like recording settings, descriptions and identifiers can be stored in addition to the parameters already contained in the stream encoding. For this task off-the-shelf software, like FFMpeg [245] can be used, which also provides functionality like compression, resampling, format conversion and filtering. Annotation tasks can be executing with standard subtitle editing software, discouraging the creation of yet another annotation tool. Furthermore, video streaming servers can be used for transporting live sensor data recordings to remote places.

The use of such a standard format for curating datasets allows for re-using existing software, however not without limitations. Asynchronous, also called sparsely sampled, data recorded at high rates is not supported. This mainly stems from the simplifying assumption that streams are recorded with a constant rate. Satisfying this constraint while recording might be easier than handling asynchronicity later on. For example, breaks, shifts or jitter due to firmware bugs can be detected earlier. Another shortcoming is that structured data can not be stored transparently, each event is assumed to consist of one data type only, e.g. multiple channels of 8-bit integers in contrast to a mix of data types. In general this is hard limitation, however different data types can also be encoded in multiple streams. Also, the en- and decoding overhead might be a limitation, which we will look at next.

Compressing sensor data as an audio stream incurs an en- and decoding overhead, while providing optimized storage. By a repetitive measurement of the relative wall clock time for decompression, its overhead is measured. The compression factor is determined by comparing the number of bytes required to store the compressed file to the original, deflated CSV file. Binary and text-based storage is compared. The Zip and LZMA2 algorithms are used for general byte-wise compression, and the lossless FLAC and WavPack compressor for audio-based compression. LZMA2, since it performs better than ZIP, is tested on text and binary files. The approach of compressing binary files with a general compressor is used by Numpy for example.

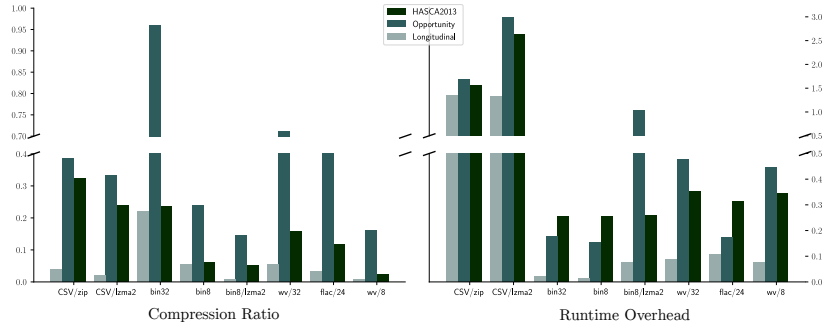


Figure 3.3: Fraction of storage required for three datasets compared to uncompressed CSV files. Zip and LZMA2 text compression, 32-bit binary, WavPack [242] with 32/8-bits and 24-bit FLAC [241] audio encodings are shown. On the right hand side the relative runtime overhead for decoding each format is visible. Shown is the fraction of wall time required to decode the respective scheme relative to the time required to parse a CSV file into memory. The result for each scheme is the (binary) data stored in memory.

The fraction of required storage after compression is given relative to the deflated, original CSV file. For the runtime overhead, the fraction of reading time relative to reading and *converting* the CSV file into a memory image is reported. The test were run on the Opportunity [61], HASC Challenge [233] and on twenty days of the Freiburg Longitudinal Wrist [240] datasets. A machine with an i7-4600U CPU running at 2.1GHz with 8GB of memory was used for all tests. Fig. 3.3 shows the results of these tests, *CSV/zip* refers to a zip-compressed CSV file, *CSV/lzma2* to an LZMA2 compressed file[1], _bin*_ refers to signed integers with the respective bit length optionally compressed with LZMA2, _wv*_ to WavPack compression of varying bit size per value and *FLAC* compressor which only supports 24bits values.

The least processing overhead is incurred by a binary format, at best a memory image which can be *memory mapped* into main memory. This is the format that will be decoded to in the following. The baseline is therefore the time required to convert a CSV from disk into a binary format in memory. The fraction of time required to do the same for each compression scheme is reported in Figure 3.3. Each test is repeated

[1] the XZ utils package was used

six times, and the first run is discarded, i.e. data is always read from the hot disk cache.

Just parsing a CSV file incurs an up to hundred-fold overhead (bin8 in Figure 3.3) compared to reading a binary file. Compressing CSV data can increase the runtime by $1.4 - 3.0$ times. So, looking only at runtime performance a CSV file should hardly be used for large datasets. When comparing compression schemes, it can be seen that WavPack provides the most consistent performance measures over all datasets. It is not slower than the more general LZMA2 compressor, and the FLAC compressor is only faster on two datasets. However, for the decoding task at hand here, an overhead of at least two times is incurred compared to raw binary storage.

The datasets show different characteristics found in other datasets as well. For example the Longitudinal [240] dataset can be massively compressed with text-based algorithm, almost down to 2% of its original size. This is mainly owed to the fact that the contained acceleration data was recorded with a resolution of only 8-bits, and that a run-length compression was already applied during recording. This run-length compression is deflated for CSV storage first, adding a lot of redundancy. For the same reason, storing data non-compressed in 32-bit binary format is actually larger than the zip-compressed text-format. However, encoding with the original 8-bit resolution in the WavPack compression leads to a slightly better storage efficiency.

The same effect is visible for the Opportunity [61] datasets, where feature vectors are stored instead of raw data. Storing in 32-bit binary increases the size again, which means that the average string-length for representing a number in this dataset requires little more than 5byte. Only when limiting the number format to 8bits a stronger compression can be achieved with an audio codec. The maximum dynamic range that can stored with a text-based format is however limited to the (decimal) encoding, (less than 10000 for five digits), while a comparable binary encoding can range up to 2^{5*8}.

The HASCA dataset [233] does not show this effect. Mainly because the CSV data contains floats with at least ten digits. These could be stored with 32 or 64bit, which would be more efficient than their text counterpart. Especially since values are stored with more than eight digits per value.

When optimizing data storage for space efficiency, the encoding of each value is the most critical factor. Limiting the number of bits per value, in essence assuming a limited dynamic range of the encoded

signal, has the strongest influence on the storage efficiency. However, when encoding values in text format and a dynamic range that is limited to four characters is enough, a text compression algorithm is not worse than encoding data in binary format. For the general case and when binary storage can be used, the WavPack compression provides the same storage efficiency as the more general LZMA2 compressor.

Compared to the de-facto standard of using CSV files, encoding sensor data as audio, annotations as subtitle and combining both with video- and audio-based provides several improvements. Important parameters like sampling rate, format and number of axes is included in the file. Adding additional information as meta-data leads to a *self-descriptive* format. *Synchronous* playback of multiple streams, which requires re-sampling, is supported by off-the-shelf software. Related problems, like un-synchronized streams can be caught earlier, since this step is explicit. The container format is *flexible* enough to support different number formats, i.e. values can be encoded as floats or integers of varying bit-size. Optional compression leads to *compact* storage, which allows for efficient storage and transmission. Additionally, when thinking about large datasets, such a container format requires *divisible* storage. This functionality (seeking without reading the whole dataset into memory *which would be required for time-coded storage*) is provided.

3.3 INERTIAL SENSOR MODALITY IDENTIFICATION

To correctly interpret sensor data, reliable information about the data itself is required: sample (and frame) format, recording rate, number of axis, position at the observed body and sensor modality need to be known. This *meta-information* is often stored along-side the data itself, either in a (semi-)structured external file, as a header of the data or as a well-known convention. Misinterpretation, resulting from missing or incorrect meta-data, has a strong influence on a subsequent application's performance. If such meta-data is not in a machine-interpretable format, slow and cumbersome recovery by a human expert becomes necessary. Here, the extent to which the *sensor modality* can be recovered from invariant statistical properties of the sensor data itself is investigated. Assuming that sample format, number of axis, and sample rate are known beforehand, but scale, and other calibration factors are not, we show that a sensor's modality can be (automatically) verified, or identified.

Providing structured meta-data enables datasets to be picked up

by search engines [246]. In the absence or partial availability of this data, an automatic identification of sensor modality provides a basic starting point that otherwise would require manual inspection by an expert. Webcrawlers could verify that meta-data was *correctly* specified. Opportunistic sensing [247], i.e. situations where the sensor type is not known beforehand, would be another application area. While proper data curation practises could alleviate these situations, and are arguably more straightforward, an error in such manually defined data is often found much later. Manually reconstructing meta-data is then hard to scale to large data collections, as inspection by a human expert is required. A second system, which identifies modality from data directly, could at least provide additional safety checks. Such kind of quality control allows to (automatically) check if datasets were correctly documented or if there might be errors in the data collection itself, for example when uploading into a public dataset repository.

Activity Recognition applications are build with assumptions about the data retrieved from wearable inertial sensors. Properties, like placement variations or body locations are assumed, even though they directly influence the recognition performance. Kunze et.al. [248] have shown such influence, and provide several techniques to mitigate those effects. Namely, using location independent features, adding location to the classification task or estimating location from long-running recordings [248]. Similarly to the last option, we look at the properties of different sensor modalities over longer time periods to extract the sensor modality. Whether sensor data arose from an accelerometer, gyroscope or a magnetometer is usually stored as non-standardized meta-data, but also strongly influences the recognition performance if incorrect. Hammerla et.al. [249] introduced the empirical cumulative distribution function (ECDF) as a mean to capture the statistical properties of acceleration data, while also serving as a feature reduction method. Inspired by this, *invariant* statistical features that capture the properties of the inertial sensor modality are investigated. A system to correlate known datastreams to unknown ones and subsequently propagate their meta-data is described in [250]. In contrast this proposal does not require prior knowledge in the form of known sensor data, i.e. a ruleset that can be readily applied is proposed. However, this proposal is limited to inertial sensor data.

Community provided datasets which included all three inertial sensor modalities, optionally mounted at different body positions were selected. These were converted into the common Matroska data format

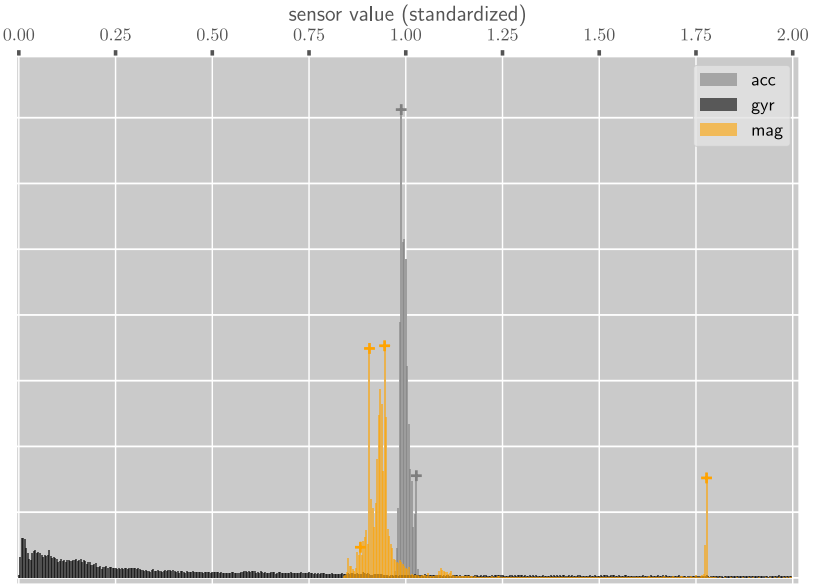


Figure 3.4: Example histogram of three inertial data distributions of the CMU Kitchen dataset. The concentration of the gyroscope data around zero, as well as the concentration of the acceleration data around its mean, and the larger number of modes for magnetometer data is clearly visible. Identified modes on the distribution are highlighted.

to simplify their usage:

CMU Kitchen [251] contains inertial data of multiple body locations, including arms, legs and the back. Even though two inertial capture systems were recorded, only the wireless one, recording at 125Hz, was used. To balance with the other datasets, only a subset of 3h and 26 participants was extracted.

ICS Forth ADL [252] contains motion data from the thigh, ankle, torso and wrists. The measurement was taken at 50Hz. 15 participants were recorded for a total of 4.5h executing activities of daily living.

Pamap2 [253] contains motion data of 8 participants executing activities of daily living at the hand, chest and ankles. In total 8h were recorded. Inertial data was recorded at 100Hz. Two acceleration (at different scales) streams, one magnetometer stream and a gyroscope stream

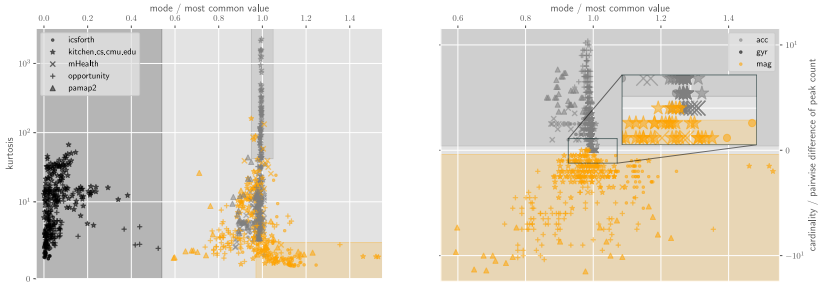


Figure 3.5: Scatter plots of two possible feature sets for sensor modality detection. One feature is the mode of the histogram (512 equal-sized bins), i.e. the most common value. The second feature is either the kurtosis of the data, or the difference between the mean number of modes at the same limb and number of modes of one sensor stream. The left hand side shows that not all cases can be identified with mode and kurtosis only. The mode count difference provides a better indication, with the necessity to assume that both a magnetometer and accelerometer stream is present. Decision thresholds are shown as highlighted layers.

were used.

Opportunity [254] contains a whole-body inertial motion recording of daily living activities. A subset of 4 participants (with video recordings) contributed 15 data points each. In total 8h of data recorded at 30Hz were investigated.

mHealthDroid [255] recorded 12 activities of daily living. Shimmer nodes sampled at 100Hz provided the inertial data used in this paper. In total 6.5h were analyzed.

Each sensor modality differs in various aspects, which requires a few transformations prior to identification. In order to simplify the overall analysis, only the *magnitude* of sensor readings is used instead of its vector form. This is achieved by applying the L2-norm to each sensor reading, which also renders all subsequent calculations *orientation-independent*. Since sensor streams might be scaled differently, e.g. the two acceleration streams in the Pamap2 dataset were recorded with 6g and 16g range, standardization is required. Dividing by the mean, i.e. *standardizing the scale* allows recordings to be compared. Additionally data was *lowpass-filtered* with a cutoff frequency of 2Hz. With the final

assumption that the sensor is *most commonly at rest* on the body, we can now look at the properties of the transformed sensor streams, which we denote as d.

3.3.1 Single Sensors

When the human body is at rest, no rotation is measurable. The rate-of-turn of a limb, measured by the gyroscope, is therefore most commonly near zero. This fact can be used to identify this sensor by the following rule:

$$mode(d) \approx 0 \Leftrightarrow gyr \tag{3.2}$$

Expressed differently, if the most common magnitude (mode) of sensor data is near zero, the sensor is a gyroscope and vice versa. This, however, only holds if the gyroscope data was baseline corrected, i.e. since the zero level of gyroscope is not at zero per default, this DC-offset is usually removed by calibration.

Due to being at rest, the accelerometer's (statistical) mode of magnitude corresponds to the strength of earth's gravitational field. Designating the field strength with $g = 9.81\mathrm{m\,s^{-1}}$, we can formulate $mode(d_{acc}) \approx a * g$, where $mode(d_{acc})$ is the most commonly measured value, and a an unknown scale factor applied to the data. If a would be known, accelerometer data could be readily identified by comparisons to earth's gravitation. However, since data was lowpass filtered, the mean magnitude of acceleration corresponds to g as well, i.e. $\tilde{d}_{acc} \approx a * g$. Due to standardization, we can formulate a rule for acceleration:

$$acc \Rightarrow mode(d) \approx 1 \tag{3.3}$$

Applying this rule to the scatter depicted in Figure 3.5 reveals why this is only a necessary condition; magnetometer data also fulfills this condition. A sufficient condition can be formulated for a subset of the overall accelerometer data, when including the kurtosis:

$$acc \Leftrightarrow mode(a) \approx 1 \text{ and } Kurt(d) > \alpha \tag{3.4}$$

Standardization is crucial for this condition, and relies on the assumption that the sensor is constantly accelerated by earth's gravitation. Other accelerations, due to limb movement for example, are only transient. Datasets which mostly contain strong movements, e.g. running or

stirring as exemplary activities from the analysed data, will likely break this assumption. This is however tested with the mHealth, parts of the Opportunity and the Pamap2 dataset, which all contain sequences of strong, continuous motion.

Figure 3.5 shows that some magnetometer readings exhibit a mode and kurtosis that is indistinguishable from accelerometer data. However, fluctuations in the measured magnetic field are more distinct than fluctuations of the gravity field. The respective distribution therefore is not uni- but multi-modal. This means there are multiple peaks, while the accelerometer distribution is rather "smooth" (cf. Figure 3.4). A mode larger than the mean (or 1 in the standardized dataset), and a smaller kurtosis can indicate this:

$$mag \Leftarrow Kurt(d) < \beta \text{ and } mode(d) \geq \gamma \tag{3.5}$$

Whether such strong fluctuations are contained in the dataset depends on the experiment's condition. By proper choice of β a subset of magnetometer data can be sufficiently identified. The smaller kurtosis can be explained due to the fact that magnetometer is often further spread out, and does not exhibit a strong concentration point. In contrast, acceleration data has a strong concentration and its kurtosis is higher.

3.3.2 Accelerometer vs. Magnetometer

The question remains whether sensor streams, which fulfil none of the necessary conditions (3.4) nor (3.5) can still be identified. More directly, when it is not possible to decide between acceleration or magnetic flux based on kurtosis and mode alone. One observation that can be made about these cases, as well as the already identifiable cases, is that the number of modes for magnetometer is larger than the ones for acceleration data. Estimation of number of modes can be achieved by adequately parameterized *peak detection* on the histogram. For a given stream, we designate the number of modes with p, as a shorthand for the *number of peaks*. However, streams have to be compared pair-wise, i.e. magnetometer and accelerometer must have observed the same motion. Let $\tilde{p}$ designate the mean number of modes of correlated sensor streams, then we can formulate the following condition:

$$d \Leftrightarrow \begin{cases} acc, & \text{if } \tilde{p} - p > .5 \\ mag, & \text{if } \tilde{p} - p < -.5 \\ unknown, & \text{otherwise} \end{cases} \tag{3.6}$$

Combined with condition (3.2) this allows to identify all sensor modalities, iff a correlated magnetometer and acceleration stream is to be distinguished.

3.3.3 Identification Ruleset

When only employing the sufficient conditions (3.4) and (3.5), we call this the *partial* ruleset. This allows to partially identify sensor modalities *without* assuming that both an acceleration and magnetometer measurement is included. If this pair-wise condition (3.6) can be assumed, we can formulate a *full* ruleset:

$$d \Leftrightarrow \begin{cases} gyr, & \text{if } m < .5 \\ acc, & \text{else if } \tilde{p} - p > .5 \\ mag, & \text{else if } \tilde{p} - p < -.5 \\ unknown, & \text{otherwise} \end{cases} \tag{3.7}$$

where $m = mode(d)$ designates the mode of the data, p the total number of modes and $\tilde{p}$ the mean number of peaks of correlated data streams contained in one dataset.

Prior to applying above conditions the data needs to be lowpass filtered, to exclude all frequencies above 2Hz. To reduce scaling effects, a standardization, by dividing by the mean of each stream is to be applied as well. The mode is determined from a histogram of 512 equal-sized bins, ranging from 0-2. Peak detection parameters were set to a minimum peak height of $.01 * m$, minimum distance of 5 bins and a minimum neighbor difference of $.008 * m$. These constants, as well as the decision thresholds in (3.7) were empirically determined.

3.3.4 Results and Limitations

In total 1003 streams with durations ranging from 7min to 1h were analyzed. All three inertial sensor modalities are included, mostly positioned at the lower arm (61%), the upper body (20%) and the legs (19%). Data is scaled differently for each included dataset, showing that

the proposed ruleset is independent of particular scale. Similarly, the sampling rates for each dataset differ. The *full* ruleset allows to identify 98% of all cases, while 2% remain for manual inspection. If streams can not be compared pair-wise, the *partial* ruleset can still identify 51% of all cases, of those less than 1% are wrongly classified, while the remaining require manual inspection.

One could argue that, since threshold and features were designed from, and tested on the same set of data points, the proposed ruleset will not generalize to unseen streams and datasets, i.e. do we observe an over-fitted solution to this classification task? This could be answered by maximizing the classification score by a search of parameters (lowpass cutoff frequency, peak detection parameters, thresholds of (3.7) ...) on leave-one-dataset-out splits. In the worst case, there is no choice of parameters that performs equally well across all splits, i.e. there is no generalizing set of parameters - best case, a *single* set of parameters which performs well across all splits is found. However, Figure 3.5 shows that even when leaving out one datasets from training, points from another set lie next to the decision boundary. However, not all parameters are chosen based on these data points alone (in contrast to what a machine learning approach would do): (1) the mode threshold is based on the insight that gyroscope data is concentrated near zero, (2) the pair-wise peak threshold follows the observation that the magnetometer distribution exhibits more modes. This is the case for 98% of the observed data points. The latter observation has examples in multiple datasets, as is visible in Figure 3.5, ruling out an over-fit. A cross-validated *automatic* choice of parameters would reveal if the opposite was true, in a formal way. Here we merely report a single set of parameters that worked. A better choice of parameters that maximizes the decision boundaries may well be possible for the non-pair-wise case. For the pair-wise *full* ruleset, a better choice can hardly be achieved on the tested datasets.

A limitation of this approach is the "critical mass", i.e. how many minutes of inertial data are required to make a decision about the sensor modality. The full dataset was used each time for feature computation. Varying this parameter would yield insights into the size of this mass, however was not attempted to avoid over-estimating the quality of the decision. Furthermore, standardization by dividing by the mean can be problematic if the sensor was asymmetrically driven into saturation. For example when the magnetometer was exposed to unipolar magnetic interference. In such cases, the mode could be nearer to zero yielding

	-	acc	gyr	mag
-		16		6
acc		339		
gyr			324	
mag				318

	acc	gyr	mag	-
-	276		205	-
acc	78		6	acc
gyr		324		gyr
mag	1		113	mag

Table 3.1: Confusion matrices for sensor modality identification with *full* (left-hand) and *partial* (right-hand) ruleset. The full ruleset fails to identify 2% of the analysed streams, but correctly identifies them for further manual inspection.

an incorrect classification. A possible solution could be to filter outliers beforehand.

3.4 PROCESSES IN ACTIVITY RECOGNITION

Activity Recognition can be split into several processing steps, which, when executed, form an Activity Recognition Chain (ARC) [4]. This can be implemented with a set of Unix tools, which can be independently developed and tested, and which only require to adhere to a specific in- and output format. This chapter presents these steps, specifies the in- and output in a simple text format that captures most Activity Recognition problems, and highlights the design of the *grtool*.

At the core of *grtool*'s design is an orchestration binary, which provides shorthands for all process steps involved. This executable resolves shorthands, executes the according executable, and connects standard in- and output channels. The ARC, which we will also call processing pipeline, is depicted in Figure 3.6.

Mathematically, the input to an ARC is originating from a set of raw data D from multiple sensors and ground truth labels L. Since all streams might be recorded at different rates, the first step is to resample them to a common sampling frequency. This step is known as resampling, and involves finding a common divisor of all sampling rates and then down- and upsampling each sensor. Ground truth labels, which are stored with timestamps, need to be discretized. This means, that each duration is split into samples according to the chosen common sampling rate, and each sample is assigned exactly one label. After, this

resampling and discretization step, the raw data D is pre-processed into D', segmented into frames/segments W, features from F then extracted, and finally classified into labels from C. Sensor and ground truth data is at the beginning of each pipeline, and we make two assumptions to simplify the overall problem:

uniform sensor sampling sensor data streams are sampled at a *fixed rate*. This facilitates demultiplexing, cross-stream synchronization and handling of missing data. Furthermore, timing dependencies of each sample can be disregarded. Some kind of interpolation needs to be applied during recording already, instead of pushing this step into the analysis phase, as is often done.

non-hierarchical and non-overlapping ground truth instead of supporting the more complex issue of arbitrary combinations of multiple set of labels, only a single set is supported. This does not limit the generality as hierarchical ground truth sets can still be supported by explicitly naming all possible combinations or overlaps.

The concept presented here, uses a text-based input format. Examples before and after each processing steps are given here, and since we assume a uniform sampling rate on the raw data, the primary input format before the pre-processing step looks like this:

```
label 1.2 2.3 4.4
NULL  0.0 0.0 1.0
...
label 3.4 4.5 6.3
```

A simple character separated value (CSV) format serves as the primary I/O format for all steps. The first row is the recognition target or label - a simple string. After that a varying number (but throughout the file) equal number of rows contains floating point number which constitute the raw measurements of multiple sensors. This number of rows is equal to n, and is the dimensionality of raw measurement vectors from D. The overall process and particularly the dimensionality changes in each step are depicted in Figure 3.6.

3.4.1 Pre-Processing

The pre-processing step is involved in any transformation of the raw measurements, which does not change the dimensionality of the measurement vectors. Elements of D' have the same dimensionality as

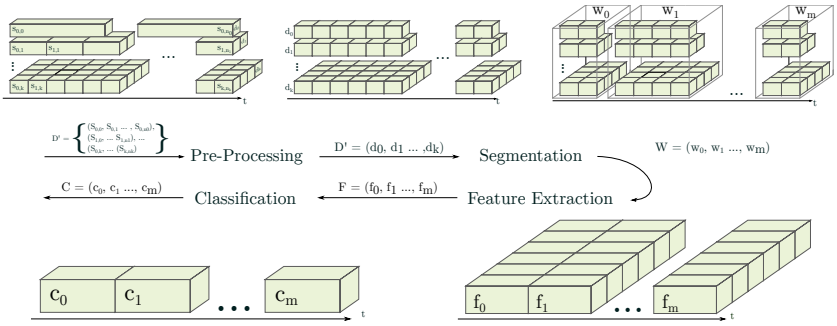


Figure 3.6: Cardinality of each processing step. Sensor data of different dimensionality (number of axes, measured values) sampled at different rates is first pre-processed. If required, rates need to be adapted to a common rate. Afterwards sensor samples are segmented into windows/segment and features f_m extracted from each window. The last step also removes any dimensionality dependencies of the sensor input and provides an input of fixed length for each window to the classifier, which then classifies these segment.

elements from D. Removal of sensor noise is a common operation. A band-pass filters frequencies that can be safely assumed to not add any information to the recognition task. Simpler filters, like a moving average or median filter are also often applied to remove sensor sampling artifacts. Sensor calibrations, for example removing offsets or applying scaling factors, are also applied in this step. Besides the values of the input, no other change is made, labels are just passed through to the next step. We designate the elements of D' as $d_t = (l_t, v_{t,0}, v_{t,1} \ldots, v_{t,n})$, where $v_{t,n}$ is the value of sensor n at time t, and l_t the ground truth label at time t.

After pre-processing the previously mentioned *rate-adaption* step changes the per-sensor dimensionality of the individual data vectors. This optional step is only required if multiple sensors were recorded at different rates. Changing the rates to a common rate allows for segmenting the input into defined segments.

3.4.2 Segmentation

Segmentation is the process of *spotting* relevant parts of the sensor data streams [256], i.e. those segments that contain information about

the activities which are to be detected. This phase yields elements $W = (w_0, w_1 \dots w_{m_i})$ which is a concatenation of several measurements aggregated into one segment/window. Empty lines designates the start and end of a segment of data in the output of the grtool. A Unix process for segmentation therefore copies data from its standard input to its output, and sporadically inserting empty lines to mark new segments:

```
segment1    1.2 2.3 4.4
segment1    0.0 0.0 1.0

segment2    0.0 0.0 1.0
segment2    0.0 0.0 1.0
segment2    0.0 0.0 1.0

...

lastsegment 3.4 4.5 6.3
```

Segments can contain a varying number of sensor samples. This support *signal-based* segmentation approaches which define a threshold on the signal's energy [257] or thresholds on the error of interpolation [36]. These interpolations can be linear or more complex [258], or change-point models [259], which are related to spline interpolations. These can be used, when such models can be safely assumed for the detection problem, for example when detecting eating one can assume turn points in the arm's motion that can be used for segmentation. Another example are quiet parts of an audio-recording, which surround segments of "interesting" sensor recordings. Generally, a heuristic on some part of the input signal might be a possibility for sensor signal segmentations. However, optimal segmentation is still an open research challenge.

In the absence of any heuristic, the standard approach is a *p-overlapping k-duration sliding window*. A window length k, which can range from sub-second to multiple seconds is chosen beforehand. This parameter is chosen to match the mean duration of a part of the target activity, e.g. single strides for locomotion. To overcome the issue of splitting a possible target activity into two parts, which only form a distinctive pattern together, an overlap of p percent can be chosen. The sliding window is only moved forward by this percentage each time. This results in segments of equal size, of which features can be extracted. One pitfall, particularly when segments overlap, is the pair-wise cor-

relation that is introduced due this segmentation, for which great care has to be taken in later steps to avoid overly optimistic results [156].

3.4.3 Feature Extraction and Selection

The feature extraction step serves two purposes: (i) to reduce the amount of data contained in each segment to minimize the computational complexity of the subsequent training and classification step, and (ii) to amplify patterns which are distinctive for a particular activity. To achieve this, *features* are extracted from the segmented sensor data and can be formally captured as a mapping from the sensor data space D to the feature space F:

$$f_i = \mathcal{F}(w_i) \tag{3.8}$$

Features are usually chosen due to application requirements, and often after a visual inspection of the sensor data and the target activities. Deep learning approaches leave the choice of features open and optimizes their selection with other parameters as well. A large number of different features were proposed, and they can be divided into three domains: _time-, _frequency- and *model-domain* features. An overview of possible features can be found in survey publications [4, 2].

Time-domain features, as the name implies, are calculated directly on the segmented data, including statistical moments like mean and variance, but also derivate, inter-quartile ranges, root mean square, segment duration, a histogram with a fixed number of bins, parameters of a fitted curve, cross-axis correlations et cetera. Frequency domain features, on the other hand, are extracted after the segment was mapped into the frequency domain, e.g. by applying a Fast Fourier Transform (FFT), Wavelet Transform (WT) or other type of "frequency" related transformations. Model domain features are those that require additional knowledge to be calculated, for example a model of the human body and the attachment points of each motion sensor [148] or the P-wave, QRS complex and PR interval from Electrocardiography. This allows to filter certain measurements, to create more distinctive features and create interpretable features.

A "good" feature set is one where features that correspond to a recognition target form clusters. The *Curse of Dimensionality*, however, dictates to minimize the amount of extracted features. Particularly if only a limited amount of collected samples are available. One strategy, that is also proposed here, is to include the feature selection into the

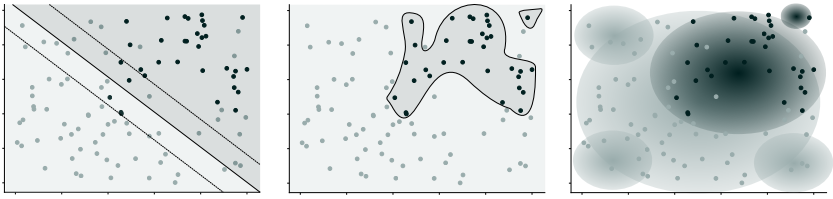


Figure 3.7: Decision boundaries of various machine learning algorithms. To the left hand side, a linear model with a soft error margin often used in Support Vector Machine (SVMs) is shown. The middle shows a more complicated decision boundary that can be estimated with a Random Forest, a non-linear SVM or Neural Network (NN). The right hand side shows a probabilistic model, multiple gaussians per class capture the feature space and estimate the probability for a particular class given a feature.

parameter tuning step and systemically test a (possibly) large number of feature combinations. Alternatively, techniques like the principal component analysis (PCA) can be applied, this however also requires careful validation [260]. Therefore, automatically selecting the most relevant features needs to be applied carefully, but can inform the design of features [147].

Implementation-wise, the feature extraction and selection step should transform each empty-line separated segment of the last step into a single line that contains the label and features for each step:

```
segment1    1.2 2.3 4.4 0.0 0.0 1.0
segment2    0.0 0.0 1.0 0.0 0.0 1.0
...
lastsegment 3.4 4.5 6.3 0.0 0.0 0.0
```

The meaning of each line changes in this step from a single sensor sample to a feature vector extracted over each segment. If the duration of the segments is not encoded by the feature extraction process and a non-equal segmentation was chosen, then this information is lost at this step.

3.4.4 Classification

The classification[2] step's goal is to determine the parameters of a mathematical model to learn the mapping $F \mapsto C$, i.e. to determine a projection from features to the recognition targets. These models range from computational complexity and simplicity of description. Visually, they can be captured by drawing the decision boundaries in a scatter plot as found in Fig. 3.7. These boundaries can take a multitude of forms on the feature space, depending on the chosen model. The related work section provides an overview of possible models.

The actual classification step is split into two phases: training and execution. For training the input data is split into multiple training and validation sets. The first one is used to estimate (train) the parameters of the chosen model for the decision boundaries. After training, the generalization and performance of the learned model can be estimated by classifying the validation set and comparing against the included ground truth. Different measures of the performance can then be evaluated in the light of the particular application to (i) choose the best performing model and parameter set, and (ii) to get an estimate of how well a problem might be modeled.

There is usually no "best" model that can be chosen beforehand, hence parameter selection and model selection should be part of the grid search for a performing ARC. Applying Occam's Razor here as well, models with fewer parameter have a higher probability of generalization. Formally, training is the process of determining the model parameter set θ from a set of observations from the training data set $\mathcal{T} = \{(f_i, c_i)\}_{i=1}^{N}$, with N pairs of feature vector $f_i \in F$ and labels $c_i \in C$. On the other hand executing a classifier is determining a prediction set $\mathcal{P} = \{(f_i, c_i)\}_{i=1}^{M}$ of M pairs where a feature vector $f_i \in F$ is mapped to a label $c_i \in C$.

One particularity of Activity Recognition is the handling of a so called NULL class, that is related to the earlier mentioned Activity Spotting challenge [256]. There might be segments that are irrelevant to the actual recognition task, but which cannot be explicitly marked accordingly. For example, the NULL class might not be sampled completely - the setting might even prohibit this. This could be the case when detecting eating, where eating gestures can be sampled, but all other possible movement that might be confused with eating might not. One

[2] Alternatively a regression is applied if the label space C is metric, i.e. there is a definition of a distance between elements of C.

solution to this challenge is to make use of the classifiers confidence, and only if this is above a certain threshold the classification results will not be NULL. For machine learning algorithms which do not output such a score, Platt's scaling method can be applied [149].

During training, the classification step consumes the input until parameters were successfully estimated, i.e. the model learned. Once a trained model is available the classification steps changes the feature vector line-by-line to a prediction. The output then looks like:

```
segment1  prediction1   0.5
segment2  prediction2   0.7
NULL      NULL          0.2
...
NULL      prediction2   0.5
```

The first field is the ground-truth label, the second field the actual prediction of the machine learned model, and the third field is an optional confidence score of the prediction. The latter one can be used for subsequent NULL class rejection.

3.4.5 Validation and Scoring

To estimate how well an ARC and its according θ parameter set performs for a particular application a validation step is crucial. The ARC design involves a choice of performance metric, and a cross-validation strategy to split the original dataset into a training and validation set. Both choices are application-dependent and should be chosen after formulating a hypotheses that is to be supported by data.

Cross-validation refers to splitting a dataset to emulate possible application scenarios. Two scenarios are common for Activity Recognition: (i) How well does the system generalize to unseen data, i.e. if a classifier is exposed to previously "unseen" data what is the probability of predicting the corresponding label correctly? (ii) When exposing the classifier to data of an unknown user, i.e. a user the system has no data on, what is the probability of correctly predicting labels? The first question allows to gauge the *generalization ability*, while the latter additionally gives insight into the *user dependence* of an ARC. Both strategies split the dataset, one randomly selects segments from the whole dataset, while the other selects segments that belong to a particular user.

Exhausting all possible dataset splits is often prohibitive, even for a limited case with 100 samples, of which 20% are to be used for

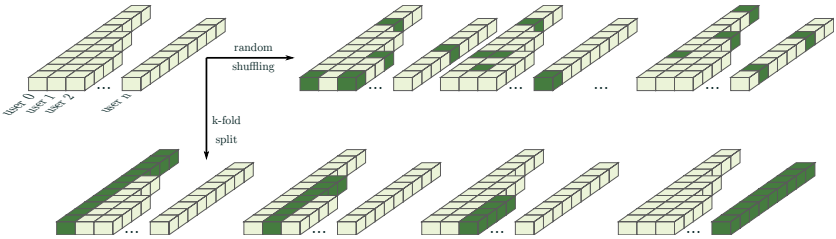


Figure 3.8: Cross-validation strategies on a segmented Activity Recognition dataset. The left hand side shows the dataset segmented by time and split by users. Segments highlighted in green are used for testing, yellow ones for training. The top one depicts random split, where segments are selected at random. The bottom an exhaustive search, where each user is left out of training and tested on.

training the amount of combinations is already $\binom{100}{20}$. A non-exhaustive strategy is therefore to select segments at random. Fig. 3.8 visualizes this. A smaller percentage of the dataset is used for testing and left out of training. It is important to leave testing samples out of training, otherwise the original goal of emulating unseen data is not fulfilled. Furthermore great care needs to be taken when selecting segments at random, since these are chosen from a continuous recording and might therefore be correlated. This correlation might lead to overly optimistic results, when picked up by during training [156]. K-Fold splits, where K segments are retained for testing and the rest used for training can be used when K is small. For *user-dependence* testing this split strategy needs to be used. K users are then left out, while the classifier is trained on data of the remaining users. All such combinations are to be tested to emulate the case of an unseen user base.

Stratification, i.e. balancing the amount of test samples throughout the splits needs be applied if the label distribution is imbalanced [261]. Otherwise, correlation performance results might just reflect the distribution of training samples. For example, imagine a dataset which contains 100 samples of sleeping, but only 10 of being awake. A classification that only "guesses" sleeping on each sample, all the time, will already provide good performance scores, since only 10 of 110 overall tests can turn out to be wrong. Either weighting results on each label by the overall occurrence, or by generating new samples for under-represented labels can be applied to counter this issue.

The actual performance analysis relies on the definition of true

positive (TP), false positive (FP), true negative (TN), and false negative (FN) cases. Implementation-wise this can be achieved by comparing line-by-line the fields of the last step. The first field is the ground-truth label l_i, and the second contains the prediction p_i from a classification:

$$score_i = \begin{cases} TP & \text{if } l_i = p_i \\ FP & \text{if } l_i = \text{NULL and } p_i \neq \text{NULL} \\ TN & \text{if } l_i = \text{NULL and } p_i = \text{NULL} \\ FN & \text{if } l_i \neq \text{NULL and } p_i \neq \text{NULL} \end{cases}$$

Counting these error classification line-by-line, allows to define a score of a trained model. Common measures include the *recall* ($\frac{TP}{TP+FP}$), the *precision* ($\frac{TP}{TP+FN}$) and *accuracy* ($\frac{TP}{ALL}$). These allow to get insights into a classifiers performance, where the recall roughly relates to the probability of detecting classes, and precision the probability that the prediction is correct. The generic goal of any classification approach is to maximize these two measures uniformly. The harmonic mean of both is called the *F1-score* ($\frac{2*precision*recall}{precision+recall}$). These measures can also be calculated from a *confusion matrix*, which counts the above cases and displays them in an n by n matrix, where n is the number of labels. The counts should be normalized to allow interpretation of imbalanced datasets.

Specifically for Activity Recognition, the given error definition is limited to the assumption that classified segments are not correlated. However, they often represent continuous times, i.e. one segment happened right after the other. For practical purposes, lower F1-scores might be preferable, if the classifier captures time-based correlation better. For example, a classifier that switches between labels slowly and introduces a detection lag, might have less spurious detection albeit providing a worse F1-score. Insights into such *event detection* can be gathered with an *event analysis diagram* (EAD) [139]. The EAD captures cases of fragmented and merged events, additionally to deleted and inserted events, which are the only ones captured by traditional measures. The output of this stage is therefore the *confusion matrix* for the detected classes, *recall*, *precision* and *F1-score* and the *event analysis diagram*, which captures the performance of a classification task.

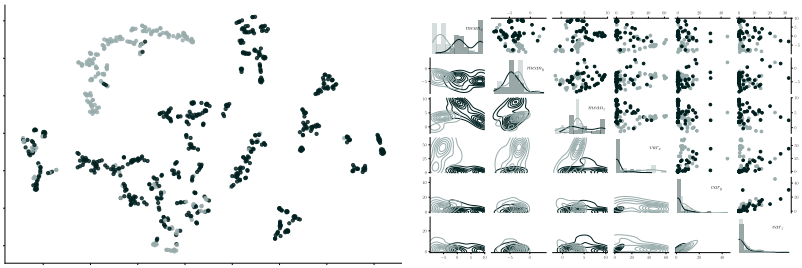


Figure 3.9: Two types of visualizations for a high-dimensional feature space. The left hand side shows a scatter plot after feature reduction with the t-SNE approach. The right hand side shows a matrix scatter plot over all feature dimensions, with optional gaussian density estimation. The dataset is a small fraction of the smoking dataset presented in the following chapter

3.4.6 Debugging and Visualization

To understand and guide the design of an ARC, performance metrics only provide an indirect insight into the workings of the chosen design parameters. A search for an optimum over a large search space of ARC parameters, will only yield a solution in this particular space. To introduce novel parameters, to simplify the problem or to find an explainable solution to the classification task, the model needs to be interpreted and therefore visualized. This involves the visualization of the high-dimensional feature space F that is mapped into the categorical class space C.

T-distributed stochastic neighbor embedding [262] is a popular technique to reduce the dimensionality of a feature space to two or three dimension. This allows to get insights whether decision boundaries can get placed easily, additionally plotting these boundaries allows for visual inspection of the learned model. However, due to the data reduction the dimensions of the resulting plot are hard to interpret. If the feature space has a rather small dimensionality (< 10), a scatter plot matrix can provide insights by plotting all pairs of dimensions. Parallel coordinates are similar to scatter plots, however axes run in parallel instead of orthogonal. This way multiple dimensions can be put next to each other, and allow for stacking. A combination of matrix scatter plots and parallel coordinates was investigated in [263]. This visualization

allows to explore all possible orders of feature combinations. Fig. 3.9 shows example of each visualization on a toy dataset.

In the proposed framework, data is loaded in the aforementioned text-format. Each line contains a segment that is to be classified. The first field contains a label, the remaining field designate the feature vector. Classes are visualized as different color on each plot, while the feature vector with a choice of different graphs. Due to the design of loading data from the standard input, a reactive plot that updates on new data is easily implemented. This allows to visualize an ARC operation while in progress, or when recording data from live sensors.

3.4.7 Scalability and Parallelization

When applying machine learning multiple factors contribute to the computational complexity of finding a performant classifier: (1) a large number of sample instances, (2) a complex feature set, (3) model selection and parameter grid search, and (4) the combinatorial complexity of cross-validation. This is not an exhaustive list of factors, see [264] for a more thorough treatment. Compared to other challenges, a large number of sample instances requires a modification of the specific training algorithm used for the chosen machine learning, and is therefore out of scope for the presented proposal. However, the remaining challenges present an embarrassingly parallel workload. When the ARC is modeled as a Unix process, the actual parallelization task requires zero implementation effort.

For this, think of an ARC either as a chain of Unix process connected via pipes, or as a single Unix process. The input to this chain is raw sensor data and the hyper-parameters of this learning and prediction workload are provided as arguments to this process. For a typical task, the hyper-parameters include the choice of segmentation strategy, choice of features like the mean, median, variance, range of values etc., and a choice of learning algorithm and its parameters. Additionally, almost always a cross-validation is done, which also requires the dataset to be split. This can be done by retaining the training set during the training phase. When implemented as a Unix process this can be expressed similar to this command line:

```
$ cat dataset | segment -W 10 | extract time |
  train -s .8 -n 20 RF | predict | score
```

which would train a RandomForest, with 20 tress, on 80% of the

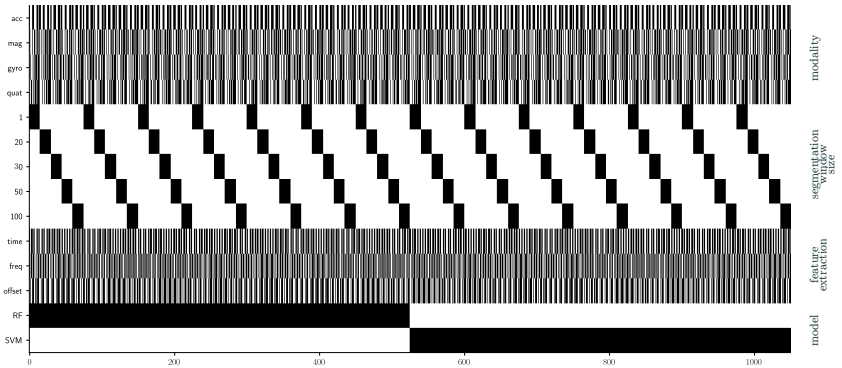


Figure 3.10: Example of a parameter grid search, which scores all combinations of a four element sensor modality set, five different window sizes for a sliding window segmentation, all combinations of a three different feature extractions and two different machine learning models. Even this rather small grid, already requires a full ARC cross-validation.

input, extract time-domain features, and segment the dataset into multiples of 10 frames. The actual data transported between the Unix pipes are in the formats as specified previously.

This command returns the score for exactly one element from the hyper-parameter set, which is expressed as the arguments to these commands. We can denote this set as θ, which contains tuples $h = (\text{split, segmentation, feature extraction, } \dots)$ that contains a discrete set of parameter choices for each step in the ARC. The combinations of all those parameters is the hyper-parameter set, of which all elements need to be tested separately. As this represents the product of each step's hyper-parameters the search space quickly explodes, and therefore needs to be selected carefully.

Fig. 3.10 depicts the combinations of parameters that need to be tested, even on a small parameter grid. For each tested parameter combination a full cross-validation needs to be executed to provide score for this particular set. However, each parameter combination can be cross-validated indecently, and each cross-validation split can be seen as another parameter of this set. Hence, this task is easily parallelized.

When implementing ARCs as a Unix process, that takes the hyper-parameter set θ as arguments to each command, the GNU parallel [265] package can transparently distribute each element of θ to multiple cores

and multiple machines. Hence, distributing the grid search for the best performing model to a cluster of Unix machines. For example the parameter sets of Fig. 3.10 can be distributed to multi-core cluster with the following command line:

```
$ parallel --slf cluster "cat dataset |\
  arc -m {0} -w {1} -f {2} -l {3} -s .8 | score"
  ::: acc mag acc,mag
  ::: 1 10 20 30 50 100
  ::: time freq time,freq
  ::: RandomForests SVM
```

GNU parallel builds the product of all parameter sets specified after the triple colon, which represents θ. It will then start each job (or particular choice of elements for the parameter set) on one of the machines specified in the cluster file, which must be reachable via ssh. The arc command in this case, is an implemented recognition chain that takes the *m*odality, *w*indow size, *f*eature set, *l*earning algorithm and the *s*plit fraction parameter. The parametrized chain will be build by replacing the concrete parameter for each run. The resulting score for each parameter combination will be printed on standard output and can be further processed, for example by selecting the best combination by the highest recognition score.

Compared to other frameworks for parallelizing machine learning, this approach is more flexible, since each step can be quickly replaced with a different implementation. In other frameworks, replacing a step usually requires a modification in the same programming language and an adaption to the provided data abstractions. Parallelization is also harder to achieve, as this usually requires delicate programming if data flow is not cleanly separated between the different steps of an ARC. In contrast, separating these steps as Unix processes allows for system-level parallel processing with little implementation effort.

3.5 SUMMARY

This chapter presented an Activity Recognition framework designed according to traditional Unix philosophies: to write programs to do one thing and do it well, and to make them work together by only specifying their in- and outputs. The steps of an Activity Recognition Chain are encapsulated in single Unix processes. This leads to greater flexibility,

as single steps can be easily replaced with different implementations in the choice of machine learning framework or programming language a developer is prominent with. These steps are then connected via the Unix piping mechanism. The core idea is to combine these steps finally into a further command that executes training and prediction, given hyper-parameters including a particular dataset split as command line arguments. This facilitates the common task of cross-validating a large dataset, as well as hyper-parameter optimization for finding well-performing ARC parameters. Process-level parallelization, including execution on a cluster, of these tasks can then be easily created with minimal implementation effort.

For implementing these task with the Unix piping mechanism, the inter-process communication format needs to be defined. Traditionally a text format is used, since modification and parsing can be quickly achieved. However, this also incurs a large processing overhead. An alternative, that can also be used for long-term curation of datasets as well, are multi-media container formats. These allow to encode multiple sensor data streams with additional compression. Each step in an ARC either adds or replaces multiple of those streams, while secondary evidence like video recordings stay untouched.

One challenge, when working with publicly available datasets, is the identification of sensor modalities. When using multi-media containers, such meta-data can be stored side-by-side with the original data. However, the current state-of-the-art is to store dataset in CSV files, which requires manual identification of fields and their meaning usually with the help of a readme file. A rule-based identification scheme that extracts this information for inertial motion modalities directly from the data, allows Webcrawlers to find new datasets automatically and provides an additional safety check when recording datasets. In the presented evaluation, the ruleset correctly classifies 98% of 1003 streams in five different human motion datasets.

A further challenge is the visualization of the usually high-dimensional feature space requires elaborate techniques. For low-dimensional feature spaces, a matrix of scatter plots provides good overviews. For higher dimensions, a dimension reduction like t-SNE can be applied but are harder to interpret. For debugging purposes, the decision boundaries of the learned model can be drawn as well, which allows to spot implementation problems and can guide further changes to the ARC.

Designing Activity Recognition chains with the help of Unix pro-

cesses, allows for more flexibility when creating each step. Process-level parallelization and cluster distribution is further facilitated by this design. Thus, a system is created which allows for fast replacement of ARC steps and quick distribution of hyper-parameter optimization and cross-validation tasks on a cluster of Unix machines.

This chapter contains contributions from the following peer-reviewed publications:

Philipp Marcel Scholl and Kristof Van Laerhoven. "A Multi-Media Exchange Format for Time-Series Dataset Curation." In: *UbiComp 2016 Adjunct - Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. 2016.

Philipp Marcel Scholl and Kristof van Laerhoven. "On the Statistical Properties of Body-Worn Inertial Motion Sensor Data for Identifying Sensor Modality." In: *International Symposium on Wearable Computers*. 2017.

SMOKING DETECTION WITH WEARABLE SENSORS

Twenty-two percent of the global population older than fifteen are consuming at least one cigarette per day [268]. And even though the percentage of daily smokers has decreased by roughly 30% in the last decades, the global number of smokers [269] increased to a total of 967 millions due to steady population growth. In Europe, for instance, 27% of all citizens are still regular smokers, especially in age groups below 55. Around 59% tried to quit at least once [270], often motivated by fear of personal health consequences. This is understandable as continued smoking has been most prominently linked to cancer forms of highest mortality rate, as well as other diseases causing premature death [271]. The resulting economic loss, both due to absenteeism in the workforce, as well as the cost for treating tobacco-related diseases was estimated to about €313 billion [272] in the EU and $191 billion in the US [273].

This chapter answers the question, if smoking behaviour can be assessed objectively with the help of wearable sensors. To be called objective, an assessment should be inconspicuous, accurate and efficient. Inconspicuous, so participants are not aware of them being tracked, as to not alter their behaviour. At the same time this assessment should reflect the real events as close as possible. Otherwise, important details could be missed, or wrong conclusions drawn. Resources, such as a participant's involvement, available battery and processing power should be minimized to allow for a longitudinal assessment. This is particularly challenging for smoking, because current approaches either ask for noting down times on paper, tracing bio-chemical markers [188], interacting with a mobile device, wearing unfamilar devices, or at unfamilar locations.

Instead of relying on the discipline of a participant to provide an ecological momentary assessment (EMA), wearables can automatically detect smoking events. The overall goal is to detect these smoking events as reliable and efficient as possible. With this *objective* assessment, new insights into smoking cessation can be gathered. For example, the efficacy or impact of an intervention program is currently measured with the abstinence rate after 3-, 6- and 12-months. These numbers are commonly gathered via telephone interviews, adding delay and uncertainties. With an ambulatory assessment, with any sensor-based technique, these number can get a lot finer. This in turn could provide new insights into the efficacy of cessation tools, like nicotine patches or personal counseling. Novel cessation approaches are another possibility. Intervention material can be (automatically) personalized, interventions can be provided just-in-time, novel insight generated for the smokers

or cessation material directly tested on its usefulness.

The first inspiration for designing the Smartlighter was to gather reliable ground-truth for smoking detection from wearable sensors, and presents yet another application. Gathering such ground-truth is one of the core challenges of any system attempting to detect smoking from sensor data. Integrating this with a background data collection of wrist motion data, is the basis for our data collection approach. With this system in place, it is possible to collect datasets with a higher ecological validity than those collected by other EMA approaches. To quantify this effect, three EMA approaches are compared to the Smartlighter: (I) estimating overall consumption by recall, (II) estimating consumption via wrist motion in a lab setting, and (III) estimating via wrist motion in the wild. Prior to these evaluations the design of the Smartlighter and its characteristics will be described, followed by a detection approach from wrist motion.

4.1 INSTRUMENTED LIGHTER SMOKING DETECTION

During the design of the Smartlighter, we followed the guidelines laid out by Li et.al., to design systems which allow "collecting data anytime, anywhere and often", to "support different kinds of collection tools" and to "reduce the upfront cost of data collection" [274]. The result of this were the prototype iterations of the *Smartlighter*, which logs its usage into the internal memory of a microcontroller, gets more usable, adds communication capabilities, and is easier to manufacture after each iteration.

Our prototype is motivated by the fact that monitoring the use of a cigarette lighter is a straightforward, robust and inconspicuous way to track a smoker's consumption behavior. Tracking itself becomes trivial, and thus provides a more reliable solution. Implementing such a device is however aggravated by the availability of lighters that generate a measurable electronic signal when lit up. Currently, there are four lighter types widely used: gas, petrol, electric arc and electric coil lighters. Coil lighters, similar to the ones found in cars, work by closing an electronic circuit which heats up a coil with a large current[3]. Arc lighters work by generating a plasma between two electrodes. Both electronic lighter are usually powered by a USB re-chargeable battery. Gas and petrol, on the other hand, store energy in form of a flammable,

[3]Other approaches, for example mixing the sample set randomly, would invalidate the post-processing step, since it assumes a time-ordered test set.

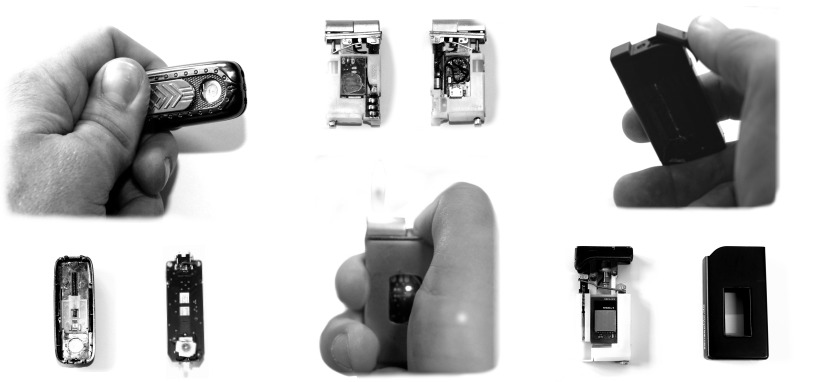


Figure 4.1: From left to right: customized USB-chargeable electronic lighter, gas lighter with mechanical switch, and a piezo-ignited jet lighter. The latest iteration is completly wireless.

evaporating fuel. A spark generated by mechanical force ignites this fuel: generated by scratching a flint stone or by a high voltage discharge of a compressible piezo element. These constitute the basic working principles from which a measurement mechanism must be deduced.

Each prototype logs the date and time of each ignition, which results in a list of events that are likely related to cigarette smoking, especially if the user was properly instructed on the lighter's use. Multiple ignitions in quick succession are filtered, assuming that only a single cigarette will be consumed in a five minute interval. All prototypes measure those events, and differ mostly in usability and communication capabilities. The lighters are described in the following sections.

4.1.1 Smartlighter v1: Heating Coil

Initially, the printed circuit board (PCB) of an electronic lighter (cf. Fig. 4.2) was replaced to include an ATMega32U2 micro-controller and an external real-time clock (RTC). A 200mA h battery provides power for heating up the coil, and the micro-controller. Events are logged into internal memory and can be retrieved via USB - they are detected by monitoring the state of the ignition contacts. Closing these contacts wakes up the micro-controller and ignites the lighter. The components were packed into the original lighter casing to provide a prototype that

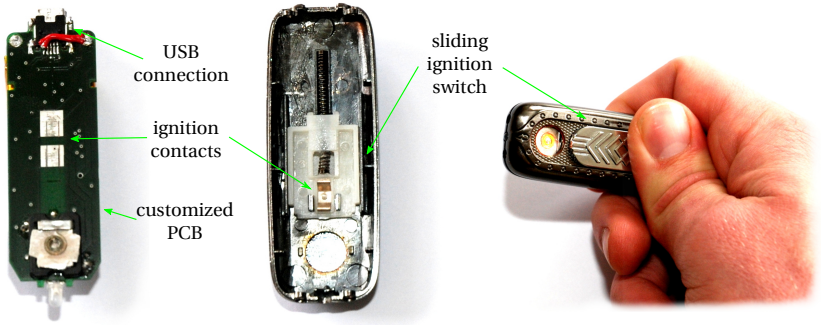


Figure 4.2: The Smartlighter v.1's internal buildup. On the left, a mechanical switch closes the circuit between battery and a coil, allowing it to heat up so that a cigarette can be lit up. The time and duration for which the switch was used is logged by an on-board micro-controller that is connected to a real-time clock. The right-hand side shows the lighter in use.

is robust enough for day-to-day usage.

The firmware is designed to consume as little power as possible; During periods of no activity the micro-controller is in deep sleep mode and only wakes up on USB activity or when the switch contacts change their state. Only the RTC is constantly drawing power, which leads to an overall standby power consumption of 0.112µA. Whenever the switch is moved, the micro-controller wakes up from sleep, reads the current time from the RTC and appends the time-stamp to a list in flash memory. Each timestamp takes up 4B, which allows to store up to 255 events in the 1kB sized internal memory of the micro-controller.

Although this first prototype was found to work well in preliminary trials (see [28]), several shortcomings were found that hinder more extensive deployments. A first issue that some users experienced was the mechanism: this requires sliding down the switch for a considerable amount of time to sufficiently heat the coil, which for several users was found to be both unfamiliar and not as pleasant as a traditional gas lighter. Due to this, cigarettes were harder to light up, as the coil needed at least five seconds to heat up. When the battery provides its nominal voltage, after it was discharged to about 70% of its initial capacity, this took even longer. The lighter was also harder to use, since it requires careful aiming of the cigarette onto the heating coil, and if the

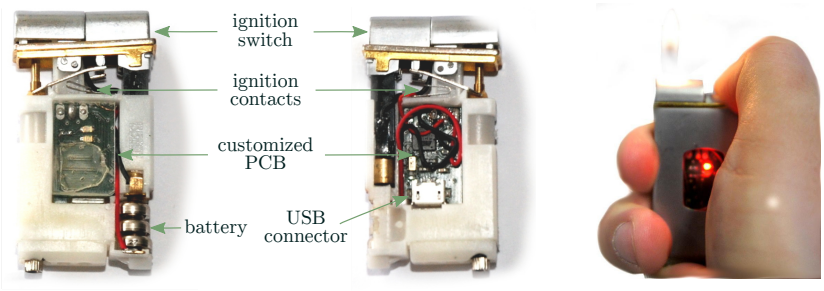


Figure 4.3: The Smartlighter v.2's internal buildup. The ignition contacts additionally close a circuit, which is read by the micro-controller. On the left-hand side the battery compartment and LEDs are visible. The center shows the RTC, USB connector and microcontroller. The right-hand side shows that the lighter operates like a traditional lighter.

coil was not yet hot enough the cigarette would break. A more critical shortcoming though, was that due to the high power consumption of the heating coil, the system runtime is limited to about two to three days for frequent smokers ($\sim 15\frac{cigs}{day}$). Several users were bothered by this, which led to a few cases of missing data logs.

4.1.2 Smartlighter v2: Gas

Despite having a very different form factor, the PCB for the second version of the *Smartlighter* essentially contains the same electrical components. The ATMega32U2 micro-controller is directly connected to the gas lighter's ignition contacts, which are the contact pads which get shortened when pushing the ignition button (see Figure 4.3). Together with a external real-time clock (RTC), a USB port and two status LEDs, the logging of smoking instances is performed. The main improvements to the first version are (1) the more familiar form factor of a gas lighter, as well as (2) the fact that the three small LR41 coin cells included in the gas lighter provide 28mA h of power. The cells can continuously power the lighter for about 7.2d at frequent usage ($\sim 15\frac{cigs}{day}$). This increase of runtime, while decreasing the available power ten-fold, is mostly due to the use of combustible fuel for providing a flame to light the cigarette.

The process of capturing and recording the smoking instances is

for the second version similar to that of the first: if there has been no write operation during the last 5min, the timestamp is written into the internal non-volatile memory of the micro-controller. The 5min interval serves a double purpose: First, it is a simple mechanism to debounce the ignition switch and second, it filters incidents of multiple ignitions sometimes needed to light up a cigarette. The 5min interval has been chosen as the mean time to consume a cigarette [275] [4]. The smoking incident timestamps can be downloaded from the lighter via a virtual serial port emulated by the Atmega32U2 in CSV-format through the USB-port.

4.1.3 Smartlighter v3: Piezo Ignited

While the (novel) form factor meant that smokers did not need to adapt their behaviour, there were several practical issues. First of all, the USB port was inside the enclosure, which needs to be opened to attach and download data to a PC. Although this makes the prototype highly robust, this version can provide long-term feedback only during maintenance phases - no real-time feedback was possible. Due to the mechanical connection to the lighter this process often required to repair the lighter afterwards.

To remedy this situation, optical transmission from the included lighter to unmodified cameras in commodity hardware was investigated [276]. Due to the limitations and unreliable frame rate of webcams only very low speeds of 17.3bps could be achieved. Implemented using standard-compliant interfaces of a webbrowser's Javascript engine, it is a platform-independent solution. It is also rather cheap, after all only LEDs are required. However downloading a day worth of data (15*cigs*, 2B per compressed timestamp, totalling 34B) takes at least 16s. And, being camera-based, lighting condition have a strong influence, rendering this solution impractical.

Further design choices considered for this iteration of the Smartlighter included the modification of Zippo lighters. Temperature or contact sensors and batteries could be added. However it turned out to be a challenging task to keep electronics and fuel safely separated. Generally, the inclusion of batteries and electronics in light-weight, com-

[4] The systematic review investigated smoking topography studies with a total volume of $n = 193$ participants, which reported $17.23 \pm 6.88 \frac{cigs}{day}$, with an *inter-puff* delay of 15.46 ± 7.18s, and puff duration of 2.32 ± 3.3s (Method 1). The total consumption time can be sampled to 306.75 ± 129.01s.

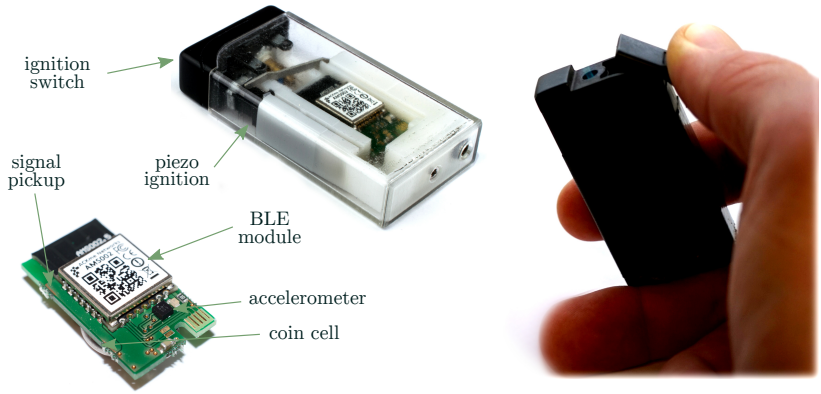


Figure 4.4: v.3's interal build. Contactless ignition detection is achieved by monitoring for a high-voltage spark generated by the piezo ignition. A Bluetooth Low Energy (BLE) communicates events directly to connected Smartphones.

mercial lighters is tough, as there is only a limited amount of designs available. The only alternative is to either attach the electronics in the compartment holding the flammable fluid or on the case's outside.

Due to the mechanical instability of prototype v.2 another detection mechanism was devised, which improves the lighter's manufacturability. Instead of relying on a mechanical switch connected to a micro-controller pin, the ignition is now picked without any contact. Common lighter use a standard piezo ignition. Compressing the piezo element inside this ignition results in a high-voltage spark, which is used to light up flammable fuel. A large copper area in the vicinity (cf. Fig. 4.4) of this ignition connected to a pulled-down micro-controller pin will pick-up a voltage above the micro-controller's logic level. This can be used to wake the micro-controller whenever the lighter is ignited. By carefully controlling the size of this area, no external electrostatic discharges (ESD) are picked up. This way, the manufacturing process is simplified to placing the module near the piezo ignition.

Another novelty of this prototype is the addition of BLE to communicate with nearby Smartphones. Introduced in 2015 it was the first wireless communication standard which was power-efficient enough to be run from a coin cell and was pervasively included in Smartphones. Chosen for its size and price, Zentri's AMS002 [277] Bluetooth Low Energy module was used to implement this prototype. The times-

tamps of ignition are now not just stored in local memory, but also communicated to nearby phones. Just-in-time interventions are now possible, and sensor recordings on other devices and external computations can be triggered directly. Once timestamps have been retrieved from the lighter, they are deleted from local memory. Communication is limited to two minutes after an ignition to conserve power. If no communication partner was in range during this period, the event will be communicated again with the next ignition event. Thus creating a robust, reliable and real-time monitoring system.

The lighter module is powered with a single coin cell with a capacity of 48mA h. As shown later, the lighter can be powered from this cell for 1.5 months for frequent smokers. Power could possibly be harvested from the piezo element itself [278], as well as other sources. Rudmann [279] investigated the specific possibilities for the presented lighter; While it is possible to send BLE datagrams with an inductive harvester, the power required to boot the included BLE module could not be generated. However, other modules with smaller power requirements maybe able to send a small number of datagrams with a single flick of the inductive generator. The battery could then be removed, however the prototype would be harder to construct and a different BLE module would be required. Other harvesting options (e.g. solar, fuel cells, piezo, pyro-electric ...) are impractical, mostly due to limited construction space, see [279] for further details.

The system is packaged in a re-purposed gas jet lighter, which originally included a decorative block of acryl where the PCB now resides. We opted for not using the included battery compartment to simplify the manufacturing process. No manual soldering is required, since the PCB can be assembled automatically. Due to this, user can replace the coin cell themselves and also use the lighter like any other. Thus creating an inconspicuous monitoring option. This lighter was used to collect the ground-truth data, which provides the database for wrist-motion based detection of smoking for an in-the-wild study. Prior to the description of this study, we look at the energy consumption of the different prototypes.

4.1.4 Firmware and Energy Consumption

To discuss the power consumption of the three Smartlighter prototypes, the firmware of the lighter can be split into five states. The state diagram in Fig. 4.5 shows the flow of states: after wake-up the ignition state of

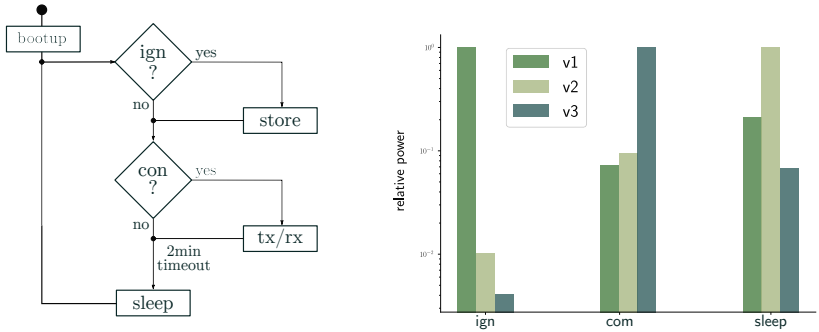


Figure 4.5: Firmware states for all version of the UbiLigher. The micro-controller is most commonly in a sleep state, only waking up for communication (com) when the lighter is ignited (ign). The right-hand side shows the relative power required for each prototype.

the lighter is checked (*ign?*), as the lighter can be woken up by a battery insertion as well. If an ignition was detected, the local timestamp is stored (*store*) in non-volatile memory. The communication-enabled lighters (v2/3) will start a discovery phase (*con?*) which advertises the event for 2min. Only the Bluetooth lighter will be waiting for a connection from an external device, the LED-lighter transmits events unconditionally. If a Bluetooth device connects during this period, the events will be transported (*tx/rx*) in an acknowledged manner. Events are deleted from lighter-local memory after successful transmission. After this communication period, the lighter will enter a low-power sleep mode (*sleep*) until the next ignition is detected. If no events could be transmitted the event is kept in a list in the internal non-volatile micro-controller memory. Each state has particular energy consumption, which allows to calculate the overall consumption in terms of usage time of the lighter.

The current consumption is typically in the mA to sub-µA range, which requires a measurement utility with a low burden voltage. For this measurement, a PicoScope [280] in combination with a µCurrent [281] voltage amplifier was used to measure the current consumption I. The system voltage V was additionally measured. Each state is then sampled and the energy consumption readily calculated as $E = \int VIdt$. The prototypes do not only differ in power requirements per state, but also in time a particular state is active. For example, prototype v1 stays

	v1 (t[ms]/E[mJ])	v2 (t[ms]/E[mJ])	v3 (t[ms]/E[mJ])
ign?	9000/44550	520/24.32	740/14.12
store	559/33.62	20/1.54	
con?			500/6.95
tx/rx			2320/33.06
sleep	1000/.112	1000/.528	1000/.036

Table 4.1: Energy consumption of each prototype. Values are given in Joule (J). Each system typically operates at 3.3V, which allows to convert to mA. These figures were measured with a PicoScope 3206 and a µCurrent Gold.

in the ignition state until the coil is heated and the cigarette lit. The other two prototype just need a quick check on a signal line for this state. Table 4.7 lists the energy consumption of each state, and the mean amount of time this particular state is active.

The first prototype (v1) is powered by a rechargable li-ion battery (200mA h), v2 is powered by three coin cell (28mA h), and v3 by a single coin cell (48mA h). Given the values of Table 4.7 the runtime of each prototype can be devised from typical usage scenarios. For the EU27, the most typical cigarette consumption amounts to 11-20$\frac{cigs}{day}$. If we always assume the heaviest smoker, the runtime can be calculated as given in Table 4.2. The first prototype exhibits the heaviest energy requirements during the ignition state, when the coil is heated. For the other lighters the *sleep* consumption becomes more important, as the *ign?* state is only active for a short period of time. As can be seen, the day-to-day consumption of v3 is only a fraction of v2, even though BLE connectivity was added. The higher energy consumption of BLE connectivity is however balanced by a lower sleep consumption.

From a technical perspective, the main routine of the instrumented lighter is to log the date and time of its ignition. A stable time source is required. In the first two version this was achieved with an external, constantly powered real-time clock (RTC). This clock was synchronized prior to handing it out to study participants, and re-set during maintenance phases. Due to the missing connectivity of the first two prototypes, a constantly powered RTC is the only option to capture exact timings. The third prototype used the micro-controller internal RTC to provide timings with an accuracy down to a single second. It is constantly synchronized to a client's clock by providing the current

	per cigarette	per day	battery capacity	runtime
v1	44.88J	897.98J/34%	200mA h/2664J	2.96d
v2	25.86mJ	46.14J/14%	28mA h/333J	7.22d
v3	255.68mJ	12.39J/2%	48mA h/570J	46d

Table 4.2: Runtime estimation based on nominal battery capacity. The total consumption per cigarette is the sum of all but the sleep state, while the total per day is the sum of sleep consumption throughout the day assuming a consumption of $20\frac{cigs}{day}$. Values are given in per-cent of total battery capacity and absolute runtime.

time on the lighter-local clock with each transmission. This way, assuming a transmission creates negligible delay, the difference of the sent timestamp and the reception timestamp equals the ignition timestamp.

On the lighter, a list of ignition events on the lighter-local clock is stored. This list is transmitted whenever a client connects, and events are deleted from memory after successful transmission. There is no difference concerning this strategy, whether the events are transmitted via the USB- or BLE-connection. For the BLE-lighter, however, the sent timestamp on the local clock is transmitted additionally for synchronization purposes.

4.1.5 Lessons Learned

Looking at the design decisions for the several prototypes in retrospect, the following design principles would have possibly provided more *mature* results quicker:

Optimize for original purpose first. While the initial prototype provided first insights into the hardware choices required for including electronics into a readily available lighter, the lighter itself was *not* (well) designed for its primary usage. From the perspective of collecting smoking data (our intended usage), starting from an existing lighter was a very good choice as it included a battery and a USB connection already, so only minimal changes were necessary. Its primary use for lighting cigarettes with a heated coil, however, was often challenging for smokers. During its study use, smokers often reported to use the instrumented lighter for logging, next to a traditional lighter for actually lighting up the cigarette.

Design for robustness of everyday objects. The second prototype had to be opened up to retrieve the logged smoking events. In principle, this created a one-time usage device, as the reassembly after data retrieval was almost the same as assembling a lighter from scratch. This was partially due the measurement principle, which required fragile, internal wiring that would often break after deployments. Contact-less data transmission, as with the third prototype, would have solved this issue.

Render tangible and immediate interactions. All prototypes had LEDs on them, to display the lighter's internal status. This interaction, however, was fairly limited and feedback from early adopters often revolved around the point that they would like to get immediate insights of their measured smoking behaviour. For the first two prototypes, such feedback was only given, when they visited us again so that we could retrieve data from the lighter - while this allowed for inconspicuous monitoring, it also made the idea for the smoker less transparent.

4.2 SMOKING DETECTION FROM WRIST-MOTION

This chapter presents the investigations on how well smoking can be detected by sampling wrist motion with inertial sensors. A feasibility study [28] showed that *prototypical* hand-to-mouth gestures can be successfully detected - even without the application of machine learning. The feasibility study also showed, that there might be motion patterns that indicate smoking but have a larger variety, which requires the application of machine learning and the existence of a high-quality data with reliable ground-truth. Collecting this ground-truth was the original motivation for developing the instrumented lighter [282]. Used in conjunction with recently available Smartwatches to capture the full set of inertial sensors while smoking, provided the dataset for a follow-up study. These studies are primarily motivated by the following questions:

- What is the probability of detecting smoking from wrist-worn accelerometer data of smokers in free-living conditions?
- Does detection benefit from gyroscope and magnetometer data?
- Is the fused wrist attitude expressed as quaternion useful?
- What are the major challenges for motion-based smoking detection?
- What is the minimum amount of energy required for continuous detection?

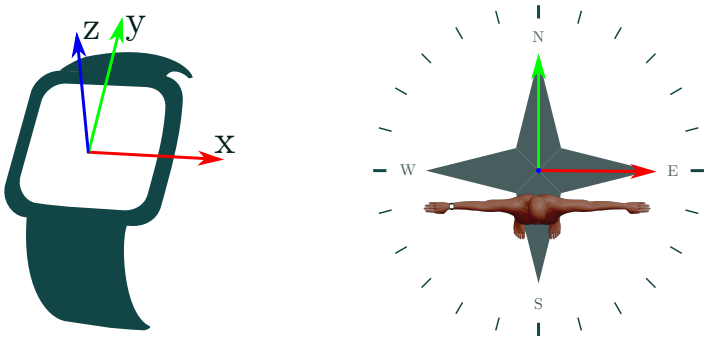


Figure 4.6: Axis alignment on Android Smartwatches (left hand side), and frame-of-reference of the rotation sensor (right hand side). The X axis of the Smartwatch is pointing along the arm, when worn on the left hand it points along the fingers, when worn on the right hand it points to towards the body. The frame-of-reference is given according to the geomagnetic north, and east, the Z axis is pointing towards the sky.

4.2.1 Sensors, Attitude and Frame-of-Reference

Before delving into detecting smoking from inertial wrist motion, the basics of inertial motion measurement need to be cleared up. Nowadays, inertial motion sensors always measure three perpendicular axes. A full set of these sensor consists of an accelerometer, a magnetometer and a gyroscope, which measure the acceleration, magnetic flux and rate-of-turn respectively. Integrating the rate-of-turn over time, provides the attitude, i.e. the orientation of the sensor in three-dimensional space. However, due to sensor drift, this orientation quickly becomes inaccurate, but can be stabilized with the help of the accelerometer and magnetometer. The accelerometer provides a static reference for the orientation on the vertical plane, while the magnetometer can be used as a compass to provide a reference on the horizontal plane. Combined, these sensors can be fused to estimate the *absolute* orientation of the sensor.

Such an orientation is typically expressed as a quaternion (though any other representation of a rotation in 3-dimensional space is possible). An *absolute* orientation refers to the fact that the orientation is expressed

according to a static, global frame-of-reference. Fig. 4.6 shows the axis alignment of an Android SmartWatch, and the global frame-of-reference, when wearing the watch on the right wrist with the display pointing upwards. The frame-of-reference of the attitude is fixed, and does not change when moving the wrist.

In contrast when using only acceleration to estimate wrist's orientations, only the rotation on the vertical plane can be easily estimated. The frame-of-reference of this rotation then moves together with the arm when moving on the horizontal plane. Expressed differently, when using only acceleration to estimate the wrist attitude from gravitational pull, only the rotation on the vertical plane (along the body) can be estimated. With the addition of the magnetometer also the rotation on the horizontal plane is detectable. And finally with the addition of a gyroscope, the orientation can also be estimated while the sensor is in motion or under the influence of a magnetic field other than the earth's.

When using a wrist-worn sensor to guess an arm's orientation, an additional rotation induced by different wearing styles needs to be accounted for as well. For a wrist measurement, a fixed rotation R can be applied, to move from device-orientation to arm-orientation. The attachment, i.e. worn below or above the wrist and with the display to or away from the user, are four fixed rotations R that can be applied to each attitude estimation of the wrist. This rotation can also be estimated from a long-running recording, or orientation-independent classifiers can be tested [283, 35].

However, to simplify the classification in the following pages, we will not use the attitude expressed as a quaternion directly, but rather define the arm/wrist orientation as an accordingly rotated reference vector. We deliberately chose the default orientation as the *right wrist with the display readable*, while the wearer is facing north and the right arm dangling along the side. With this it is possible to define the unit vector v as the representation of the current arm's orientation. See Table 4.3 for the combination of reference vector and wearing styles. Applying the rotation to these vectors, removes the influence of the wearing style and result in an expression of the arm's attitude in a global frame of reference. Still, the device-local reference of Table 4.3 must be known beforehand

left wrist	right wrist	
(-1,0,0)	(1,0,0)	top or bottom
(1,0,0)	(-1,0,0)	rotated top or bottom

Table 4.3: The orientation vectors rotated by the wrist attitude. Using these vectors allows the arm's orientation to be expressed as a vector independently of whether the sensor was worn on the left or the right, or rotated around the wrist.

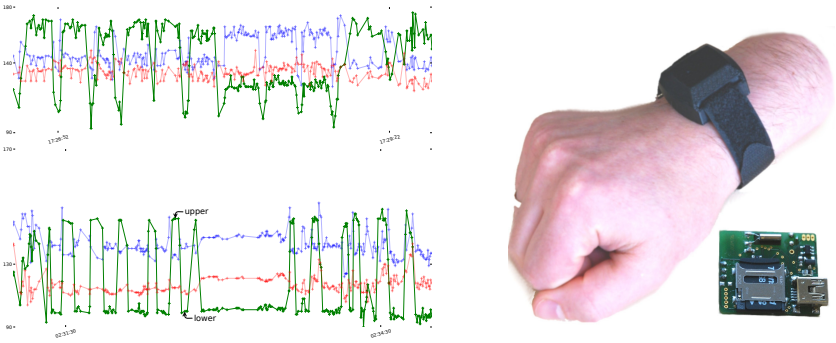


Figure 4.7: Recording setup for the accelerometer feasibility study. Participants were asked to wear the HedgeHog sensor device on their wrist, which continuously recorded acceleration data during the wake-period.

4.2.2 Accelerometer Only Identification

For this feasibility study, we asked four regular smokers (aged 26 to 40, 2 male, 2 female) to wear a wrist-acceleration logger (see Fig. 4.7) through the course several days. The monitoring time, number of total recorded data samples, covered timespan and some basic statistics on manual labeling can be found in Table 4.4. Participants were asked to double-tap the wrist-worn logging device prior to any smoking session. Smoking sessions were later marked by visual data inspection, which was simplified by this tapping indication. Participants wore the sensors on the respective dominant hand's wrist (all participant were right-handed) because we assume that this is the hand most often used to hold cigarettes. Furthermore, participants take off the sensor only occasionally, which gives a lot of background data to which we can

compare our detection algorithm.

Acceleration data was collected with the wrist-worn "Hedgehog" sensing platform prototypes (cf. Fig. 4.7). The design is based around a PIC18F microcontroller, which contains an ADXL345 acceleration sensor and a µSD-card. Acceleration (range $4g$) is continuously sampled at 100Hz and written to a FAT32 filesystem on the µSD-card in a compressed (run-length) format. Data can be retrieved by accessing the µSD-card via a standard USB mass-storage interface. The 180mA h battery included in the package can power the system for a total runtime of at least 7d without the need for recharging.

After recording, raw accelerometer data was labeled by visual inspection with the help of double-tap indicators. These double tap patterns were quite distinctive, however also often forgotten by the participants. Therefore, we also search for patterns similar to the ones depicted in Fig. 4.7: a clear hand-to-mouth gesture, repeated several times and stopped after a few minutes must have been visible to count as smoking. This manual classification was sorted into confidence classes on how strong the patterns followed this description, i.e. perfect, fair and hard patterns. Note that these classes represent the authors' confidence whether the participant was smoking and the subjective similarity to other posture patterns. While perfect means full confidence in a pattern having emerged from consuming a cigarette, fair means partial confidence due to noisy data and very limited number of repetitions and hard means that there might have been smoking but data is too noisy, the number of repetitions is limited or the pattern is highly different. Fig. 4.8 shows a pattern in each class for every participant.

As previously mentioned, a systematic review of smoking topography research [275] provides a simplistic model for smoking: for a total volume of $n = 193$ participants, which reported $17.23 \pm 6.88 \frac{cigs}{day}$, with an *inter-puff* delay of 15.46 ± 7.18s, and *puff duration* of 2.32 ± 3.3s (Method 1), the *total consumption time* can be sampled to 306.75 ± 129.01s. These numbers were retrieved with CReSS devices, a device which measures the airflow while smoking. The extracted model, however, was shown to be inconsistent (see [275] for details) but can still be used as a starting point. Detecting single puffs from acceleration would then allow to apply this model to distinguish smoking from other activities.

Finding single puffs from wrist motion follows the observation that a hand-to-mouth gesture, in its prototypical form, is split into two states: an *upper* state when the hand is kept near the mouth and a *lower*

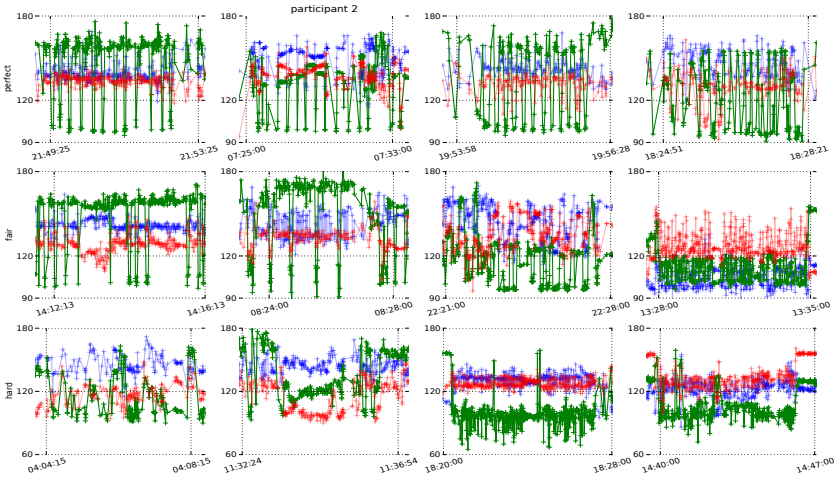


Figure 4.8: Raw X-Y-Z accelerometer data (red,green,blue). The pattern for smoking while standing is clearly visible in the top row.

state when the hand is not at the mouth. An algorithm to identify these states from acceleration data needs to address the following challenges:

Different Wearing Styles and Unfixed Sensor Position. Wrist-worn sensors can be attached in multiple positions, resulting in different axis rotations. For each style, the sensor values measured in the *upper* state will look different. However, the number of possible axis rotations is limited. For example a band with an integrated sensor, can only be rotated around the axis along the arm, without moving the wrist. A further source of complications is whether the sensor is worn on the *left* or *right* wrist, as this typically inverts the measurement. This is particular challenging as the wearing style can change while sensor data is being recorded.

Different Smoking Styles. A prototypical puff starts with the hand next to the hip, which is then moved to the mouth, kept there for a few seconds, before moving the hand back to the hip. The hand can be kept "inverted", i.e. with the palm pointing away from the mouth, instead of the other, more typical, way around. Also the wrist posture attained when not at the mouth is a source of great variety, which can be categorized as follows: Smokers might choose to keep the hand still at the hip (the Prototype), gesticulate while talking (the Socializer), rest the hand at the forehead (the Thinker), or on some flat area (the

participant	timespan	#patterns	duration	#samples
0 (male)	8 d	35 (2,7,26)	4.6 min	3.08M
2 (male)	5 d	28 (1,6,21)	6.8 min	3.57M
1 (female)	5 d	34 (11,8,15)	8.1 min	4.58M
3 (female)	5 d	19 (10,3,6)	8.7 min	1.53M

Table 4.4: Summary description of collected data. The number of total smoking gesture patterns (number of "hard", "fair" and "perfect" samples in brackets), the mean duration of those gestures and the number of total accelerometer sample points. Note that the last figure can be misleading as sample points are only recorded when subsequent measured values changed (using run-length compression), not representing the equidistant sampling points.

Casual), might just flip their wrist while keeping the arm upright (the Sophisticated), be lying (the Relaxed), switching between dominant and non-dominant hand (the Switcher), or resting the whole arm and moving only horizontally (the Leaner). All of these smoking styles have a strong influence on the measurement that can be taken on the wrist, especially for the *lower* state.

Superposition of Other Activities. Concurrent activities, executed while smoking, are another challenge that a detection algorithm needs to tackle. Standing, Sitting, Walking, cycling, or driving (a car) are a few examples which have a direct influence on the measured inertial wrist motion, i.e.˜when the whole body is in motion, this influence also shows on the wrist motion. For example, walking super-imposes a regular step pattern that hinders the detection of *upper* and *lower* states.

Confounding Activities. Motion that contains hand-to-mouth or similar looking gestures are another challenge. Eating, Drinking are prime examples thereof, but also manual work like pipetting do look similar when seen through the lens of a wrist-worn accelerometer.

Availability of Reliable Ground-Truth Data. For limited study settings, for example under laboratory condition, ground truth labels are a lot easier to get than under free-living conditions. Either the participant needs to be shadowed, which is very expensive, or the participants needs to be asked for self-reports. In both cases, smoking events could be recorded but mis-labeled, or the behaviour of the participants might be changed, as he becomes aware of him being observed. Labelling, and with that gathering reliable data, can present

participant	lower	upper
0 (male)	$-8.42 \pm .09$	$9.43 \pm .19$
1 (female)	$-8.42 \pm .08$	$4.29 \pm .25$
2 (male)	$-8.05 \pm .1$	$9.69 \pm .21$
3 (female)	$-9.21 \pm .11$	$5.46 \pm .28$
	$\mathbf{-8.51 \pm .09}$	$\mathbf{7.22 \pm .23}$

Table 4.5: Empirically determined mean and standard deviation of the "lower" and "upper" states of the participants, for "perfect" smoking patterns. Units are in $\mathrm{m\,s^{-2}}$. What is clearly visible is the limited amount of data, as only the Prototype style is included, and that participant's wrists are not always pointing straight up while smoking.

a great burden. Active Learning approaches, were participants are continuously asked whether they just had a cigarette might be another way to gather ground-truth. For example, a high-recall low-precision recognizer could potentially limit the amount of required user feedback.

While thinking of *prototypical* smoking gestures the actual detection might seem trivial. Upon closer inspection, it becomes clear that smoking exhibits a lot more variability, and that prototypical gestures are not the major group of gestures. Hence, for properly detecting smoking gesture, a dataset which captures a large quantity of different styles is needed. Sample signals of the aforementioned challenges can be seen in Fig. 4.8. All "perfect" (first row, Fig. 4.8) samples were recorded from a *standing Prototype* style. A noticeable difference is visible when a participant is *sitting* (middle row, Fig. 4.8), where the smoking style shows a higher variety. In particular, the Switcher style (the tendency to switch hands while smoking), a rotated attachment and the Casual are visible. An example of the effect of different attachment styles can be seen in Fig. 4.7, which shows sensor data recorded when the sensor was worn on the top and the bottom of the wrist.

A basic classifier can be deduced, however only addressing a partial number of these challenges and smoking styles, namely the Prototype. The cigarette-to-mouth gesture or puff can be split into two postures, which we call the *upper* and *lower* posture. Limiting only to one axis, the one along the arm, allows to extract rotation of the wrist along the body. We selected the mean and variance on this axis, i.e. the Gaussian, of these two postures as the feature to classify by, and manually extracted those from one "perfect" pattern of each participant. The numerical

result of this can be found in Table 4.5. We then combined these four Gaussians into two single cross-participant Gaussians representing the "upper" and "lower" posture.

We now want to find occurrences of the Gaussians in the complete dataset. An adaptive segmentation, based on the variance of incoming acceleration data is used. Accelerometer data is added element-wise to this buffer, until the calculated standard deviation is greater than half of one of the pre-determined Gaussians standard deviation. In which case we record the deviation between the calculated mean and pre-determined Gaussian mean, empty the buffer and continue with the rest of the accelerometer data. Applying this algorithm for both the "upper" and "lower" Gaussians, results in two lists of deviations between the accelerometer data and pre-determined Gaussians. Summing up those lists over a fixed time window of roughly 5.4s, i.e. the mean length of two subsequent cigarette-to-mouth gestures [7], results in the similarity score we used to identify the gesture. After applying an empirically determined threshold to this summed list, we were able to identify time windows where participants had a cigarette. Because participants also tended to change the hand which holds the cigarette, we furthermore merged identified windows which were separated but do not span more than 4-8min, the mean time to consume a cigarette. To account for simple concurrent activities, like walking or cycling, a low-pass filter is first applied to the data. Any motion that is executed faster than 5Hz is filtered out to remove very quick motions which are likely not related to smoking. A carefully chosen band-pass filter based on the human stride frequency could provide improved results.

Table 4.6 shows the results of this automatic classification compared to our manual labeling. What is visible there is the precision ratio of the classification, i.e. how many automatic classifications match our manual labeling and how many do not, as well as the hit-ratio, which describes the number of matches of automatic classifications in each class of manually labelled data. In total 116 episodes of cigarette-smoking were monitored, of which for all but one participant more than 55% could be identified with automatic classification. This is a promising result, since this is achieved by straight-forward thresholding and by Gaussian modeling of the "upper" and "lower" posture states before and after the cigarette-to- mouth gestures. Further analysis of actual gesture data, as well as higher-level models of sequences of posture changes might in combination with this method attain better classification results. The proposed algorithm is however both fast and has a small enough

	positives (precision)		hit-ratio (recall)		
	true	false	hard	fair	perfect
0	56.4%	43.6%	0.0%	14.3%	48.6%
1	61.8%	38.2%	63.6%	50.0%	73.3%
2	69.2%	30.8%	100.%	16.7%	76.2%
3	17.4%	82.6%	30.%	0%	16.7%
	51.2%	48.8%	48.8%	20.2%	53.7%

Table 4.6: Detection score of the basic classifier. Positives are calculated as the ratio between total number of automatically identified occurrences and the ones which matched the manual labelled ground-truth (true positives) and ones which did not match (false positives). The hit-ratio is the number of matches between manually labelled occurrences and automatically identified occurrences.

footprint, so that it could be implemented on the sensor and act in an on-line fashion, i.e., on the streaming sensor data. The algorithm can be summarized into these steps:

1. low-pass filter accelerometer data with a cut-off frequency of 5Hz.
2. split data into regions of varying length where the standard deviation is smaller than the thresholds of Table 4.5.
3. calculate the deviation of the mean of the regions and the ones of Table 4.5.
4. sum up the deviations with a fixed-time window of roughly 5.4s.
5. record the timestamps when the sum of deviations rises and falls below pre-determined threshold.
6. determine smoking by the number of rise and fall times during a 4-8min window.

Several things should be noted when interpreting the results of this algorithm, which is presented in Table 4.6. First of all, only approximate ground truth is used, which was gathered by letting the participants double-tap the sensor prior to, during, or at the end of having a cigarette. We then manually labelled the timespan in which we could identify a pattern which we deemed to result from a cigarette-to-mouth gesture. While the probability that this gives us a wrong label is low (since we have been looking for repeating patterns in the whole dataset) the probability that we missed a similar pattern is inevitably higher. Often, the participants simply forgot to double-tap the sensor, or the pattern is

just not similar enough to the ones we identified beforehand. The true number of false positives is thus likely more optimistic than reported here, as the classifier indeed identified the cigarette-to-mouth gesture correctly but our manual labeling was too conservative.

Furthermore, for this feasibility study, we concentrated on a single frequently occurring cigarette-to-mouth gesture. While the accelerometer pattern that results from this is prominent in the data for all participants, it does show an interesting variation over different days. At most times the axis along the arm is influenced the most, while the others are quasi-static. This is only observable as long as the sensor is worn tight on the wrist. This also shows in the data after the participant gets up and re-attaches the sensor in the morning, when the whole dataset shows then a different "smoking"-pattern. This also explains the low number of recall and precision in the dataset of participant 3, which tended to wear the sensor in a loose way that made it harder to recognize our identified pattern with this basic classifier.

The proposed classifier is based on a number of assumptions regarding the cigarette-to-mouth gesture, which could hold only in specific cases. We assumed that the participants were smoking while standing still and moving their dominant hand between their mouth and a lower position. This is of course only one specific gesture smokers tend to exhibit, others for example might prefer to smoke while moving or walking, which would also result in a different accelerometer pattern. Another assumption that this classifier builds on is that a cigarette is usually smoked in a time-frame of 4-8min.

Certain cigarettes or cigars might however cause different smoking times. It is finally important to stress that the dataset for this study is a realistic one. It was recorded in an unobtrusive manner with the participants reporting being unaware of wearing the sensor for most of the time. Furthermore, participants wore the sensor during their entire wake-phase which gives a large amount of background data to assess the possible confusion with other activities, for example eating or drinking. Those might exhibit similar hand-to-mouth gestures and postures, which could explain the rather high false-positive rate. Because of the way we obtained the ground-truth data, we are unable to assert this. However, compared to a study under laboratory condition our data can be expected to be highly naturalistic, since we used an unobtrusive sensor that has been worn over the course of several days.

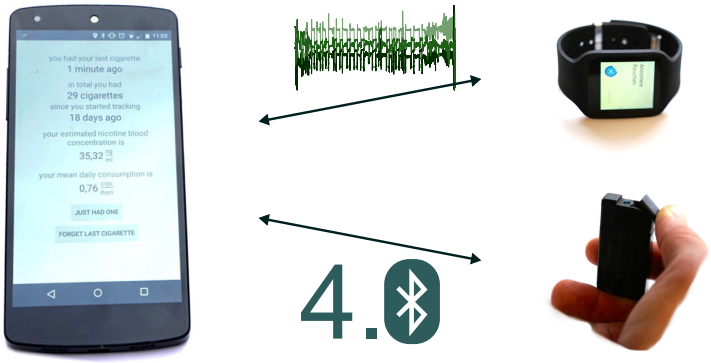


Figure 4.9: The detection system handed out to participants. A Smartphone application consolidates smoking instances from the Smartlighter, as well as wrist motion data from a Smartwatch. The user is presented with basic statistics about his or her behaviour.

4.2.3 Generalized Symbolic Detection

While the last chapter provided insight into the principal feasibility, it only used accelerometer data. A generalized classifier, based on similar assumptions, which is independent of the sensor used for estimating the arm's orientation is presented here. Basing such a classifier on quaternions as input allows for this generalization, as these can be estimated from any number of inertial sensor input. Intuitively, such a classifier can be designed on three observations about smoking gestures: (1) a smoking session consists of consecutive puffs characterized by smoking topography, (2) the device's orientation on the wrist is known, and (3) puffs can be identified by detecting hand-to-mouth gestures.

These observations are tested on a dataset consisting of 6 participants, which encompasses 351 smoking instances in total. Data was recorded in-the-wild with an ensemble of a Smartphone, Smartwatch and Smartlighter as depicted in Fig. 4.9. Other study setups require either shadowing of the smoker or rely on self-reports. Such ground-truth gathering can lead to non-natural behaviour [131, 284], or reporting biases [285, 286, 287]. Comparing solely to events detected by the Smartlighter increases the validity of the collected dataset by avoiding these detrimental effects.

The first step of detecting smoking sessions from consecutive puffs

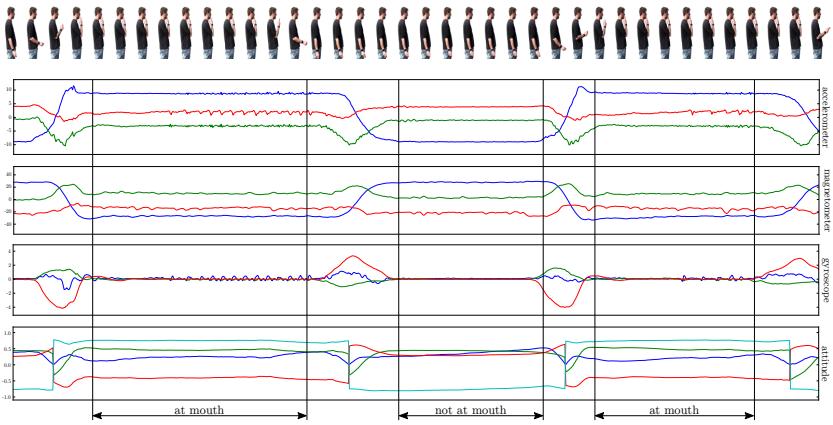


Figure 4.10: Prototypical wrist motion as measured through different inertial wrist-worn sensors.

identified in continuous data recordings is visible in virtually all related work on wearable smoking detection. Rather complex models, like Conditional Random Fields [29] were employed, as well as simplistic threshold models [182, 276]. Here, we opt for a simple model that is derived from smoking topography [275] (instead of the dataset itself): a volume of $n = 193$ participants reported $17.23 \pm 6.88 \frac{cigs}{day}$, with an inter-puff delay of 15.46 ± 7.18s, and a puff duration of 2.32 ± 3.3s, which yields an overall consumption time of 306.75 ± 129.01s. In the following, we treat this topography as constants and threshold the detected values with those.

Prior to applying this topography, in any form, puffs need to be detected from wrist motion (cf. Fig. 4.10). Any estimation of the arm's motion in three-dimensional space can be expressed as a sequence of quaternions $q_t \in H$. The motion can either be estimated from acceleration-only signals, or from the full set of inertial motion sensors. As explained earlier, the influence of different wearing styles of the watch can be removed by applying the estimated rotation to a fixed reference vector:

$$v_t(q_t) = \begin{cases} q_t^T(-1,0,0)q_t, & \text{if } left \text{ and } rot \\ q_t^T(-1,0,0)q_t, & \text{if } right \text{ and } \neg rot \\ q_t^T(1,0,0)q_t, & \text{if } left \text{ and } \neg rot \\ q_t^T(1,0,0)q_t, & \text{if } right \text{ and } rot \end{cases} \tag{4.1}$$

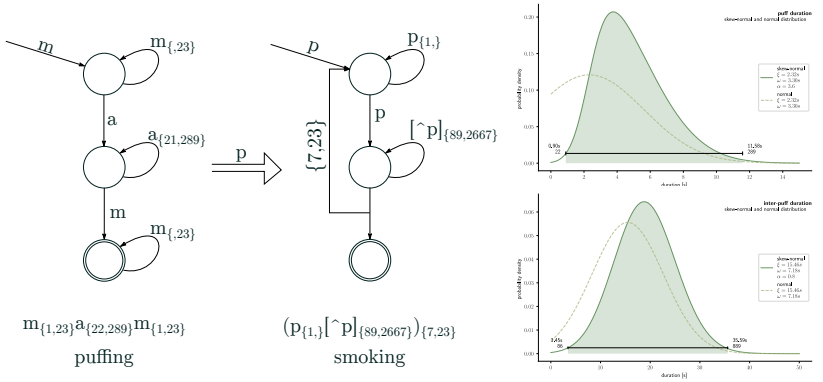


Figure 4.11: Regular expressions and respective state machine for symbolic smoking detection. Symbols are generated from wrist motion according to Equation 4.2. Symbol repetitions are chosen according to the smoking topography skew-normal distributions in multiples of 25Hz.

which then results in an estimated vector $v_t \in R^3$ that captures the current wrist's attitude at time t. The reference vector changes if either worn on the left and rotated, or on the right and not rotated, in all other cases if does not need to be changed. This transformation generalizes from the actual sensor set that was used for estimation. Furthermore, it provides the wrist's attitude in a global frame of reference, that is independent of the attachment style on the arm.

With the wrist's attitude expressed as a vector v_t, we can encode a further assumption about puffs: *a puff is imminent when the wrist is pointing towards the sky*. This condition is not only fulfilled while taking a puff from a cigarette, but for other activities like eating, drinking, crossing the arms, scratching one's head, etc. as well. However, due to the number of required puff repetitions and the particular timing, smoking might still be discernible from other con-founding activities. Since we can express the arm's attitude in vector form, a threshold on the axis that points towards the sky is sufficient to express this condition. In our case, this is the z-axis (cf. Fig. 4.6), which spans both the sagittal and frontal (coronal) plane.

Another observation about the puff gestures is that a *quick movement is executed to bring the hand to and from the mouth*. This can be captured by calculating the angle between consecutive arm attitudes, i.e. we

let $r_t = cos(\frac{v_{t-1} \cdot v_t}{|vt-1||v_t|})^{-1}$. Note that this is related to the rate-of-turn measured by a gyroscope, however r_t is drift-stabilized. Furthermore, estimating the rate-of-turn from attitude vectors also allows the rate-of-turn to be estimated from non-gyroscope sensor readings. Given r_t, it is possible to detect quick movement by yet another threshold t_r.

These quick movements enclose an interval where a puff was probably executed. If a puff was executed, the interval defined by two such spots corresponds to the *puff duration*. Additionally, if the wrist was at the mouth, this might actually be a puff. Given this, we can also define the *inter-puff* interval as the time between two such consecutive puff candidates.

The above description lends itself to model smoking as a finite-state automata. Durations, elicited from smoking topography, can be encoded as repetitions into the states of such an automata. This is similar to detecting gestures by transforming sensor data into strings [288]. Such a model is comparatively simple and does not require any dataset for parameter estimation. First, we define the input alphabet $A = \{a, p, m, s, \_\}$, where a designates the wrist being (presumably) near the mouth, m an ongoing wrist motion near the mouth, p identified puff candidates, s when smoking, and $\_$ for all other conditions. As the first step, the input vectors v_t and r_t are converted into symbols according to:

$$a_t \in \{m, a, \_\} = \begin{cases} m = \text{moving} & \text{if } r_t > t_r \text{ and } v_{t,z} > t_z \text{ else} \\ a = \text{atmouth} & \text{if } v_{t,z} > t_z \\ \_ = \text{don't care} & \text{otherwise} \end{cases} \tag{4.2}$$

where t_r and t_z are empirically derived thresholds. The input alphabet encodes if a movement near the mouth is currently observed (rotation $r_t > t_r$), or if the wrist is at rest near the mouth (Z-component of attitude vector $v_{t,z} > t_z$), or not matching any of these conditions. The time-dependent smoking topography parameters are then expressed in terms of repetition of these symbols. A finite-state machine, i.e. a regular expression on this alphabet as depicted in Fig. 4.11 can then detect smoking instances. The smoking topography durations are reported as normal distributions, i.e. mean and standard deviation are given. By deliberately selecting the .99-significance interval on the probabilistic distributions for each duration, hard repetition boundaries are extracted. However, the standard deviation is larger than the mean, so we assume a skew-normal distribution with an estimated skew value. Fig. 4.11 presents the chosen values.

The detection automata are encoded in a two-layered approach. First, the wrist attitude sampled at a fixed rate is transformed into symbols according to Equation 4.2. Next, puffs are detected according to their duration: all matches of the regular expression $m_{\{2,45\}}a_{\{40,578\}}m_{\{1,45\}}$ are replaced by the p symbol. This is the first layer, afterwards all matches of $(p_{\{1,\}}[^{\frown}p]_{\{310,5337\}})_{\{7,23\}}$ detect smoking. The first expression encodes the *puff duration*, the second encodes the *inter-puff* duration. Explained in plain English, the first expression requires the wrist being at the mouth for at least and up to *puff* duration (in number of sensor frames) and surrounded by short sequences of strong motions. Likewise the second expression matches sequences of puffs that were repeated at least 7 times, and where no puffs were detected for at least and up to *inter-puff* duration.

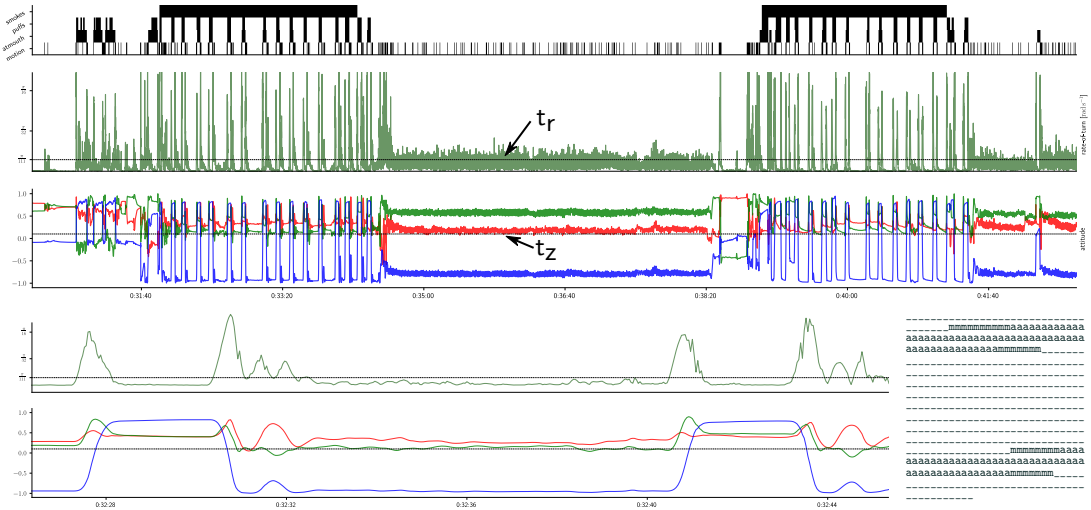


Figure 4.12: Symbolic representation of smoking instances and single puffs. The bottom row presents puffs transformed into a symbol stream by means of Equation 4.2, the top row presents the result of applying the finite automata applied to the generated symbol stream and the resulting smoking detection.

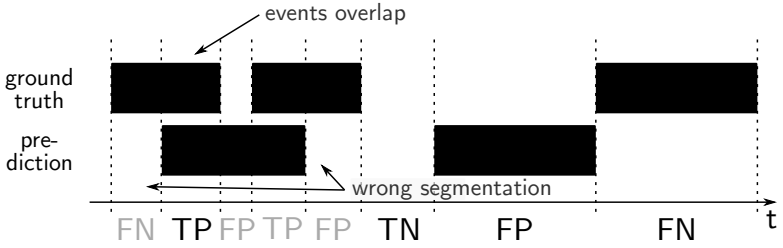


Figure 4.13: Definition of event detection error. Any overlap is counted as a detected event (TP). Segmentation errors (in grey), i.e. incomplete overlaps are ignored as they neither influence the actual recognition task nor inform the design of the recognizer.

Fig. 4.12 depicts this approach graphically. In the bottom row a single puff transformed into a stream of symbols is shown. This encodes our original assumption that a puff is executed after a quick wrist movement, and then staying at the mouth for several seconds. The symbol stream on the bottom right is the target sequence that is detected by the finite state machine described in Fig. 4.11. The results of this recognition are shown for two smoking instances on the top plot. Only sequences of several hand-to-mouth gestures (puffs) are detected with this approach, and no time-based segmentation is required.

For smoking detection, the number of events that are correctly classified as smoking, and correctly rejected as non-smoking are most interesting. Hence we define the following errors for this classification approach, also depicted in Fig. 4.13:

True Positive (TP) user was smoking and was detected.
False Positive (FP) user was NOT smoking and was detected.
False Negative (FN) user was smoking and was NOT detected
True Negative (TN) user was NOT smoking and was NOT detected

detected refers to the condition when a smoking event is emitted which overlaps with a smoking instance that is present in the ground-truth. This can also lead to conditions where smoking was detected too late or too early (c.f. Fig. 4.13). However, these cases are ignored as the exact beginning or end of smoking sessions are not important. It is enough to correctly detect the presence of a smoking session, hence we ignore *segmentation errors*.

	smoking	NULL			smoking	NULL	
smoking	47	5	90%	77%	95	28	smoking
NULL	12	59			165	278	NULL
	79%	8%			37%	9%	

Table 4.7: Confusion matrix for a dataset of large variety containing 6 participants (right hand), and a selection of less variety with 4 participants (left hand). On the bottom line the recall and False Positive Rate is shown, the rows in the middle show the precision.

One measure for this smoking classification is its *recall* (also known as Sensitivity or True Positive Rate) . This is equivalent to the probability $P(detected|smoking)$, i.e. smoking is detected given that the user was smoking. This measure should be as high as possible to not miss any smoking events, while allowing for occasional false alarms. The amount of false alarms can be quantified in two ways: (1) as the *precision* (Positive Predictive Value) and (2) the False Positive Rate (FPR). The first one (precision) is equivalent to $P(smoking|detected)$, the probability of smoking when it was detected. This allows to answer the question at which ratio users where smoking when it was detected. The FPR in contrast indicates how often a detection would happen given that the user had not smoked $P(detected|\neg smoking)$. This is the rate at which the system wrongly labels other activities as smoking (see Table 4.6).

For the presented data recordings (351 instances, 6 smokers) a precision of 77% at a recall of 37% could be reached (9% FPR). Background data (not smoking) and smoking data was balanced to 50% each. The results show that the assumption that smoking can be modeled as a fixed sequence is too simple and does not capture the full diversity of different smoking styles. However, the presented approach can be used to prove that enough diversity was captured during data collection to avoid overly optimistic results. For example, when removing just two participants from the above mentioned dataset, we can test the remaining 123 smoking instances. This than leads to a precision of 90%, a recall score of 79% and an FPR of 8%. Scores that are comparable to the currently best recorded recognition systems.

These results should be interpreted in the light of possible applications: (a) providing an exact cigarette counter, (b) providing cessation information on a mobile while smoking, and (c) tailoring of interactive information. For (a) the score clearly shows that the offset of real

and estimated number of smokes after a short time period will be too far apart for a user to build trust into the system. The FPR rate of $8 - 9\%$ indicates that when splitting the day into 10-minute interval about $24 * \frac{60}{10} * .08 = 11.52$ misreports per day are probable. Such over-reported numbers are in contrast to the psychological bias [289], which a smoker would then catch quickly. In the case of providing "just-in-time" (b) intervention material, this would be less problematic, as the smoker might simply receive an occasional intervention when not smoking. Even when only detecting 37% (recall) of actual smoking events, a system that uses smoking as a cue to provide timely information might be useful. For tailoring information, the influence of information on the smoking behaviour is to be quantified. This means that a highly precise classifier would be required, the higher the probability of detecting real smoking events the more reliable decision based on these numbers can be - with a precision of 77%, such an application would be possible.

Additionally to the just mentioned applications, the presented symbolic detection approach can check that a smoking dataset does contain smoking data of enough variety. If high scores ($> 80\%$) can be achieved the dataset contains mostly smoking events of prototypical style with little confounding activities. If low scores ($< 30\%$) are returned, the dataset might contain smoking with more variety, like the Thinker, Socializer or while driving a car. The approach therefore provides a baseline to assess the quality of a dataset and subsequently the reliability of detection scores of a smoking classification system.

4.2.4 Machine Learning Detection

Established that the recorded datasets contains smoking events of high variety one might wonder how well a machine-learned model, the classical approach of Activity Recognition, would perform in detecting smoking gestures. The major differences to the previous approach are: (1) domain knowledge in form of smoking topography can not be directly modeled, but have to expressed in the dataset or the calculated features, (2) characteristics not captured by the previous approach might be automatically picked up. To show this, several learning approaches (SVM, RF), different feature sets and segmentation parameters are tested and reported here. The full combination of tested parameters is depicted in Fig. 4.14.

The annotations present in the recorded data sets needed to be adjusted for two reasons: First, participants annotated by marking only

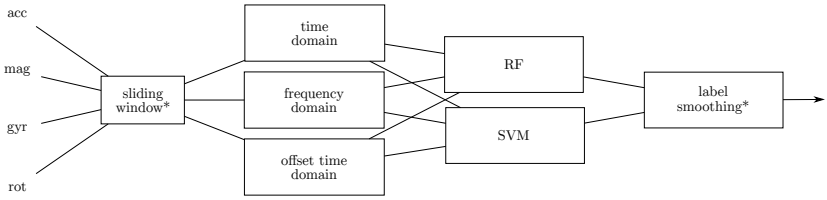


Figure 4.14: Tested parameters for the machine-learning approach. Different sensor inputs (acc, mag, gyr and rotation), are combined with different lengths of sliding windows, of which time, frequency and relative time features were extracted. These features are tested with an SVM or RF classifier, and finally different lengths of label smoothing are applied. Each path in this graph represents one tested parameter combination, nodes marked with a * provide several additional parameters.

the beginning of each smoking session, which gives only an estimate of where a smoking incident resides in the data. For training and evaluation, the annotation end was set after visual inspection of each incident. A second adjustment that was required involved moving the beginning of an annotation as well, depending on the way the participants annotated. While the recording of smoking always immediately started after the lighter was used, the annotations made on the smartphone or smartwatch often preceded the actual incident significantly. In the latter case, the beginning of each annotation was moved, again, via visual inspection of raw data, no annotations were removed.

The recording of wrist motion data was started only after an annotation had been registered: This conserves energy at the price of possibly missing the first seconds of a smoking incident, especially when annotating with the Smartlighter. While the lighter advertises itself directly when it is lit up, we have sporadically noticed that the Bluetooth LE connections between Smartphone, Smartlighter and Smartwatch introduce a delay of up to a minute. Such delays present a major trade-off, though these exist for alternative monitoring systems as well.

Intuitively, a smoking incident can be split into single puffs of a cigarette. As can be seen in Figure 2, such segments can be further characterized into three stages: raising, keeping, and lowering the wrist. This observation has been used for adaptive segmentation [29] and for motivating the choice of time-window for static segmentation [32, 178, 290, 291]. A window with an overlap of 50% and a duration of 10 -

15 secs, have been derived from smoking topography research. We followed the approach of using static time-windows. Whether or not a participant was smoking is predicted for each segment, however, we did not fix the duration upfront, but tested a set (512, 1024, 2048, 4096 samples, which is equivalent to 10.24, 20.48, 40.96, 81.92 seconds) of intervals.

Such extracted segments carry data from each sensor modality. Each of which has different power requirements: Accelerometers require the least amount of power, followed by magnetometers and gyroscopes, while fusing all these into an attitude measurement requires the most power. For efficiency reasons, we are interested in the sensor modality that provides the maximum classification score by itself. Therefore, we evaluated all modalities independently to investigate their respective influence. A hierarchical approach, where the accelerometer provides a high-recall and low-precision classification could be used to switch on less efficient but higher- precision sensors. This however is of little use in our case, since this is only useful if a subsequent classification on power-hungry modalities would provide higher precision.

The raw data of each segment is compressed with accumulating functions to decrease the computational complexity of the classification. Three feature sets are compared in this work: (1) a time domain set, which includes the mean, standard deviation, euclidean norm and root mean square of the segment, (2) a frequency domain set, which includes the center frequency, the three frequencies with maximum amplitude and spanned spectrum, and (3) an offset time domain, which includes the same features as the time domain set but relative to the first sample. The offset feature set was chosen to test whether the starting position of the wrist has a strong influence on the classification results. However, the offset feature set cannot account for different styles of wearing the Smartwatch (e.g., display below the wrist, or between the left or right wrist) and the resulting dissimilar measurements. While this challenge could be addressed by rotation-invariant features (e.g., as in [292]), we instead rely on sampling enough such examples and train the classifier for this setting. Permutations of feature sets and sensor modalities were not tested.

For classification, we compare two classifiers that are commonly used in related work: Support Vector Machines (SVMs) and Random Forests (RFs). SVMs were run with a standard radial basis function (RBF) kernel. The number of trees in the RF was limited to ten. For both classifiers, a NULL rejection threshold was additionally estimated to

pick out background data. This threshold is applied to the confidence score of each classification result to avoid training the NULL class explicitly, which would decrease the classifier's generality.

The mean consumption time of a cigarette is generally much larger than the chosen segmentation windows. To identify individual smoking instances, a label smoothing is applied after the classification procedure. Assuming that misclassification can happen sporadically, we applied a majority voting on the classified segments. The window over which this majority voting is applied presents a further parameter, which was tested on a set of one, five and ten segments. Depending on the segmentation's window size, between 10.24 - 819.2 seconds of data is classified at once. The gesture recognition system was evaluated once without a post-processing pass, and once with the different post-processing parameters applied, to explore the best detection approach and to give a realistic estimate of smoking incident detection.

In the previous subsections, all parameters and the tested values were detailed. These are summarized in Fig. 4.14, where each path through that graph represents one tested pipeline. Nodes marked with a star have additional sub-parameters (e.g., window duration for segmentation) that further increases the number of combinations. A grid search was performed over all these parameters to select the parameter combination (also called pipeline here) with the highest average F1-score. Each parameter combination was repeated 50 times on a random stratified split of the whole (cross-user) motion data. 40% of the data was used for training, with the remaining 60% for testing. In total 19200 combinations were tested. The sample order was retained[5] during the random split, only removing training samples from the original sequence. Motion data encompassed all participants, the evaluation results are therefore valid for a person-independent classification. For computational tractability, a GNU parallel [265] based cluster with 42 cores was employed. The implementation of all evaluated recognition approaches is based on a modified version of the Gesture Recognition Toolkit.

Fig. 4.15 shows the average precision and recall of a single parameter combination, evaluated with all sensor modalities combined with one of the classification algorithms for all sliding window durations, with and without post-processing. After applying label smoothing, the predictions correspond to actual smoking incidents. Before this step,

[5]Other approaches, for example mixing the sample set randomly, would invalidate the post-processing step, since it assumes a time-ordered test set.

predicted labels can correspond to the same event multiple times. As can be seen from these results, the RF classifier has a 3 - 5% higher F1-score than the SVM classifier for all parameter combinations. For a large smoothing window of ten frames and a segmentation window of 512 samples, the highest F1-score of 84% is achieved. This corresponds to a classification window duration of 102 seconds, or roughly one and a half minutes. These results show that detection of smoking instances from wrist-motion is feasible.

It is furthermore visible that measuring wrist acceleration alone is not only sufficient, it is in fact the best overall perfoming modality on several occasions. Interestingly, the best-performing feature set choice for acceleration depends on the used classifier. For RF, the time feature set performs best, while for the SVM classifier, the frequency feature set is a better choice. The choice of segmentation window size is also influenced by the choice of classifier. For SVM, a larger size of 4096 samples clearly outperforms the choice of 512 samples for RF, and vice versa. A large segmentation window on frequency features seems better suited for classification with an SVM than for RF.

Magnetometer measurements show the highest precision, after the measurements are corrected to include only relative movement instead of the absolute direction of magnetic north (i.e., the offset feature set). It thus appears that the global reference of magnetometer measurement needs to be corrected to provide useful measurements. Correcting only relative to the first sample, as we did, however proves to be inefficient: While independent of the cardinal direction when starting to smoke, changing this direction while smoking still has a large influence. A baseline correction, i.e., assuming that the cardinal direction of the body changes relatively slow compared to the direction of the wrist, would allow to apply a high-pass filter. This way, the cardinal direction of the body could be removed from the compass signal.

The challenge of global orientation in IMU signals has also been addressed in the RisQ system [29]. There, the classification was based on the attitude of the wrist expressed as quaternions with a global reference. The approach was to use the difference between two quaternions to remove the global reference, effectively transforming the quaternions into a stabilized turn-rate measurement. Even without any quaternion-specific feature set, there are instances where this modality performs comparatively well, reaching an F1-score of up to 83%. However, only when using a RF classifier, and in cases where label smoothing has been applied (cf. Figure 4.15 RF, smoothing=5).

Turn-rate measurements show the highest recall throughout all parameters. This is most likely due to the differential nature of this signal - during rest points of the smoking gestures the signal measures zero (cf. Figure 4.10). These restpoints, when the wrist is near the mouth or kept next to the body, are easily confused with other activities, however, leading to a higher recall but a lower precision. Compared to that, the absolute nature of the other modalities do not exhibits this. For example, the magnetometer measures the (globally) absolute deviation of the wrist to magnetic north - if not all possible deviations are included in the training dataset recall will accordingly be lower.

In previous works [178, 29], segmentation windows of 10 - 15 seconds for identifying single puffing instances. Our results provide further evidence that this is indeed a reasonable decision for some classifiers: The RF classifier performs best with this window length (512samples at 50Hz amount to 10.24seconds) with the time domain feature set. However, the frequency-based SVM classifier operates best with a much larger window of 81.92seconds. The choice of segmentation window is mostly independent of the smoothing window for RF. In case of the SVM, a larger smoothing window also performs better with a larger segmentation window. One can also note that different choices of segmentation window size have a stronger influence on SVM classifiers. Smoothing is finally a viable option to further increase the F1-score: In our case, the score of both classifiers is increased by 6 - 7%, to a maximum of 84%.

The authors of two related studies [29, 30] remarked that acceleration data might not be sufficient for detecting smoking from wrist motion. Our results indicate the opposite, and show that wrist acceleration is indeed sufficient. While wrist motion alone does not yield a fully-defined body model, the key to detecting smoking gestures seems not only its prototypical movement pattern but also its repetition. Additionally, smoking is mostly executed on the vertical body planes [6], on which movement can be measured with an accelerometer alone. Additionally, an accelerometer is the most power-efficient inertial sensor currently available. The idea of using a hierarchical filter, in which an efficient high-recall classifier is used to switch on sensors that provide a high-precision classification in a second step, might not be necessary.

Compared to the previously presented symbolic detection approach the machine-learning approach can provide increased performance. For the best-performing combination (F1/recall/Prec=84%, RF, smooth-

[6] The coronal and sagittal planes.

ing=10, time features, 4096 segment size), the recall was increased by 47% and precision by 7%. This shows that applying even a non-specific (in terms of engineered features) machine learning approach can results in a smoking detection system. However, these results should be interpreted in the light of a still limited dataset that might not be fully representative of a real smoker population. For this, a larger dataset would need to be recorded. With the current system missing about 16% of smoking incidents is probable, while 84% of the detected smoking instances were really smoking.

4.2.5 Alternative Sensors

While wrist-worn motion, and the instrumented lighter already provide practical insights into personal smoking behaviour, other sensor modalities that correlate to smoking could be used too. In related work the following additional sensors were presented: mobile phlethysmography [181], audio-based lighter usage detection [182], neck-worn body audio breathing [182], RF-proximity of necklace and wrist device [30], wrist-worn gas-sensors [184] and wifi-infrastructure [34].

A barometer which detects height changes of the wrist in the range of meters might provide an additional insights. However, only for the Prototype smoking style, i.e. where there are strong changes around the vertical body planes and only when the smoker is not moving, as his movement might change surrounding air pressure.

Jaw muscle contractions might also be indicative for deep inhalations during smoking. These be picked up by head-worn Electromyography (EMG) sensors [7], or by a infrared distance sensor mounted in the ear [37]. Alternatively, the wrist-worn motion detection could be extended with a reference measurement at the head to further distinguish the global magnetic orientation of the wrist with respect to the mouth. Furthermore the RF-distance could be recorded with such a device. A smoker's pulse also increases while smoking. Furthermore gas sensors mounted near the mouth could provide another modality. A more intrusive measurement device would periodically measure the blood cotinine levels, similar to devices for continuous blood sugar estimation [293], which would allow to measure other biomarkers as well.

[7] for example the JINS MEME!: https://jins-meme.com/en/

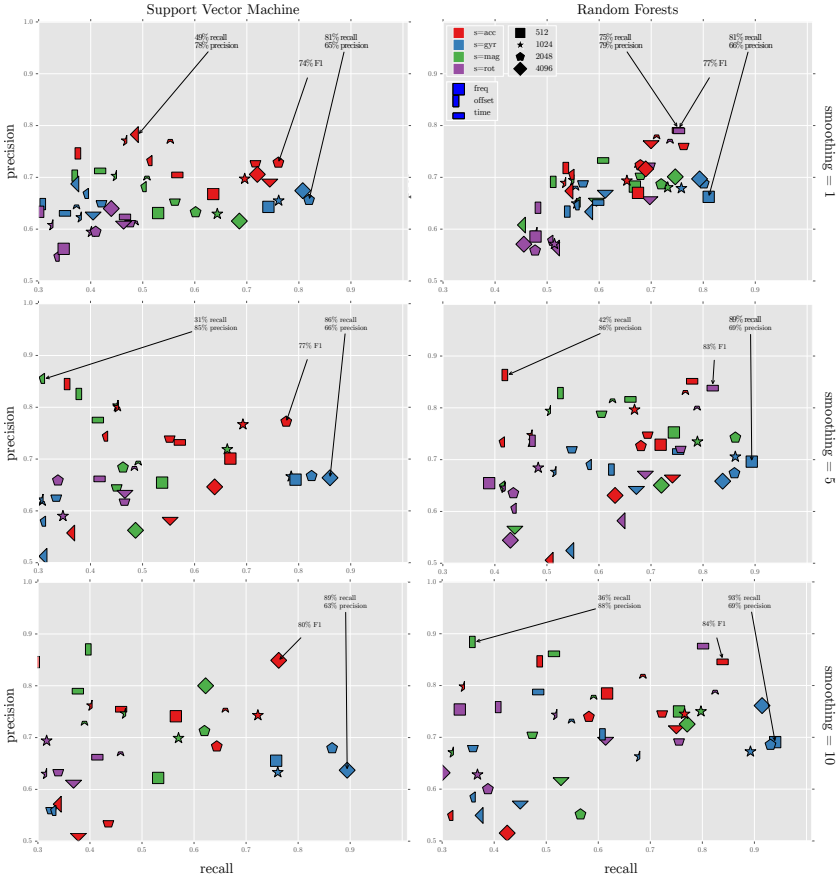


Figure 4.15: Mean precision and recall scores for the RF and SVM classifier. Four parameter were varied: (1) the sliding-window size (time is given in number of 50Hz samples) (2) the extracted features including time domain features, time domain features relative to the first sample in the window (offset) and frequency domain features (3) the sensor modality and (4) a smoothing of 1, 5, and 10 samples was applied. Each parameter combination was tested in a 50-times random stratified split over the whole dataset.

4.3 WRIST-MOTION VS. INSTRUMENTED LIGHTER

When comparing a wrist-motion smoking detection with an instrumented artifact there are two main dimensions to compare on: detection accuracy and energy consumption. For the first, the former chapters gave an indication where we can assume that the lighter provides a voluntarily gold-standard for smoking detection. Wrist-Motion detection is challenged by confounding activities, and by a sampling problem for the large variety of different smoking styles. For now, we can assume that about 84% (recall rate of ML classifier) of smoking instances can be detected, probably less in unconstrained conditions. However, a wrist-motion detector might be applied post-recording - for smoking detection it would be enough to simply collect *motion data continuously* and apply a detector later. An instrumented artifact needs to be deployed before smoking can be sampled. Another difference is the detection delay, the lighter detects smoking at its onset. A motion detection algorithm requires at least a few minutes of data before predicting a possible smoking incident, hence providing an event earliest at the middle of smoking.

Another comparison dimension is the required energy for actually doing the smoking detection. For the instrumented lighter Table 4.2 lists the required electrical energy per-day: 12.39J. This is the energy required to detect the lighting event, store the event in memory and communicate this to a connected Smartphone via Bluetooth LE. A possible detection algorithm running on a core connected to inertial motion sensor would be required to run sampling and detection within this energy budget (disregarding any additional costs of communicating the detection to other systems). Table 4.8 lists the energy requirements for sampling the sensors of two current IMUs. One can see, that just the sampling process of the full set of sensor requires a magnitude more energy. However, a 24h sample of acceleration (TDK acc) is on par with the consumed energy of the instrumented lighter. Comparatively, 1.7J could be used for communication or computation. Bluetooth LE [8] requires $29.6\mu J\,bit^{-1}$: disregarding energy required for a controller, 7179B could be transmitted. The same micro-controller, containing an ARM Mo-core, running at 24MHz, draws 6.6mW, i.e. can be powered for 257s per day with this energy budget. For implementing a detection algorithm, this is quite challenging.

[8] Assuming non-acknowledged transmission at 8mA TX consumption, and $.27bit\,s^{-1}$, see [296].

sensor	µA @100 Hz	µW @100 Hz	J@ 24h
Bosch acc	180@2.4V	432	37.3
Bosch gyr	850@2.4V	2040	176.26
Bosch mag	660@2.4V	1584	136.86
TDK acc	68.9@1.8V	124	10.71
TDK gyr	1230@1.8V	2214	191.29
TDK mag	90@1.8V	162	13.99

Table 4.8: Energy consumption of a Bosch BMX160 [294] and a TDK ICM-20948 [295] 9-axis IMU sensors. Values are given at typical operation condition as supplied in datasheets. To keep the discussion concise, the sensors are assumed to be sampled continuously.

4.4 SENSOR-ASSESSED VS. EMA SMOKING

This section details how the prototypes were deployed and used in an experiment by 11 participants, providing data covering a combined timespan of about 2800 hours. The first two Smartlighter prototypes were additionally evaluated during those user studies. The 11 voluntarily participating smokers were recruited from the University of Darmstadt, and were mostly members of the university, Table 4.9 summarizes ethnographic data of these participants. Three participants (number 8-10) were asked to use version 1 (coil-based lighter), while eight others (number 0-7) were asked to use version 2 (gas lighter) of the Smartlighter. Each participant was asked to use the lighter exclusively for 4 days and could afterwards decide to continue its usage. All were aware that their cigarette consumption is being monitored by informing them at the beginning of the study that the lighter is logging when it is being used and by providing feedback of the prototype working through the indication LEDs during operation.

Pre- (Table 4.11) and post-study surveys (Table 4.11) were conducted to elicit an estimate of the smoker's cigarette consumption awareness and a subjective opinion of the overall system. The questionnaires contain a number of statements that smokers were asked to grade along a 5-level Likert scale on agreement (agree strongly, agree, agree somewhat, disagree or strongly disagree). Results in Table 4.11 are the normalized mean and standard deviation of those assessments. Participants did not access their recordings during the course of the study, but were exposed to a statistical summary (see Figure 4.16) before

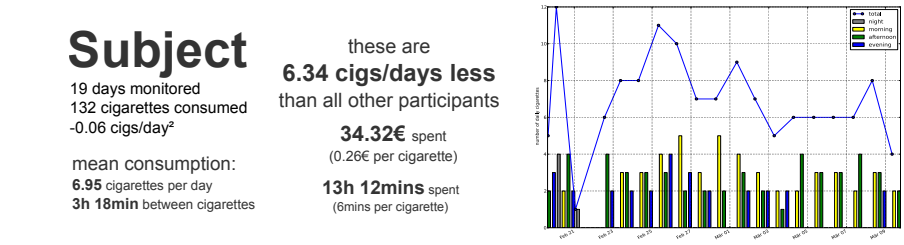


Figure 4.16: Example report generated for the study participants. The plot on the right side shows the amount of daily smoked cigarettes on four different times of the day. To the left are personalized smoking statistics as captured by the Smartlighter.

Number of study participants	11
Average age of participants (in years)	34.53 ± 12.14
Number of (reported) cigarettes per day	15.11 ± 5.95
Average years of cigarette consumption	13.03 ± 6.54
Average days of participation	11.36 ± 8.15

Table 4.9: Summary of the study participants' smoking habits.

answering the post-questionnaire. The gathered data provides the basis of the following findings below.

Hypothesis I: Smokers overestimate their daily consumption. One of the basic metrics of cigarette consumption is the total number of smoked cigarettes over the course of a day. A comparison of this self-reported and measured number of smoking incidents is given in Table 4.10. The self-reported number of incidents was extrapolated by multiplying the self-reported daily consumption by the timespan of participation. It is a dominant trend that the participants overestimated their daily consumption compared to the number of measured incidents. This can be attributed to a number of different factors: First of all, the gathering process might not always have worked reliably and some incidents might have been missed. This effect is also visible in the post-study questionnaire results (question 10, Table 4.10), hinting towards an unreliability of some prototypes. While the lighters needed regular maintenance during the study, this effect should be especially strong for those with short participation time. However, there are participants (number 0 and 7) which are quite near to their estimation and the

	number of incidents		Morning		Afternoon		Evening		normalized	
	estimated	measured	estimated	measured	estimated	measured	estimated	measured	mean/std.dev.	daily recurrence
0 (4 days)	36	28	2.10	0.50	*3.90*	**4.00**	2.95	2.50		
1 (27 days)	324	227	1.78	2.89	1.78	**3.81**	*8.45*	1.70		
2 (13 days)	260	88	*9.05*	1.54	9.05	**3.69**	6.79	1.54		
3 (26 days)	312	173	*4.36*	2.58	4.36	**3.19**	3.27	0.88		
4 (14 days)	77	50	1.15	0.57	2.09	**1.71**	*2.26*	1.21		
5 (12 days)	228	85	4.39	2.17	*7.54*	**4.25**	7.07	0.67		
6 (6 days)	120	54	7.58	1.00	2.07	1.50	*10.35*	**6.50**		
7 (4 days)	48	46	*6.30*	3.50	4.64	**5.25**	4.64	2.75		
8 (4 days)	80	11	*7.10*	0.25	7.10	**2.50**	5.68	0.00		
9 (3 days)	60	31	*7.61*	2.33	5.07	**6.00**	7.32	2.00		
10 (12 days)	180	72	4.57	0.08	*6.52*	**3.67**	3.91	2.25		

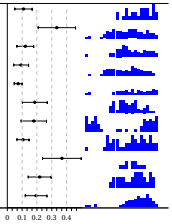


Table 4.10: Estimated and measured (via the Smartlighter) cigarette consumption figures for all participants. Time of day (Morning, Afternoon, Evening) has been extracted from the pre-study questionnaire (Table 4.11). The standard deviation of the absolute difference between normalized (over total per-participant cigarette consumption) estimated and measured consumption shows that only some users were able to estimate their main consumption time of the day. The plots to the right show the daily smoking patterns per user.

estimation difference also varies from large to small differences for other participants. It is therefore more likely that smokers find it hard to estimate their average daily consumption.

Since we compared the daily average of consumed cigarettes as measured by the lighter to the extrapolation of single estimate, there is also another plausible explanation: the strong difference of consumed cigarettes might stem from an unawareness of daily variances in behaviour, these variances are captured by the lighters but not by the extrapolation of the single self-report average consumption. To gain a deeper insight into this effect, we distributed the self-report measure over three times of the day (morning, midday and evening) by weighting the total number of cigarettes with the help of the questionnaire results (Table 4.11). Morning is attributed to question 4, 6 and 9, midday to question 2, and evening to question 1, 3, 16 and 20. Smoking incidents measured by the lighter are attributed to same time-of-days and averaged out over the participation time. This results in an estimated and measured average consumption number on time-per-day basis and can be seen in Table 4.10. The maxima of the measured per time-of-day consumption are highlighted in bold, while the maxima of the estimated consumption are in italic. Furthermore, the mean and standard deviation of the normalized absolute difference are depicted as well. This difference represents a comparison of the data gathered through

the questionnaire and by the Smartlighter. Standard deviation accounts for the fit of measured time-of-day consumption to the estimated one. The smaller the standard deviation, the more cigarettes have been smoked at the time-of-day extracted from the questionnaire. The mean value of this difference depicts the fluctuation of daily consumption, i.e. a larger mean value signifies more deviations from the estimated daily consumption on day-to-day basis, which presents another reason why smokers might find it hard to estimate their daily consumption. Taking another look at the table one can see that participants 7 and 4 do have a good idea about their time-of-day consumption. Overall, it emerged that participants found it hard to gauge their average daily consumption and the usual times when they are smoking.

Detecting daily recurrences, in order to forecast smoking incidents, could be improved by taking further contextual data into account. Separating the week into work- and non-work days and calculating the likeliness on these time-spans could improve the accuracy as a lot of participants smoke during work (Table 4.11 question 2). Additionally using location or activity recognition sensors could give an even more detailed view on the factors that could cause smoking incidents. This finer-grained context information would further improve the users' experiences.

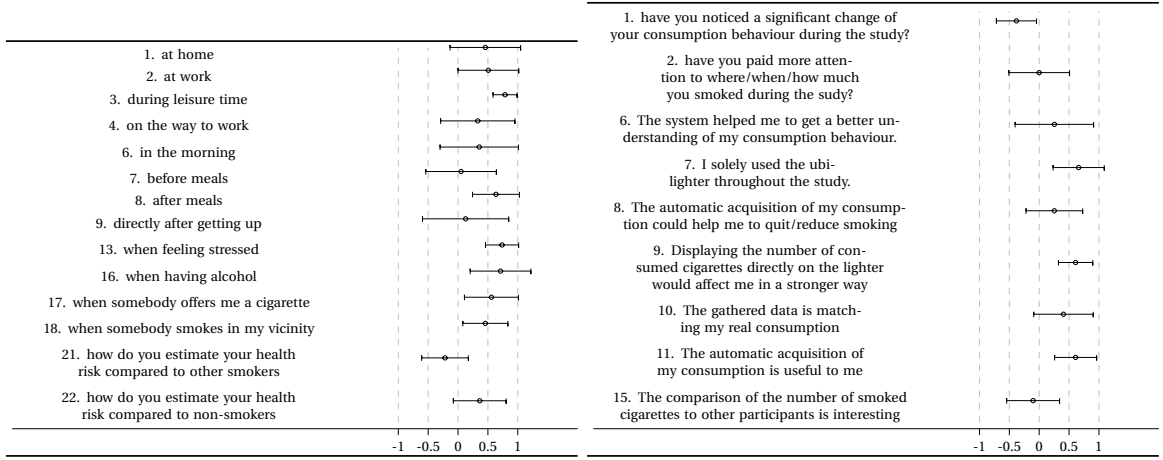


Table 4.11: Pre- and Post-study questionnaire results on smoking self-awareness. Participants graded statements via a five-level likert-scale from ″definitely not applicable″ (-1) to ″strongy applicable″ (1).

Another important aspect visible in the recorded data are daily recurrent patterns, i.e., the likeliness of a smoking incident given a specific hour of the day. The daily recurrences for each participants can be found in Table 4.10. The figure depicts the normalized histogram over 24 hours throughout the course of the study, where each bar represents one hour, for single smoking incidents. It is apparent that, besides the night-time, there is no fixed cross-participant distribution and that the precision of this measure likely depends on the time of participation. For smokers that have participated longer, this is likely more precise. However, the time-of-day where a participant is most likely to have a cigarette can thus be estimated. For example, participant 3 has consumed a cigarette at 13.00h on 77% of his 13-day long participation. Other similarly strong patterns are visible for other participants as well. This kind of analysis could allow forecasting the times when a participant is most likely to smoke and could serve as the basis for a more exact ahead-of-time intervention.

The strong differences of self-reported and measured total cigarette consumption are quite pronounced. This can partly be attributed to an unawareness or difficulty to estimate one's daily consumption. During the post-study questionnaire we also asked the participants whether the gathered data is matching their real consumption (Table 4.11 question 10), i.e. if the participants are "trusting" the system, which all participants rated as "applicable" or "somewhat applicable". Apart from the indication LEDs, there was no direct feedback of the recorded data. Participants thus could check whether data was recorded but only from smoke to smoke, not overall. The participants were however also asked to check the report of their daily consumption (see Figure 4.16) before answering the post-study questionnaire. This absence of direct feedback likely presents a limit of this study: Integrating a display into the lighter itself to display the gathered data during smoking incidents could affect smokers in a stronger way (Table 4.11 question 9) and would also allow to build more trust into the system.

The fact that the Smartlighter v1 (coil based) needed to be recharged often and others lead to significant adoption problems. The time needed for the coil-based lighter to heat up has been pointed out as problematic by some participants. Also the mechanical setup of the lighter, with only a small hole to match a cigarette in, was found problematic. This can be seen in the results of question 7 (Table 4.11), where participants have been asked if they solely used the provided prototypes.

From our experiments with different sensor modalities, we learned

that multiple complementary sensors to detect smoking incidents may increase accuracy. People may forget to pack their device, batteries may run out, etc. Such a cross-calibration of multiple sensor modalities can of course also be used for other types of sensors, which allows users to adapt the sensors they are wearing to their specific needs.

Furthermore, participants felt that the system helped them to raise their awareness of their smoking behaviour (question 6), could help them to quit (question 8), and deem the automatic acquisition to be useful (question 11).

4.5 SUMMARY

Wearable systems, as presented here, can detect smoking incidents *continuously*, with *minimal user interaction* and provide *detailed insights* into day-to-day smoking behaviour. This allows to gain new insights into one's personal addiction, for example patterns like a post-lunch or post-wake cigarette can be spotted. Based on such an insight, coping strategies like actively trying to suppress the urge to smoke, could be suggested on a mobile computing device. At the same time, it can be detected if such a suggestion was helpful. Just-in-time intervention based on a model, which predicts when a cigarette is most likely to be consumed, would be another novel possibility to persuade smokers to quit.

This chapter has highlighted the design of the Smartlighter, a device to detect smoking in an inconspicuous and energy-efficient way, which was tested in several user studies. Of those user studies, design guidelines for such gadgets were extracted and this instrumented artifact approach was compared to an EMA approach of eliciting smoking behaviour. Furthermore, a purely symbolic approach of classifying wrist motion, which can asses the variety of smoking datasets, was presented. Additionally a machine-learning approach was shown to achieve an F1-score of 84% on a smoking dataset with high variety (351 smoking instances), based on acceleration data only.

This chapter contains contributions from the following peer-reviewed publications:

Philipp Marcel Scholl and Kristof Van Laerhoven. "A Feasibility Study of Wrist-Worn Accelerometer Based Detection of Smoking Habits." In: *International Workshop on Extending Seamlessly to the Internet of Things*. 2012.

Philipp Marcel Scholl, Nagihan Kücükyildiz, and Kristof Van Laerhoven. "Bridging the Last Gap: LedTX - Optical Data Transmission of Sensor Data for Web-Services." In: *Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication*. 2013.

Philipp Marcel Scholl et al. "When Do You Light a Fire? Capturing Tobacco Use with Situated, Wearable Sensors." In: *Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication*. 2013.

Philipp Marcel Scholl and Kristof van Laerhoven. "Lessons Learned From Designing an Instrumented Lighter for Assessing Smoking Status." In: *ACM International Joint Conference on Pervasive and Ubiquitous Computing*. S, 2017.

CONTEXT AWARENESS IN THE WETLAB

Figure 5.1: Typical wetlab environment, top row shows workbenches and other environemntal circumstances, middle and bottom row typical steps in a low-safety laboratory.

Biologists work in highly dynamic, shared laboratory environments. Lab benches are packed with all kinds of different equipment such as tubes, racks, compounds, specialized machinery, computers and shrink-wrapped documentation. The experiments performed in such labs often take a significant amount of time and occasionally need to be suspended, for instance to wait for an organism to grow or until sufficient amount of DNA has been incubated. To provide valid results, these experiments need to be repeatable, which is why biologist heavily rely on handwritten notebooks to keep track of the specific steps taken during their procedures. Protocols are either established from scratch, incrementally refined or executed as close as possible.

Lab protocols are routinely written "offline", after the experiment has been performed, and are mostly reconstructed from memory, if written down at all. One reason for this is the large effort in putting down all experimentation equipment (e.g., pipettes, flasks, gloves, or containers) and moving to a different bench to take notes or retrieve information from a PC. Another, more profound reason is the risk of contamination - laboratories are classified into four bio-safety levels with increasing precautions to contain harmful agents, as they may pose a threat to the experiment, the health of the experimenter and to the

environment. This means that (possibly) contaminated materials (such as notebooks, protocols, laptops and cameras) are not allowed to be taken out of the laboratory without decontamination. At the same time, it is not allowed to take materials into the laboratory to avoid falsifying experimental result by contaminating the agents under investigation. This hinders the "in-situ" documentation retrieval/creation abilities of biologist while performing experiments.

In this chapter, an ensemble of wearable sensors and computing devices is presented, which is to support the short- and long-term memory of a biologist. Without the possibility to capture abstract thoughts directly, digital signals related to protocol steps can be captured. As most machines and environments are built to be experienced by humans, video and audio recordings, our main senses, are assumed to provide the largest portion of insight for remembering details. Continuous recordings of these modalities on their own are however hardly useful as they cannot be browsed quickly. Therefore the use of continuous, ego-centric audio, video, motion and artifact recordings for indexing pre-defined manual work is investigated. In contrast to other approaches, which instrument the environment to enable detection of process step, we look into instrumenting the biologist in order to gain flexibility, and to avoid the effort involved in instrumenting a wetlab.

To test whether it is possible to detect process steps via motion recordings, a dataset encompassing 22 participants in a entry-level DNA extraction experiment was conducted. Wrist acceleration and over-head video, serving as ground truth, were recorded. Additionally a wetlab course at the University of Darmstadt was recorded with seven participants, which allowed to gain insights into the practicality of system. With the gathered insights a third dataset, with a body-worn video, audio, inertial wrist motion and inertial head motion was recorded. Different recognition approaches were tested on both datasets. The remainder of this chapter is structured as follows: First, the recording setups for three scenarios are described, as well as a study into a wrist-worn object identification system. We then look into the detection of atomic action from wrist-motion. Afterwards a Hidden Markov Model (HMM) extracted from a textual description of the workflow, to detect workflow steps from actions is described. We then investigate a system that generates navigation cues without any prior knowledge about the process at hand, and finally present a unifying concept for integrating the detection of workflow steps and continuous sensor recordings for biologists, and other professions that

Figure 5.2: A selection of recording setups used for experiment throughout this investigation. To the left is the setup for the DNA extraction experiment consisting of Google Glass and the Hedgehog acceleration recorder, in the middle the one used for the wetlab exploration. To the right is the latest setup in which wrist motion is recorded with Android Smartwatches and video with a shoulder-worn 360° Theta camera.

require documented manual work.

5.1 MOTION-AUGMENTED VIDEO RECORDINGS

Searchable databases of multi-media recordings provide a way to augment one's memory. This database might contain video, audio and motion data, which is indexed to allow for quick searches. Ultimately, queries for similarity on each recorded modality would be supported. For example video sequencing showing similar objects or similar motion sequences can be retrieved quickly. An important aspect of this challenge is how to encode such multi-modal data, and how to make it searchable. One approach, based on a multi-media container format, is proposed here together with an architecture to allow for similarity queries on multiple sensor modalities. Examples of manual work, where this is useful are wetlabs, while cooking, while assembling products, or while maintaining machinery.

Each sensor generates a stream of data, in a certain format, at a certain rate and with parameters that have to be stored in order to compare them to other recording sessions. All of these recording

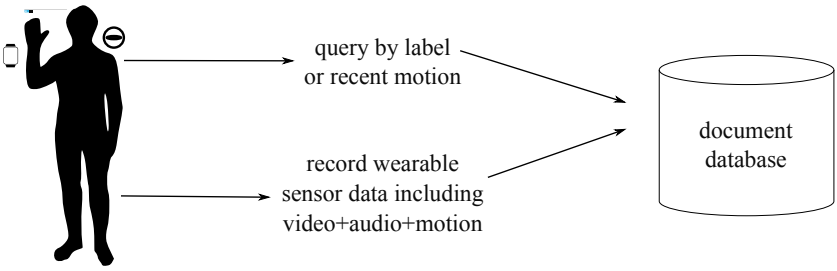


Figure 5.3: Conceptual overview of a query system for augmented video recordings, in which body-worn sensors as well as audio and video are recorded and analysed for quick retrieval of similar sequences

parameters should be stored in conjunction with the actual sensor data. Additionally, recording multiple sensors also involves synchronization. Even when running on the same device, independent sensors operate on their own clock, which renders time synchronization an important issue. For example, an accelerometer will deliver its samples faster, than a light sensor, if its package is placed in the vicinity of a heat source. Therefore, even system-local sensors, need to be synchronized on a common clock, in the same way multiple recording devices on a network need to synchronized.

Additionally to these issues, organizing recordings of such an ensemble of wearable sensors quickly becomes a burden. One possible approach is to use a multi-media container to store synchronized sensor streams on a common time-axis, together with their recording parameters. The matroska standard defines such a file format [232]. In this format, multiple video streams can be stored, as well as multiple audio, subtitle and data streams. These streams can be compressed with state-of-the-art codecs. Motion data can be stored as an audio stream as well, storing some of its recording parameters like sampling rate and sample format, and optionally be compressed. This provides a standard way to encode recording parameters, as well as the data itself in a synchronized way. Recording sessions or, if you will, memories, of a user can then be stored as such a multi-media file. Labels, which encompass a beginning and end time on this common time axis, and free-form text can be encoded as subtitles serving as navigation cues. This allows for storing ground truth data, as well as results from a recognition system. All side-by-side on the same time-axis in the same file.

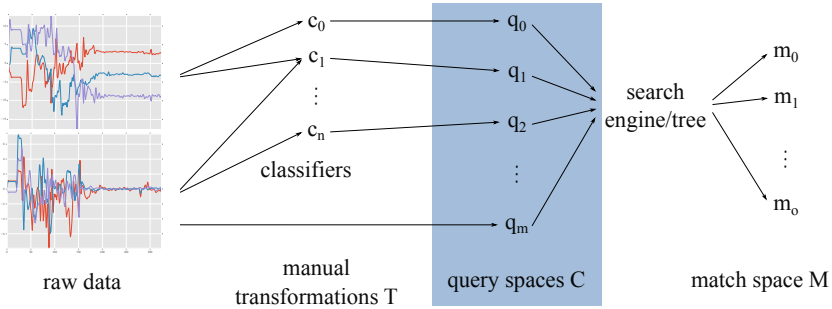


Figure 5.4: The actual similarity of raw motion data, which is used for indexing the multi-media files is fully defined by a mapping into a query space Q. This can either be done manually, or with the help of machine learning algorithms. The mapped sequence of data can then again be encoded into the multi-media file, and indexed by a search tree for fast k-Nearest Neighbour queries.

The question however is how to render these multi-media files searchable. How can we provide a way to query these for similar gestures, both by name and by the raw data recorded during different sessions? For example, how can we enable a biologist to execute a "pipetting" gesture and find recordings where she was executing a similar gesture. Both by querying the systems manually, in its literal sense by moving her arm, and by querying explicitly by a keyword like "pipetting" (see Figure 5.5). A way to encode this motion similarity is required.

The problem of querying for similar motion and by descriptive keywords can be viewed as finding a mapping from the space of raw sensor data S to a query space C, in which a clear definition of similarity is encoded (see Figure 5.4). Approaches range from manually defining this mapping to fully automated machine learning systems, where only a limited set of parameters are pre-selected. The input for such a system is always (labelled) raw data, and for unknown raw data the output is an element of C (or a label a if you like). The input labels, as well as the output labels are bound to regions on the time axis of such continuous sensor streams. These can be encoded into the matroska document as subtitles, which encodes a beginning and end timestamp, as well as a corresponding label, exactly the output of aforementioned mapping.

Put into the context of the biologist's example, the query space

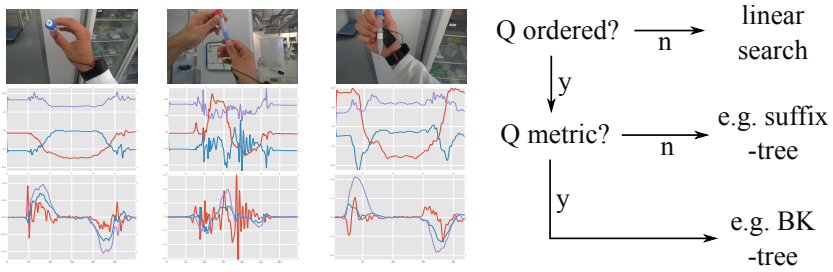


Figure 5.5: Illustration of motion patterns that might be queried in a wetlab environment and how to decide on a search algorithm.

C consists of labels such as "pipetting", "mixing" and other manual actions. Once a mapping from the raw data S to elements of C is established, it can be used to generate subtitles for the recording. This means a gesture recognition system, which maps movement data from S to labels such as "pipetting" of C, can be encoded as a subtitle. These subtitles are then stored side-by-side with the multi-media recording, and used as a possible search cue. The encoding of such subtitles is general enough to span sequences of fixed and varying sizes, and can also be used for hierarchical label sets. In the presented matroska document based architecture, classifiers (a different name for the mentioned mapping), will run on a just recorded document. A subtitle stream will be added to the document from the result of this classification step, and added to the document.

Under which conditions is a database of such classified recordings quickly searchable? This depends on the nature of the query space C, and the similarity measure g defined on it (see Figure 5.5). If g is metric, a *metric tree* data structure can be used to accelerate the search on. This means that the addition operation is defined for elements of C, and that the triangle inequality is fulfilled. If only an order relation can be defined other tree search algorithms need to be used. If neither condition is fulfilled by g, only a *linear search* is possible. Complexity-wise, a tree-accelerated search query can be executed in $\mathcal{O}(\log n)$, while a linear search has a complexity of $\mathcal{O}(n)$, where n is the database size. Therefore, the goal is to transform raw sensor data into a metric or ordered query space C.

One important design decision for a wearable remembrance agent is the format in which sensor data is stored. Here, an approach based on a multi-media container, which encompasses all relevant parameters of

a recording session, is proposed. Furthermore, assuming that a number of classifiers for motion (and other) data is available, the classification results are to be stored side-by-side with the sensor data. If no such classifier is available, a similarity mapping would have to be defined manually (e.g. euclidean distance), or enough labelled data would have to be collected to build such a classifier. The container format renders the exchange of data easier and by having a standard format, allows for easier development of novel classification systems. The results of these classifications can then also be quickly searched, and by proper choice of similarity measure also accelerated. With such a system in place, the user might not only easily document his manual tasks, but also quickly query for similar recording by his recent motions. Hence, the activity recognition systems can also be interpreted as a search problem, with the major challenge of defining a similarity measure on sensor data.

5.2 OBJECT RECOGNITION WITH WEARABLES

Additionally to wrist motion, *Object Identification* could provide a sensing modality which allows process step recognition. Detecting RFID marked objects with a wrist-worn unit for activity recognition was shown before already [222]. Rather large RFID tags were attached to household items to detect activities of daily living. With a similar setup (wrist-worn RFID reader), we investigated how well test tubes in common use in the microbiology lab can be identified. For this, miniature RFID with a diameter of less than 15mm were attached to test tubes. Figure 5.6 depicts the whole system setup.

The first design challenge for such a system is the antenna design. The placement on the human hand mainly dictates the possible choices. The first trials of such a reader placed the antenna on the back of the hand [222, 174, 221] which allowed for reading distances of $1-2$cm. Antennas looped around the wrist [223, 222, 224, 221] have replaced this design. However loop antennae need to be rigidified to keep their performance controllable, a $10-15$cm reading range with a common 5cm-patch RFID-tag has been reported. Flexible antennae placed between thumb and index finger [81] are challenged by sweat and by changing (antenna) parameters due to movement. While placing the antenna on the thumb achieves the best reading performance, especially for miniature tags, its attachment point also hinders the movement of the wearer's hand. We therefore decided to compare a flexible antenna worn in the palm, and a rigid loop antenna worn around the wrist. Re-

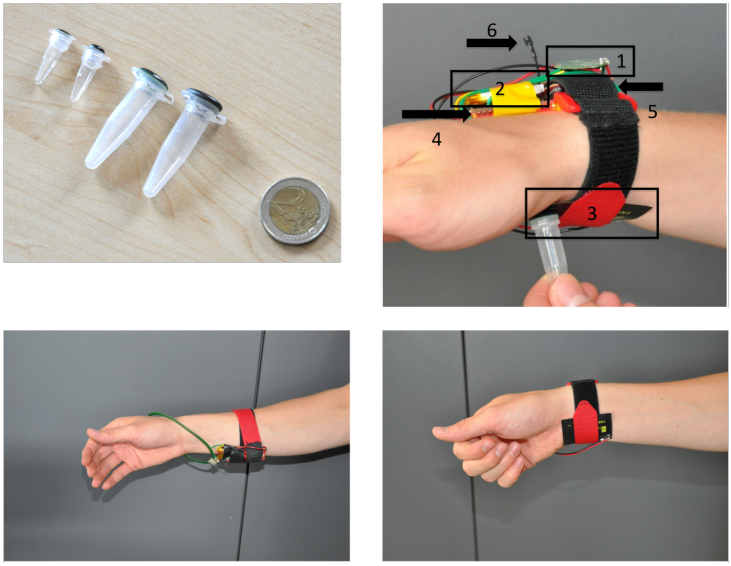


Figure 5.6: Tracking miniature components in the Wetlab (top left). The wrist-worn RFID reader (upper right) is built from a (1) Skyetec M1 Mini reader (2) battery pack (3) RFID antenna (4) Arduino Fio module (5) Wifi module (6) Wifi antenna.

cent developments propose Glove-integrated fiducial, and RFID readers [298, 299].

The presented system is thought to ease the identification and labelling of sample tubes in wet laboratories by using a wrist-worn RFID-unit and to tag them. Two tasks are of importance for the experimenter using this system: (1) identifying a sample tube (reading a tag) and (2) labelling a sample tube (writing a tag). Both tasks should be supported in a hands-free manner, during normal laboratory routines, to relieve an experimenter from manual labelling tasks. Hand-written or colored stickers with a separate lookup table, or hand-operated label printers are the current state of this art. In turn, this requires the experimenter to put down all tools and concentrate solely on the labelling or identification task. The design of a hands-free system for identifying sample tubes with RFID is based on these practises and by the idea to limit the amount of interaction to a safe minimum. Not only single tubes, but also multiple correlated ones are typically labelled.

This correlation is often a variation of one parameter, for example the amount of concentration of one compound. Labelling a series of tubes is therefore also included in our design.

RFID tags. We chose RFID tags that can be glued on the lid of sample tubes, since sample tubes count as consumables in a wetlab, and are usually thrown away after usage if not kept for long-term storage. Integrating RFID tags directly into the tube is thinkable, however it is more likely that tags are integrated into some kind of removable attachment. For example, they could be integrated into re-usable protective caps which are routinely used during storage and transportation. Since the caps of the smallest sample tubes have a diameter of only 10*mm*, we decided to test tags of according size. Miniature tags[9] (cf. Figure 5.6) with a diameter of 15 to 5.5*mm* are glued to the top of sample tubes for our prototype. The associated information for each tag is stored in a central database to avoid local RFID storage limitations.

Reader. To test different antennae and attachment points, an off-the-shelf RFID reader module (Skyetec M1 Mini) was hooked up to a micro-controller. The reader could be operated in *continuous* and *on-demand* mode. In continuous mode, the reader actively scans at 20Hz, which draws 86mA of power. On-demand mode draws the same power, albeit only when the user explicitly interacts with the system. Implicit interaction requires the reader to operate in continuous mode, since a detected tag is an interaction cue. Slower continuous reading rates are possible but need to be carefully balanced with the reaction time of the system.

Connection. The wrist-worn reader does not include any user interface. Google's Glass, specifically its activated interface, allows for interaction. The connection between Glass and the reader is established via a WiFi interface. WiFi was chosen since it would principally allows the system to be used in other scenarios as well (e.g. statically placed reader), and is easier to integrate into existing applications. The wrist unit provides a TCP server, that only operates the RFID reader while a client is connected. In continuous mode a tag's identifier is directly transported to the client, while in on-demand mode, a read needs to be requested first. With this design, energy-saving modes can be readily implemented.

Interaction. Two interaction mechanism are provided. Both support reading and writing tags, however the first *implicit* one requires a lot less spoken interaction but requiring more energy. Constant operation

[9]manufactured by MicroSensys GmbH `http://www.microsensys.de/`

Table 5.1: Reading times and maximum distance (0*mm* for those that need direct contact to the antenna) for each RFID tag.

	large antenna		small antenna	
	distance	time	distance	time
d14-special	5mm	.7s	12mm	.7s
d14-tag	20mm	.8s	15mm	.6s
d7-tag	0mm	.6s	0mm	.6s
d6.7-tag	0mm	.6s	0mm	.6s
sticker	30mm	.8s	100mm	.7s

of the reader unit is required for this scheme. We hypothesize that the advantages of an implicit interaction outweigh a shorter system runtime. The implicit interaction refers to an interaction that is activated by placing a tag next to the RFID reader, subsequently the current label is read and displayed on Google Glass. An optional voice command allows to re-label the current sample (cf. Figure 5.7 top). For *explicit* interaction, the identification and labelling task has to be started manually via a voice command. A voice menu on Google's Glass supports this by providing key phrases, after Glass has been activated by head movements. Afterwards, the user is guided through the whole process of tag detection, i.e. placing the tag on the reader, and displaying the results (cf. Figure 5.7 bottom). After a tag has been detected, the interaction is the same as for the implicit case.

Both techniques differ in the time spent for tag detection. In the implicit case this time is "hidden" from the user, by activating interaction possibilities only after successful detection. The explicit interaction provides more feedback to the user, giving hints on what to do next if tag detection fails. A user study to shed light on which antenna is acceptable, and which interaction mechanism is more usable was conducted.

Experiment. Seven students, aged 25 to 35 years, one female and eight male, were recruited at TU Darmstadt. For all of them a technical affinity could be assumed, and they were partially experienced with Google's Glass. Two small sample tubes (diameter 8*mm*) and two large tubes (diameter 13*mm*) were provided. Small tubes were tagged with a d6 and d6.7 tags, while the larger ones were tagged with d14 tags. Table 5.1 highlights that small tags do not allow for non-contact reading.

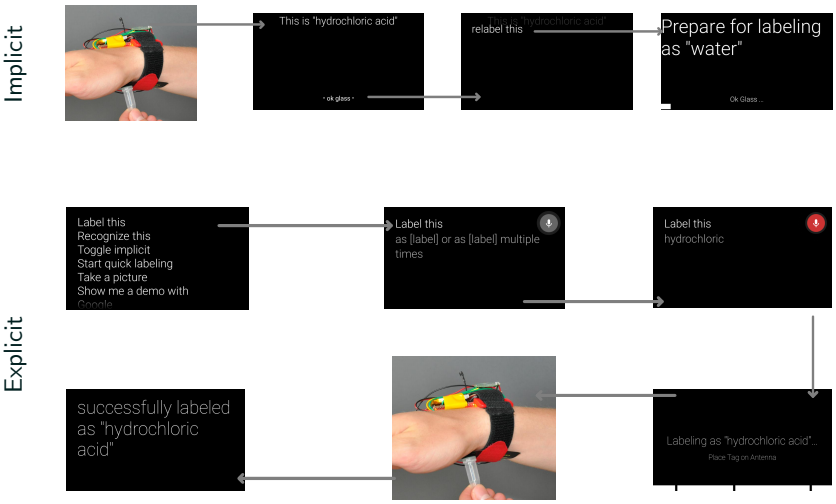


Figure 5.7: Implicit and explicit interaction to identify and label an RFID-tagged test tube in the wetlab with a wrist-worn reader and a head-mounted display. Top shows the implicit interaction, bottom the explicit one which needs to be started via voice command.

A water bottle tagged with a $25x25mm$ standard tag was also provided to emulate a large container in a wetlab. For transferring liquids a pipette was provided. Google's Glass and our wrist-worn RFID-reader prototype was worn by each participant.

Each participant was outfitted with Glass and our wrist-worn RFID-reader. A small introduction to the first interaction scheme was given. Afterwards the participant was asked to identify and re-label the D14-tagged tube. The next task was to label a series of all containers (including the water bottle), in order to test the series labelling process. The final task was to transfer water into a D14-tube and label it accordingly. These task sets were repeated for each interaction scheme and each antenna, four times in total. Starting with either explicit or implicit interaction was counter-balanced over all participants, selected at random by the examiner. The rigid antenna was always tested first. After each test, the participants were asked to complete a System Usability Scale (SUS) and were asked for general remarks.

Results. Implicit interaction (81.1) scored only slightly higher than explicit interaction (79.1). When looking at Figure 5.8 implicit interac-

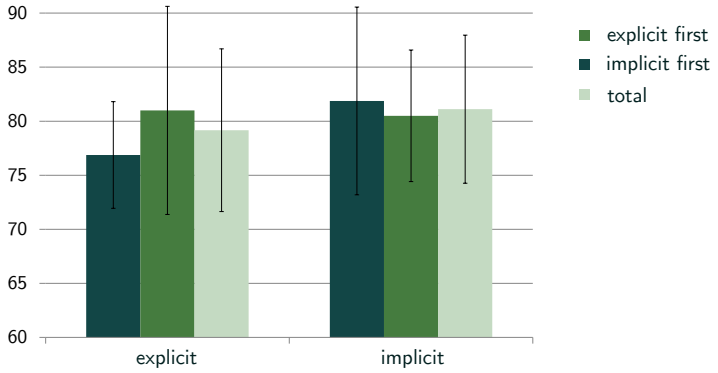


Figure 5.8: Mean and standard deviation of SUS scores. Total score, and score when explicit or implicit interaction was done first are shown. Implicit interaction generally scores higher, especially when introduced last.

tion is scored higher when introduced after explicit interaction. We assume that this is due to a familiarization with the system. While implicit interaction is not self-explanatory, it becomes much more obvious when introduced after the more verbose explicit interaction. This confirms our earlier assumption that user training can replace more explicit feedback. Participants identified a major speed-up for identification tasks as one of the strengths of implicit interaction. However, the missing feedback when tags were not detected, even though they were next to the reader was mentioned as a short-coming, mainly by those participant that have started with implicit interaction. A reliable reading process, when using RFID readers for initiating interaction is therefore of major concern.

Only one participant scored the large, rigid antenna higher than the small flexible one, even though the small one provides a better reading performance. Besides concerns of comfort for the large antenna, it was unclear for most participants how to hold the tags to achieve good reading performance. The small antenna made this clear, since it was attached to a flat surface, rather than spun around the wrist. This is only an issue for small tags though, since larger tags also provide better reading performance where orientation does not have a strong influence. A combination of both antennae, for example one integrated in the wrist-band and one on the wrist, would remedy those issues.

Dicussion Generally, it can be said that implicit interaction is preferred over explicit interaction. However, object identification and tagging remains an open issue. Better reading performance, particularly for tiny tags is required for practical usage of this system. Further integration would allow for detection sensor readings from such tags as well, for example monitored temperature or light intensity for containers of biological compounds. This way storage problems can be identified quickly prior to falsifying the results of an experiment, or of a diagnostic test. Currently, fiducial markers are ubiquitous, without the additional benefit of reading sensor data from RFID tags. A wrist-worn fiducial reader without explicit interaction (i.e. pointing onto a marker) is probably more practical.

Object identification is the major addition to the previously presented recording infrastructure, and also the major difference to the data collection for the smoking case study. These object identifications can be encoded into the matroska file as well. The results of this non-continuous sampling are encoded in subtitle format, i.e. marking the beginning and end of intervals while a tag was in the vicinity of the reader. Object identification, due to its practical limitations was not further studied in the following experiments, however should be considered for practical deployment, as it offers a robust way of detecting activities when marking objects is not prohibitive and tags of practical size can be used.

5.3 DEPLOYMENTS WITH GOOGLE'S GLASS

For the work presented here, a basic voice-interacted task guidance and logging system running on Google's Glass was developed. The system is able to display workflow steps, and navigate or mark steps as done through voice interaction (see Fig. 5.9). It also includes the ability to capture audio and video from Google's Glass, and log all interactions. Additionally, gestures made by the dominant hand are recorded with the low-power inertial sensing unit HedgeHog (see Figure 5.12). The data of both systems is merged offline, as this implementation was only used for this particular experiment.

Google's Glass was used to capture experiments via its side-mounted video camera and microphone. Additionally, an application was designed to guide experimenters through pre-determined experimental protocols. For the latter, an example user interface presented to participants can be seen in Figure 5.9 and shows some of the steps of

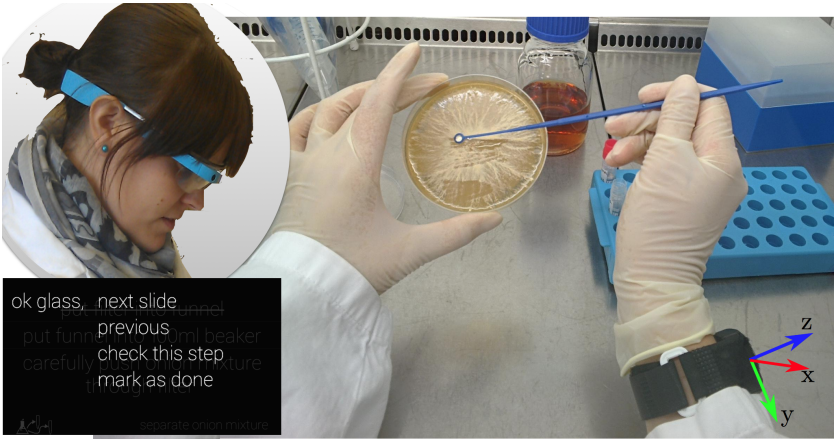


Figure 5.9: The Task guidance system used in the DNA extraction experiment. A wrist-worn accelerometer logs wrist motion, while participants are guided through the experiment with tasks displayed on Google's Glass which are selected via voice commands.

the DNA extraction protocol that was used during this study. Single steps in the protocol are shown in a timeline, that could be navigated to the left (for past steps) and to the right (for future steps). The wearer has the choice to navigate the protocol using either the swipe gestures on the touchpad[10] on the Glass, or by voice commands. The subset of voice commands chosen for this were "previous", "next slide", "check this step" and "mark as done". These commands were chosen by experimenting with their recognition rates, trying to increase their phonetic dissimilarity for multiple speakers. By saying "check this step", "mark as done" or by tapping the touchpad, the user can let the system know that the step was performed, which is visualized by striking the current item through. The "ok, glass" guard phrase, which is usually required for Glass apps, was removed to minimize interaction time. If this item was the last step on the slide, the system automatically displays the next step. The application was implemented using the Android Framework, and works on Google's Glass as well as on Tablets and Smartphones.

The guiding part of our system is designed to be easily adaptable to different protocols. For this, we decided to use a document-driven ap-

[10]The touchpad was enabled in this study to avoid problems with voice recognition for non-native english speakers.

Figure 5.10: The deployment of the system combining Glass and the wrist-worn accelerometer, while recording biologists. The environment is often simultaneously used by large groups of researchers or students and is equipped with a multitude of shared instruments and special safety zones, making it challenging to augment the environment (top half). Hands-free recording is a strong advantage: Often, experiments require gloves for minimizing contamination risks; Wet labs furthermore contain a large variety of compounds, instruments and lab equipment that require both hands to be used (bottom half).

proach, in which a human-readable and machine-parseable document contains the steps of a procedural protocol. These steps are written down in Markdown [300] documents. This allows experimenters to modify and present these workflow steps on different personal computing accessories (like PCs, laptops, Smartphones, and Wearables) without much implementation effort. It furthermore allows for linking documents, and referencing additional media files. However, the major reason for using a document-driven design is that modification of the protocol can be captured easily in a distributed fashion. Only the transformations of the document need to be transported to a central repository. The protocols can then be shared cross-device and can be scoped on per-user basis. This provides the means to share, collaborate on and synchronize the experimental protocol. The system was deployed with biologists in two different wet laboratory environments:

an academic teaching and an academic research facility.

5.3.1 Recording in a Teaching Scenario

Four recording sessions of up to a full day were held. In each recording, up to five microbiology researchers or students were wearing the system simultaneously, working within groups of up to four persons. Figure 5.10 shows some examples of the video footage taken with Glass. All users were at all times aware of the recordings (Glass showing the current recording) and were encouraged to discuss their experimentation steps and methods, and to provide feedback of the system and its possible advantages and disadvantages. The sound recording's quality of Glass was good enough in all environments to understand both the user and the people in the immediate proximity. After the system was handed out and activated, we did not remain present in the laboratories, and examined the contents of the videos afterwards. Due to battery and efficiency limitations, about one hour of continuous recordings was possible.

In general, the acceptance of wearing the system was high, and even in the teaching laboratory, where approximately 20 fellow students were working in the same immediate environment of the user, nobody expressed concerns about the possibility of them being recorded. The latter observation might be due to the fact that experimenters do frequently take their personal cameras with them to photograph or record important experimental results (if safety considerations allow for this). Apart from a few remarks made towards the end of the teaching sessions (which lasted over three hours each), we did not note any big signs of discomfort in wearing the system: Glass was at two occasions taken off to concentrate on using a microscope, and once to demonstrate it to a fellow user. One of the wrist-worn accelerometer sensors did not record consistent data as it was not strapped on tight enough and had rotated along the wrist during the course of the experiments. The video quality of Glass (at 720p) tends to be good enough to be able to read most compound labels and handwritten notes.

Results. Several findings that emerged from the video footage are especially noteworthy: (1) Even in laboratories with a lower safety clearance which implies minimal contamination risks and therefore does not require gloves, the option of taking pictures or videos hands-free is a strong advantage. On many occasions, users required both hands simultaneously to handle instruments and the fact that Glass was

	task	actions
1	**solvent** combine 50ml lukewarm water, 1/2 teaspoon salt and 3ml dishsoap in 200ml beaker and stir	pouring, transfer, pipetting, stirring
2	**cutting** peel and cut onion/tomato	peeling, cutting
3	**mixing** mix into 200ml beaker, add 1ml detergent, and stir	pouring, pipetting, stirring
4	**waterbath** put 200ml beaker into hot waterbath for 10mins	
5	**waterbath** put 200ml beaker into cold waterbath for 5mins	
6	**pestling** pestle mixture	pestling
7	**filtrate** put filter into funnel, funnel into 100ml beaker, push mixture through filter	
8	**pouring** pour 1.5ml of mixture into test tube, mix in 5ml freezing ethanol	pouring, pouring
9	**detection** carefully invert test tube multiple times	inverting

Table 5.2: The (shortened) DNA extraction protocol as shown to participants on Google's Glass. The protocol was interleaved for both an onion and tomato, creating 18 steps in total. Gestures used to detect each step in the protocol are shown in the right column.

able to record from a first-person perspective was mentioned as a great feature. (2) The use of pen and paper notes is still largely preferred as the primary capturing system. Partly, this is due to its flexibility, but the videos also made clear that ad-hoc written notes, labels, instructions and lab books are truly ubiquitous in the wet lab. A digital system for providing assistance in these surroundings, apart perhaps from some tightly-regulated laboratories, has more chance of adoption when introduced as a complementary technology. (3) The ability of following what is recorded by Glass in the peripheral display was at multiple times used to guide the capturing of the video. When looking through a microscope, for instance, several users used their Glass' display to make a recording through the eyepiece. Instead of taking immediate notes on compound quantities, e.g. to record how many millilitres of a solution

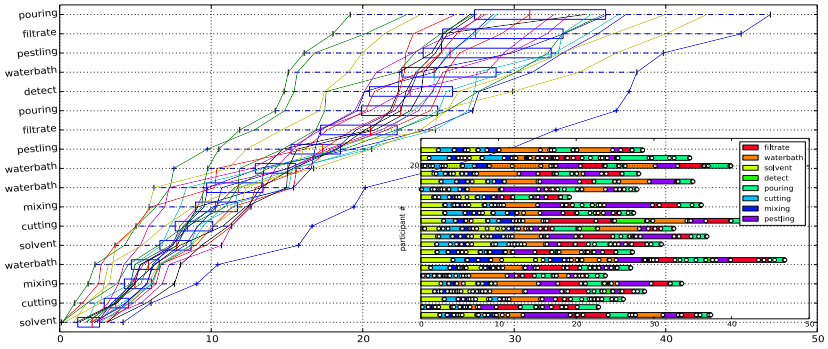


Figure 5.11: The mean duration of protocol steps per participants. The figure in the background shows the per-step mean duration across all participants (box-and-whisker) and the individual flow for each participant (lines). The figure in the bottom right shows the mean duration of each step per participant, color-coded to individual steps in the DNA Extraction protocol. Note that each step is repeated twice and interleaved during the experiment, once for the onion and once for the tomato. Markers on this figure show when an actual user interaction happened (when marking a step as done for example). The x-axis on both figures is the time taken in minutes.

were obtained, users would hover it closely to Glass' camera. Both during and after the deployments, many of the users were interested in using the system for subsequent times and expressed that they could envision continuing using it for capturing their experiments.

The idea for this deployment was to get first insight into the working environment of experimental biologists, and also get a first idea of the usability of using Glass for recording only. To this end, we elicited challenges from several point-of-view video recordings from several laboratories. One specific challenge, that was mentioned by participants, is the navigation of such continuous recordings. We will now look at the possibility of using wrist motion to detect the actions conducted during an experiment. Possible actions, extracted from a procotol's description, inform the extraction of time-codes which index such continuous recordings - serving as navigation cues for these recordings and providing guidance just-in-time.

5.3.2 Guiding in a Research Scenario

DNA extraction, a common entry-level laboratory experiment, was chosen for testing the feasibility of guiding novice users through an experiment and recognizing single activities in an experimental procedure using the wrist-worn sensor. The experiment is guided by a protocol, visualized as a textual step-by-step guide on the Google's Glass display. It contains a sequence of two extractions of DNA, first that of an onion and second of a tomato. The procedure is identical for both vegetables. This way all actions are repeated at least once with different material. The two experimental procedures were interleaved to save time, and each participant had to complete 18 steps in total, containing 9 different protocol steps to identify (shown in Table 5.2). Each protocol step was described and displayed when activating the screen of Google's Glass (by tilting the head up or tapping Glass' swipe area). Participants were instructed to primarily interact with Glass by speech and to move through the experiment protocol by marking each single step as done. In case the speech recognition would prove to be impractical, touch interaction (tapping) and swiping back and forth for moving between steps was kept as a backup option. The time for which a step was active on Glass' display was recorded in a log file, and is assumed to be the time it took to go through the displayed instructions. Figure 5.12 shows the experimental setting before and during the experiment, as recorded from a camera that was mounted at the ceiling.

The experiment was run at the Federal Institute for Occupational Health and Safety in Dortmund. In total, 22 people took part in the experiment. Participants were recruited via advertisements in a local newspaper, representing persons with no prior experience in biology experiments and no affiliation to our research. Before the task started, the participants were introduced to the functionality of Google's Glass and how they should use it during DNA extraction, what the different ingredients e.g. ethanol or detergent are, and where to find them. The participants were then fitted with Glass and the wrist-worn sensor, and asked to follow each displayed step and mark them as done as soon as they are completed. Conducting the experiment took the participants between 18 and 45 minutes. A successful experiment would result in the DNA becoming visible as a set of small stripes in a test tube, although for our evaluation it did not matter whether the DNA extraction was successful in the end.

Figure 5.12: Four selected steps during the DNA Extraction study. The participants are wearing a wrist accelerometer and Google's Glass. The latter guides the participant through the experiment.

The choice of novice users instead of professional experimenters might be surprising. However, this study was designed to show that hand motion sequences can be used to detect protocol steps and actions. Especially to create a dataset that can serve as a benchmark for different detection algorithms. It is therefore important to have a high variability in executing different actions, as this is the case also for professional experimenters, i.e. everybody has their own styles. More execution variability also means a harder challenge for the detection system, so if it works for untrained personnel it will most likely work for trained personnel as well. Also, since participants were non-trained, they also adhere stricter to the protocol, a professional in turn might take shortcuts in the experiment since he is aware of the overall goal and working of the experiment. This would create datasets with different execution that are harder to compare. The recorded dataset can therefore be used as a baseline benchmark for recognizing actions that are related to those in a wet laboratories - in a systematic manner.

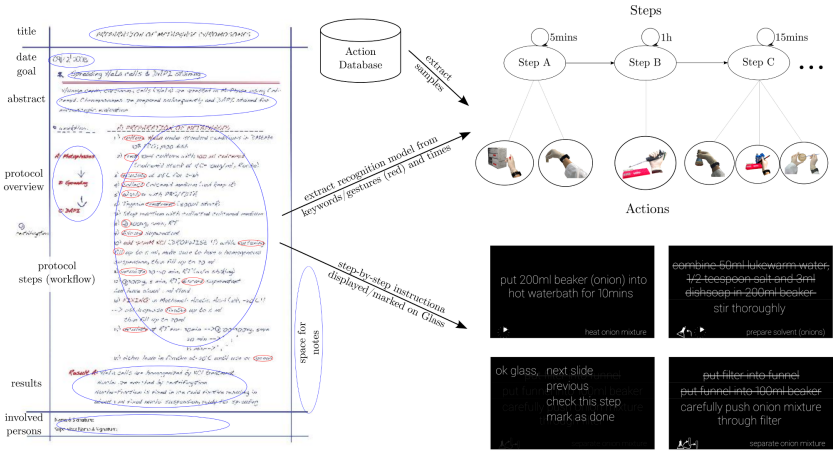


Figure 5.13: A (typical) page in laboratory notebook and the extracted recognition and guidance system. An action database contains recordings of wrist motion samples. Actions roughly correspond to verbs in the description. This database will be used for action detection, which in turn serve as the observations of a Hidden Markov Model, which contains each step in the protocol as a hidden state. Time is implicitly encoded via the number of observations. Each protocol step is displayed on Glass for guidance.

The goals of this study were threefold: (1) Examine the specific interactions with the guidance applications to see different usage patterns of participants. (2) Show how well actions (defined as repeated motion sequences) can be detected by wrist acceleration - or put differently, how discriminative the measured data is when applied to typical workbench activities. (3) Evaluate whether the detectable motion sequences (or action sets) correlate with specific protocol steps. Our interest is first and foremost in knowing how well a system could detect the combination of recording interactions, while guiding people through a wet laboratory experiment, and wrist movements. This could be used to automatically detect steps in wet laboratory workflows.

During the whole experiment, every interaction (both voice commands and swipes) with the Google Glass application was logged for later investigation. As this was done in the background, participants were unaware of this during the experiment, though they were informed about the interaction logging beforehand. A first observation is that

most of the participants did not follow the protocol in a strict linear fashion. Sometimes this was due to unclear instructions, which became clear when looking at future steps. Though sometimes this was also because of slight delays in the voice recognition with Glass, leading to commands which were given twice. To still extract the currently active step, the interaction log was filtered to include only steps that were visible for more than a few seconds.

Results. From the recorded data, it is possible to extract which step of the DNA extraction experiment was viewed at which time, and the duration for which this instruction was visible on the display. Figure 5.11 visualizes the resulting workflow for each participant, as well as the mean interaction time per step. It can be seen that the overall interaction time ranges from 18 to 45 min for all participants, and the experiment was completed in 35 min on average. Furthermore, a larger break can be observed in the middle of the experimental workflow (two consecutive water bath steps), which matches the instruction from the experiment's protocol: during these steps, participants had to wait until both the onion and tomato mixture had been cooled or heated up respectively, since no other task could be performed during that period. While interpreting these figures it should be kept in mind that the increasing variety in later steps is an artifact of the cumulative display of this particular step's duration. The figure to the bottom right contains the color-coded steps which are the same for both the tomato and onion extraction, i.e. the steps which are repeated for each participant. For example, preparing a solvent agent needs to be done twice, and is encoded in yellow in this figure. A large variety for solvent preparation time, in the duration of keeping the mixtures in the water baths, for filtrating and pestling can be observed.

Based on these interaction logs in combination with video footage made during the systematic study, the following three observations stand out particularly:

- There is first of all a large cross-user variety concerning the duration of each step, most critically for the steps where participants were instructed to keep a fixed time, e.g. keeping the mixture in the water bath for a certain amount of time. For several participants, browsing through the steps using Glass' voice commands was too time-consuming and they switched to swiping gestures, mid-experiment. This large variety in performance times has as a consequence that timing within an experiment and duration of single steps are important parameters, though they are also less valuable for

automatic detection or reconstruction of the experiment's protocol.

- Participants were asked, for steps that consisted out of multiple items, to acknowledge each task item separately as soon as it was completed. Several participants found this too cumbersome and did not adhere to this instruction - most often, these participants worked through the whole instruction set for one such particular information card, and then marked all items in one go.

- Finally, it is important to note that all participants were able to finish their experiment with sole guidance of the wearable system, without abandoning the experiment, and extracting the DNA successfully. Figure 5.11 shows the time (in minutes) that all experiment steps took per participant.

5.4 WRIST-MOTION WETLAB ACTION RECOGNITION

For evaluating the detection of experiment steps by means of the wrist-worn accelerometer data, the ground truth was gathered by annotating the recordings of an external camera [11], pointing at the manipulation space of the participant. By manually annotating the video we extracted 9 different actions, which had a high visual similarity and were repeated often during the experiment: The onion and the tomato were both *cut*, and the onion was also *peeled*. A pipette was used for combining different ingredients, e.g. *pipetting* the mixture into the test tube. The *transfer* activity describes using a spoon for putting, e.g. salt, in a beaker, but not using it for *stirring* for which a stirring rod was available. *Pouring* describes putting the mixture from one beaker to another or when pouring it into the filter. *Pestling* refers to mincing the mixture and *inverting* to putting the test tube upside down and back again. We refer to these video annotations as the ground truth in the following (cf. Table 5.3).

Methdology. Wrist 3D-acceleration data was recorded throughout the experiment on the participant's dominant hand with a sampling rate of $50Hz$, and range of $\pm 4g$. In total, 1258 mins of accelerometer and video data were recorded. Additionally, the interaction with Google's Glass was logged during the experiments, i.e. the timestamps when users switched to the next steps. All data was stored on the respective

[11] The camera on Glass was not used to make sure that the whole manipulation space was visible throughout the experiment.

devices and aggregated post-experiment on a PC. For time synchronization we relied on the internal clocks of both Glass and the wrist-worn sensor. We manually fine-tuned the alignment by matching the video and sensor data stream according to easily identified activities, such as stirring.

To recognize the activities listed in Table 5.3 from acceleration data, we chose a k-nearest Neighbour (k=8) classifier[12] with a 6D feature set, containing the mean and standard deviation of the 3D acceleration data during 20%-overlapping windows of 800*ms* duration. The video-annotated data was validated *cross-* and *per-participant*. Cross-participant validation was achieved with a leave-one-participant-out strategy, while per-participant validation was done by a 250-times stratified shuffle split. Precision, recall, and F1-Scores for each evaluation are listed in Table 5.3. It is visible that cross-participant activity recognition is worse than per-participant: On average, the cross-participant F1-Score is 36% worse than per-participant, which is most probably caused by participants performing activities in a slightly different fashion or due to sensors not being firmly attached. *Inverting*, *pestling* and *pipetting* are the three actions that show a particular high F1-score per-participant and comparable F1-scores across participants. In contrast to *stirring*, which is detectable per-participant but not cross-participant. *Cutting*, *peeling*, *pouring* and *transferring* (which in our case meant moving material with a spoon) are already hard to detect per-participant. The confusion matrices (cf. Figure 5.14) show that *pipetting* and *pestling* are most often confused with other actions, which therefore makes it advisable to exclude those from recognition. From this we conclude that several characteristic actions can be detected with reasonable performance, when trained per-participant. In a practical system this could be achieved by continuously learning gestures from the interaction with Glass. However, it also shows that an improved feature set might be a possible option for cross-participant identifications, as the one presented here is one of the simplest choices.

Results. From the confusion matrices shown in Figure 5.14 it is visible that *pipetting* and *pestling* are most often confused with other actions, per-participant. However pipetting is also the most often performed activity. This effect gets worse cross-participant (cf. Figure 5.14). *Transferring* and *pestling* worsen this effect even more. Therefore a general cross-participant activity recognition remains challenging. Detection rates could be increased by removing the almost non-detectable actions:

[12] the scikit-learn implementation was used.

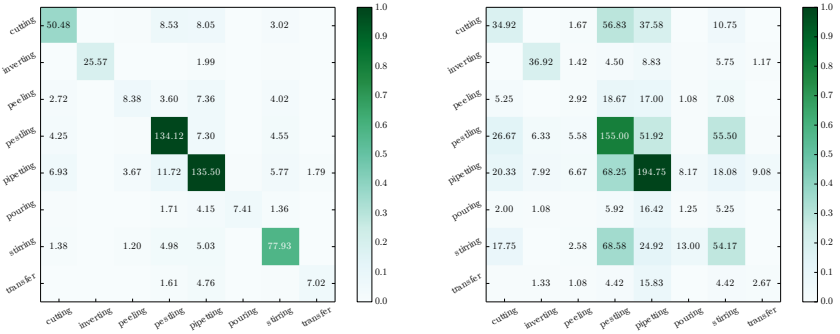


Figure 5.14: Confusion matrices for kNN detection based on 800*ms*-windowed mean and standard deviation features extracted from 3D-acceleration data. Cells contain the average absolute number of identified samples. The color designates the normalized total occurrence. Left hand side is the per-participant stratified random split repeated 250 times. Right hand side is the leave-one-participant-out score.

peeling, *pouring* and *transferring*. However since per-participant scores are quite good, an approach which allows to retrain at certain steps or reuse already learned samples could be fruitful.

To achieve this, a more reliable but not always available activity recognition approach could be used. Since our ultimate goal is to map back from actions to steps in a protocol (for indexing purposes), we could for example use the data from the interaction log, i.e. data that is gathered by biologist explicitly telling the system where in the protocol they are. It should be clear that this will only work in a few select cases, for example when the experimenter has never done the protocol before and needs this information or when asked specifically to do this for a study. In principle trying to map wrist movement back to steps displayed on Google's Glass.

With the current system, an average F1-Score of 64.5% per-participant, and 29% cross-participant is feasible. These score corresponds to the probability of detecting an action correctly for each 800ms window in the wrist motion stream. For the envisioned application of cueing video material, this action detection is probably not useful. However, one should keep in mind that this systems detects single actions on very short time-frames. However, this action detection might be sufficient to identify protocol steps, which are ordered timely and where actions are known beforehand.

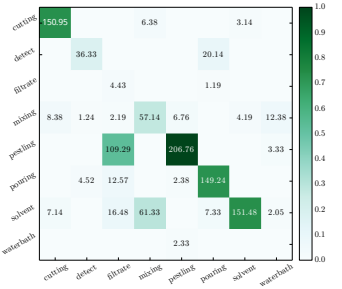
action	cross-participant precision	cross-participant recall	cross-participant F1-score	per-participant precision	per-participant recall	per-participant F1-score
cutting	0.23 ± 0.12	0.29 ± 0.17	0.25 ± 0.12	0.62 ± 0.20	0.75 ± 0.11	0.66 ± 0.16
inverting	0.60 ± 0.21	0.64 ± 0.34	0.58 ± 0.25	0.80 ± 0.21	0.93 ± 0.16	0.84 ± 0.18
peeling	0.04 ± 0.04	0.12 ± 0.15	0.06 ± 0.06	0.26 ± 0.17	0.51 ± 0.21	0.32 ± 0.17
pestling	0.62 ± 0.21	0.42 ± 0.13	0.47 ± 0.12	0.87 ± 0.08	0.80 ± 0.07	0.83 ± 0.07
pipetting	0.58 ± 0.13	0.52 ± 0.11	0.54 ± 0.11	0.79 ± 0.09	0.77 ± 0.06	0.78 ± 0.06
pouring	0.03 ± 0.04	0.13 ± 0.27	0.04 ± 0.05	0.45 ± 0.22	0.80 ± 0.24	0.55 ± 0.21
stirring	0.32 ± 0.25	0.37 ± 0.21	0.29 ± 0.15	0.83 ± 0.11	0.80 ± 0.08	0.81 ± 0.08
transfer	0.08 ± 0.10	0.25 ± 0.34	0.08 ± 0.09	0.33 ± 0.28	0.49 ± 0.34	0.36 ± 0.28
	0.31 ± 0.29	0.34 ± 0.29	0.29 ± 0.25	0.62 ± 0.29	0.73 ± 0.23	0.64 ± 0.26

Table 5.3: Precision/recall/F1-Scores for leave-one-participant-out evaluations (left-hand). And per-participants 250 times stratified random split. It is visible that cross-participant scores are suboptimal and not practical, while a per-participant model might be usable for practical purposes.

5.5 INFORMED WORKFLOW RECOGNITION

For investigating if protocol steps can be detected from action recognition, protocols can be modeled as Hidden Markov Models (HMMs), where hidden states correspond to steps in the experiment's protocol, and observations map to the actions executed in that steps as defined in Table 5.2. The upper right of Figure 5.13 depicts this graphically. Protocol steps were marked according to the overhead video recording, which serves as the ground-truth and represents our detection target.

Methdology. The HMM's transition probabilities were set to mimic a linear chain, with a high probability for staying in the same state (workflow step) and a non-zero probability to switch to the next step. This models the linear nature of a protocol execution. The emission probabilities for each state can be generated by uniformly distributing the occurrence of each action in the state's action set. For example, the *solvent* step (cf. Table 5.2) has high emission probabilities for the pouring, transfer, pipetting and stirring actions. The *detection* step in contrast only has a high probability for inverting. To account for possible mis-classifications of the kNN-detector, each action has a low occurrence probability in *each* workflow step. This represent a layered approach, in which the first layer detects actions from wrist movement via a kNN-detector and the second layer detects the workflow steps of the protocol via a HMM.



#	task	precision	recall	F1-score
1	solvent	0.64 ± 0.17	0.93 ± 0.18	0.73 ± 0.15
2	cutting	0.88 ± 0.23	0.83 ± 0.28	0.84 ± 0.25
3	mixing	0.69 ± 0.31	0.49 ± 0.21	0.52 ± 0.18
4,5	waterbath	0.27 ± 0.37	0.09 ± 0.19	0.10 ± 0.20
6	pestling	0.64 ± 0.30	0.88 ± 0.25	0.72 ± 0.27
7	filtrate	0.62 ± 0.40	0.09 ± 0.21	0.12 ± 0.20
8	pouring	0.86 ± 0.23	0.79 ± 0.24	0.81 ± 0.22
9	detection	0.62 ± 0.36	0.75 ± 0.40	0.65 ± 0.35
		0.65 ± 0.35	0.61 ± 0.41	0.56 ± 0.36

Table 5.4: Per-Participant scores for workflow step detection based on a Hidden-Markov Model, combined with k-Nearest Neighbor detection of actions. It is visible that classification scores vary between participants. Some steps are not detectable (waterbath, filtrating) since they have no definable and therefore detectable actions/observations (cf. Table 5.2).

Results. For the evaluation, we detected the workflow of each participant with the above-described kNN-HMM approach. We compared this workflow with the data gathered by the interaction log, i.e. which step was executed when. Assuming that participants had the currently executed step also active on their display, we could also say that we check whether the *currently executed step* was detectable. The result of this evaluation can be seen in Table 5.4. The confusion matrix shows that the *mixing* and *solvent* step are most often confused, which is due to the fact that they have almost the same action set. Only an additional (hardly detectable) *transferring* action distinguishes them. With the presented layered approach, a mean F1-Score of 56% for detecting workflow steps from wrist movements is achievable. This however includes workflow steps which have an empty action set, and are therefore difficult to detect. These steps include the *waterbath* and *filtrate* step, which for instance did not have definable activities linked to them: excluding these steps from the calculation improves the mean F1-score to 71%. It is important to note, however, that such steps do occur in real wet lab experiments and therefore demand complementary detection approaches (e.g. object detection through RFID-markers).

Discussion. The presented layered approach can not only be used to filter wrist movement data on a time-based scale, but also to integrate different sources of information. Therefore, it lends itself well to

integrating further sensors. For example, the current observation vector includes only the detected actions from wrist motion. Additionally detected objects, or head movement based action detection could be easily integrated by augmenting the observation vector. Instrumented object use, e.g. a pipette that detects its usage and details about its setting would be further additional information sources. One shortcoming of the presented HMM approach is that the actions set is assumed to be not ordered, i.e. it does not matter in which order the actions are executed. This might be important information, that is not directly modeled. In this case, conditional random fields (CRF) might prove to be a more suitable alternative. To be practical, a system like the one presented here would need to be either continuously re-learning motion sequences, or limit itself to actions that have proven to be detectable across users such as *inverting*, *pestling* and *pipetting*.

The system could integrate many other features such as maintenance and resource allocation ("Is the centrifuge available for the next five hours?", "Is compound X in store?") across multiple lab members, and tools such as reminders for lengthier procedures or concentration calculators. Provisions to avoid cross-contamination through shared lab equipment (e.g. flasks, pipettes) could be extended by recording their usage, which would also allow tracing back contaminations after they have been detected. Retrieval and editing of the workflows after a completed experiment, would provide an extra possibility for the biologist to reflect on the results in detail. A recording infrastructure, such as the one presented in this paper, can be extended to provide a memory extension, detected actions, protocol steps, or used tools can serve as searchable cues for other recordings like video or audio. Furthermore, these cues might be employed to compare repetitions of the same protocol, allowing to quickly spot differences in their execution.

5.6 TRANSITION DETECTION FROM MOTION

Annotating recorded motion and video sequences is a cumbersome task, and usually estimated to take at least twice the duration of the actual video that is to be annotated. Even when only a limited set of pre-defined actions is to be chosen for each scene. For activity recognition, segments of a time-series need to be marked with a beginning timestamp, end timestamp and a text describing the current action. This means that annotators have to scan through the whole

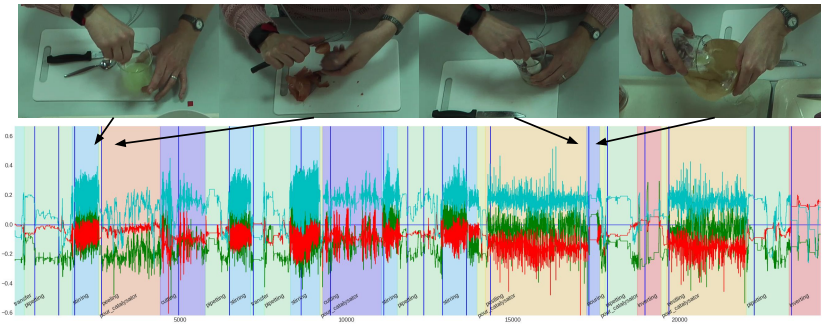


Figure 5.15: Example of a segmented motion-augmented video. Data is taken from the DNA extraction experiment. Ground truth segments are colored blocks in the background, while extracted segments are highlighted as vertical lines.

video frame-by-frame. To accelerate this process, a pre-segmentation based on the recorded sensor data is investigated here. This way, only a summary of each segment needs to be classified by a human annotator (cf. Figure 5.15). Additionally, the identified segments could also be classified by a machine-learned model, potentially providing a novel segmentation approach.

Here, we investigate whether unsupervised clustering algorithms allow to segment motion-augmented videos. Since there is a multitude of different clustering algorithms to choose from, as well as a multitude of parameters for each algorithm it is inevitably challenging to choose these upfront. To find the best combination, a grid-search over all parameter combinations is applied. For this, the *grtool* framework of chapter 3 is executed on a 48-core cluster, in order to keep the required computational time manageable. The tested datasets are all stored in the Matroska video format, to provide a common data-format. Segmentation results are stored in subtitle format.

Methdology. Clustering implementations of *scikit-learn* were tested. This includes the KMeans clustering algorithm, which iteratively approximates k points which equalizes the distance between all data points. Agglomerative clustering provides a bottom-up approach, where cluster are continuously refined based on a distance measure (euclidean here) until k cluster are found. Gaussian Mixture Models (GMM) is a generalization of the KMeans algorithms, in which not only the k central points are determined, but also the (co-)variance of clusters. The DNA

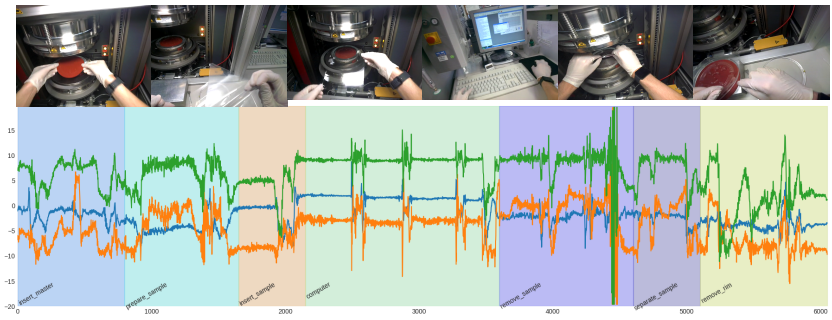


Figure 5.16: Video stills and motion data of the Thermoforming experiments. Video was recorded with Google's Glass, motion data with a wrist-worn Smartwatch.

extraction dataset and a dataset of a Thermoforming process recorded at the Hahn-Schickard-Institute Freiburg was used. The thermoforming process for lab-on-a-chip disks is observed with the head-worn Google's Glass camera, head-worn and wrist-worn (dominant hand) inertial motion data at 50Hz. The process consists of seven different steps (see Figure 5.16) and was executed by two participants.

Unlabelled segments (e.g. *NULL* labels) are removed. While, under more naturalistic circumstances, such segments do exists we limit ourselves here to sequences that are fully classified. A standard sliding window (no overlap) segmentation of variable length w is applied. Mean, variance, min-max range and median are extracted as the feature set for each segment. Each combination of features is tested. For KMeans and agglomerative clustering no cross-validation test is applied, as there is no learning phase. For GMM exhaustive leave-one-participant-out cross-validation is applied. The evaluations were run for the following varied parameters: *Window length* between 200 ms to 2000 ms in 200ms steps. *Feature sets* of mean, variance, min-max range and median, and an *error margin* of 1 to 5 samples.

Results. The result of this evaluation can be found Table 5.5, in which the recall score is reported for each dataset. We concentrate solely on recall here, since it is only important to correctly find segment, not their particular "content". To score this performance, the time-series ground truth data is compared to the clustered time-series segments. An *error* margin, which is the acceptable offset between ground truth and prediction, allows for slight shifts of the prediction. We therefore

	win / feat / marg	recall 1 / 2
KMeans	80 / mean / 4	.92 / .95
Agglo.	90 / time / 5	.93 / .95
GMM	90 / mean / 3	.88 / .95

Table 5.5: Top scoring parameter combinations for all data sets (1 / 2) per method. It is visible that a parameter combination that works well for all datasets can be chosen.

define a True Positive (TP) if a transition is found by clustering within a maximum of 5 windows, a False Positive (FP) if there was no ground truth transition but a prediction, a False Negative (FN) if there was a ground truth transition but no prediction transitions and a True Negative (TN) for other samples.

Just by clustering the mean wrist acceleration, single steps can already be distringuished. A time window of 2s and an error margin of 4 windows works surprisingly well for all datasets, i.e. for more than 86% of segments, the edges are correctly identified. This is a rather surprising result, since usually much more elaborate methods need to be employed to provide good recognition results. However, our goal was not to identify particular steps in a protocol but simply detect significant changes which indicate a possible transition to a different step. This is a much simpler problem, hence the surprisingly good results from this basic approach. Still, this can provide transition marks for a potential automatic labeling system that provides indices for archival video footage or documentation, supporting skipping over uneventful video segments which contain little changes in wrist motion.

5.7 SUMMARY

This chapter presented a wearable system to support experimenters in wet-lab environments by multiple techniques: (1) through memory augmentation by activity-indexed videos, and (2) through task guidance by detection of workflow steps. Storing body-worn sensor recordings, including video, audio, motion and object detection side-by-side in a standardized multi-media container provides a common way to achieve a robust, and distributed database of manual processes. Results from ground truth annotation of videos, as well as results of machine learned

models of motion data are stored in subtitles. Hence, creating the basis for indexing these videos.

A prototype, based on Google's Glass and a wrist-worn inertial data logger, was used to capture experiments and process steps, navigate back and forth those steps, and mark them as done. An analysis of the challenges, as well as the acceptance and wearability of the system, based on "in-situ" observations in several different microbiology laboratories was conducted. The original motivation of using Glass as a recording and guidance tool for an experimenter was tested for feasibility in the presented user study: 22 novice participants were asked to complete an interleaved entry-level DNA-extraction experiment. Participants solely relied on Glass for guidance on the procedure and were all able to finish their experiment successfully.

For object detection, a wireless, wrist-worn RFID reader connected to Google's Glass was tested, particularly whether *implicit* or *explicit* interaction is preferred. A user study of seven participants, indicated that implicit interaction is the preferable way for identifying RFID-tagged test tubes.

Participants' wrist motions were recorded throughout all experiments for studying whether actions made during experiments can be recognized, as well as used, for example in navigating continuous video recordings of procedures. Actions were detected with a k-Nearest-Neighbor classifier, of which only a limited set could be detected reliably (per-participant, F1-score $> 80\%$). A Hidden Markov Model, built by extracting action sets for each protocol step from the digitized protocol, was used for detecting the currently executed step. This layered approach allowed to reconstruct the majority of experiment steps afterwards.

A pre-segmentation step, which splits videos into segments of little to no change of current actions was investigated. Unsupervised clustering algorithms, even on a simple mean of wrist acceleration, already provides segments that are similar to human annotation. In more than 86% of cases, a crop mark resulting from such unsupervised approach is at maximum 4 windows away from a crop mark placed by a human annotator. The approach was tested on two datasets: the original DNA extraction experiment and a thermoforming process for lab-on-a-chip systems. This provides evidence that applying a clustering to the original motion time-series, can already provide a practical segmentation of motion-augmented videos, and serve as simple navigation cues, and increase the efficiency of subsequent human annotation.

The ability to capture and review an experiment up to several weeks later is in wet labs more important than the actual guidance - when the experimenter finds out that something went wrong with the experiment, a detailed review of executed steps could shed light on the cause. Moreover, a wearable and touch-free system does not only decrease the chance of contamination, it also provides the means to interact with a computing system right on the spot, in turn minimizing the required interaction efforts. This is also applicable to other applications, involving manual processes that can be detected with wearable and instrumented artifacts, for example cooking in a kitchen, maintaining machinery or assembling products. The proposed concept of motion- and action-augmented multi-media files is general enough to encompass these applications as well, and the experiments have shown that (1) a simple clustering can provide useful crop marks, and (2) that procedural knowledge, like the sequence of actions and used tools can be encoded and detected with a hidden Markov Model.

This chapter contains contributions from the following peer-reviewed publications:

Philipp Marcel Scholl and Kristof Van Laerhoven. "Wearable digitization of life science experiments." In: *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing Adjunct Publication - UbiComp '14 Adjunct* (2014).

Philipp Marcel Scholl, Matthias Wille, and Kristof Van Laerhoven. "Wearables in the Wet Lab: A Laboratory System for Capturing and Guiding Experiments." In: *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. 2015.

Philipp Marcel Scholl and Kristof Van Laerhoven. "RFID-Based Compound Identification in Wet Laboratories With Google Glass." In: *WOAR '15 Proceedings of the 2nd international Workshop on Sensor-based Activity Recognition and Interaction*. 2015.

Philipp Marcel Scholl and Kristof Van Laerhoven. "Remind - Towards a Personal Remembrance Search Engine for Motion Augmented Multi-Media Recordings." In: *MUM 2016 - Proceedings of the international 2016 Conference on Multimedia and Ubiquitous Systems*. 2016.

CONCLUSION

Human Activity Recognition from (body-worn) sensors can provide the cues for human memory augmentation, and the foundation for an objective quantification of human behaviour. Technically, a projection from continuous sensor recordings to a sequence of activity descriptions has to be found. This requires to sample the space of possible executions of each activity as close to reality as possible: participants should behave as natural as possible, no external observer should influence that recording, and the scenarios should not include any arbitrary limitations. Often, this results in annotating datasets from video recordings, or by momentary assessments which both remind participants of them being observed, or break the natural process of an activity. The core idea of this thesis is to trivialise the annotation process by instrumenting artifacts that are solely used for a particular activity. This way reliable ground-truth annotation, with minimal interruption of study participants, can be extracted. Detecting smoking with a Smartlighter and wrist motion, detecting manual processes in a wetlab, and a software framework for combining the recorded data, validates this idea.

6.1 SUMMARY OF CONTRIBUTIONS

The findings of this thesis touches multiple facets of Activity Recognition systems. These are summarized in the following paragraphs:

Unix Framework for Activity Recognition. The idea of encapsulating each step of an Activity Recognition Chain (ARC) as a separate Unix process, and combine them to a detection chain, is proposed. Command line argument represent the hyper-parameters of the overall recognition, and this approach allows for effortless parallelization. As such providing a flexible approach to quickly test novel machine-learning based activity recognition systems.

Format for data exchange. A proposal for curating, exchanging and querying based on a multi-media container is described. This format allows to store multiple sensor streams in an interleaved and time-synchronized manner, and supports optimized storage through compression as well as streaming. The use of audio compressors for inertial motion data shows that a limited processing overhead can be traded for efficient storage and transmission. This is compared to the commonly used CSV format. The container format also stores meta-data, which includes sample format and sample rate amongst other details.

Detection of Inertial Motion Modality. For inertial motion data, the sensor modality was shown to be detectable from the data itself with a rule-based system in 98% of cases for a database of five human activity recognition datasets. This provides additional safety when recording datasets, as well as reconstructing meta-data post-experiment in cases this information got lost.

Design of a Smartlighter. Detecting smoking continuously, over long-term, with minimal user interference was often attempted by detection with body-worn sensors. Instrumenting the lighter used for lighting cigarettes provides a energy-efficient, and reliable alternative. The thesis describes several alternative implementations, with a final completely wireless solution.

Detection of smoking gestures. The detection of smoking from inertial wrist motion data, validated against the detection by a Smartlighter, is an example of the thesis' core idea. The lighter is connected to a Smartphone and Smartwatch ensemble for collecting 351 instances of smoking from six participants. By using the Smartlighter to collect ground-truth the actual detection process is trivialized. A symbolic detection approach that is devised from smoking topography can only

detect *trivial* smoking gestures. This was shown by selecting a subset of the data, and showing that the symbolic detection achieves a high F1-score of 82%, but drops sharply when applied to the full dataset. A hyper-parameter optimized machine-learning approach achieves an average F1-score of 84% on the full dataset, showing that a machine learned model can indeed pick up non-trivial smoking gestures. Challenges in detecting smoking are discussed in chapter 4 as well.

Comparison to questionnaire elicited smoking behaviour. A study observing eleven smokers, for a mean time of eleven days has shown that smokers over-estimate their consumption, and have a hard time telling the most probable time-of-day when consuming a cigarette. For this study, smokers were assessed with a questionnaire pre- and post-observation, and continuously observed with a Smartlighter. This illustrates that a continuous, sensor-based assessment of smoking can provide novel insight for cessation research, and probably fuel novel mobile cessation techniques.

Evaluation of Google Glass as a wetlab recording tool. Google Glass was used in a university-level wetlab training session to elicit requirements for a wearable support system used in such environments. In total eight groups of students used Google Glass to document their experiment by recording their findings.

Object detection with wrist-worn RFID sensors. A wrist-worn RFID reader is presented, that is able to detect objects and tools used during an ongoing wetlab experiment with little to no user interaction. The solution was found to be insufficient for practical use, however a study encompassing nine participants showed that an RFID-activated detection is preferred over a voice-activated one, when including Google Glass as the interface.

Evaluation of Google Glass as Guidance tool. In a study encompassing twenty-two novice participants an entry-level microbiology experiment was executed. Google Glass was used solely for guiding participants through the experiment, which was found sufficient for almost everyone. This also provided the recordings for the subsequent detection of activities from body-motion.

Detection of wetlab activities with body-worn sensors. A hierarchical detection system for the actions in wetlab experiments based on inertial wrist motion is proposed. Atomic actions, like pestling and pipetting, are detected first and the actual executed process is modelled as a Hidden Markov Model that was extracted from the textual description of the experiment. This model allowed to detect the pro-

cess step with an F1-score of $> 80\%$ when trained on a per-participant model.

Evaluation of unsupervised motion-augmented video segmentation. An unsupervised video segmentation approach which can summarize recorded material without any prior knowledge is proposed. In 86% of all cases, a crop mark resulting from such unsupervised approach is at maximum four windows away from a crop mark placed by a human annotator.

In total a system for recording manual tasks with body-worn sensors, in a standardized container format, for analysis with a Unix framework of machine learning tools, applied on smoking and wetlab action recognition was investigated.

6.2 OUTLOOK ON FUTURE WORK

Collecting ground truth data by instrumenting artifacts is one of the core ideas, which were explored in this thesis. By limiting the detection of activities to tool usage, or their combined usage, the actual detection is trivialized and therefore allows to *automate* the process of data collection. In turn, this allows to replace these highly specialized sensors with more general ones, like motion recorders or video cameras and solves the issue of having to (manually) collect large datasets.

The space of instrumented artifacts was, however, only explored in the setting of smoking and in wetlab environments. Further health scenarios, were specific tools are used, e.g. fitness instruments like dumbbells, can be instrumented to collect larger datasets. Specifically for smoking, a dataset that encompasses a multitude of different smoking styles with a large number participants in a naturalistic setting is still missing. For the wetlab environment, further exploration of instrumentable tools could benefit possible activity recognition, and memory augmentation scenarios. One example would be a pipette, that records pipetting events, dispensed amount of liquid, and the dispensing location. An even more concrete example would be the detection of cooking, where kitchen tools, like knives, mixers, spoons, stoves, and ovens would provide observations of the task at hand.

One important practical consideration is an estimation of the *quality* of a dataset. A ratio that puts the theoretically possible combinations of sensor data in relation to the amount of possible executions could provide such an insight. For example, single smoking puffs detected

over a window of 3s, recorded with a three axis accelerometer at 10bit resolution and at 10Hz, already encompasses 30720 unique sensor data patterns for a *single* execution of this gesture. The question is, how many of these unique patterns are sampled in a particular dataset, to asses its quality. A comparison method based on such a ratio could provide confidence in particular datasets.

Storing body-worn data recordings in multi-media containers, and building a remembrance agents with head-worn displays would be a further exploration area. A recording software, similar to video recorders on a mobile phone, that additionally samples sensor data from the local device and from wirelessly connected body-worn devices, like a Smartwatch or Smartglass, which stores this data in a multi-media file could be further explored. These files can either be streamed live to other devices, or transferred later and subsequently analyzed for activities and other patterns. Such analysis is achieved by classification systems that work on one of the streams provided in the multi-media file, and create a time-coded meta-data stream that is added to the file. These time-coded meta-data information is then a categorical space that can be efficiently searched. This setup provides the foundation for in-situ task guidance, and facilitates review of recorded workflows.

LIST OF FIGURES

LIST OF TABLES

BIBLIOGRAPHY

[1] Vannevar Bush. "As We May Think." In: *Atlantic Monthly* July (1945), pp. 112–124.

[2] Oscar D. Lara and Miguel a. Labrador. "A Survey on Human Activity Recognition using Wearable Sensors." In: *IEEE Communications Surveys & Tutorials* 15.3 (2013), pp. 1192–1209.

[3] Akin Avci et al. "Activity Recognition Using Inertial Sensing for Healthcare, Wellbeing and Sports Applications: A Survey." In: *Proc. of the 23th Int. Conf. on Architecture of Computing Systems* (2010), pp. 167–176. URL: http://doc.utwente.nl/70138/.

[4] Andreas Bulling, Ulf Blanke, and Bernt Schiele. "A tutorial on human activity recognition using body-worn inertial sensors." In: *ACM Computing Surveys (CSUR)* 46.3 (2014), pp. 1–33. URL: http://dl.acm.org/citation.cfm?doid=2578702.2499621.

[5] Peter Norvig. *As we may Program - Google Tech Talk*. 2016.

[6] Matthias Kranz et al. "The mobile fitness coach: Towards individualized skill assessment using personalized mobile devices." In: *Pervasive and Mobile Computing* (2012). URL: http://dx.doi.org/10.1016/j.pmcj.2012.06.002.

[7] Eduardo Velloso et al. "Qualitative Activity Recognition of Weight Lifting Exercises." In: *Augmented Human*. 2013.

[8] Cassim Ladha, Nils Y Hammerla, and Patrick Olivier. "ClimbAX: Skill Assessment for Climbing Enthusiasts." In: *UbiComp '13: Proceedings of the 15th International Conference on Ubiquitous Computing*. 2013.

[9] Ernst A Heinz et al. "Using Wearable Sensors for Real-time Recognition Tasks in Games of Martial Arts – An Initial Experiment." In: *IEEE Symposium on Computational Intelligence and Games*. 2006.

[10] Miikka Ermes and Ilkka Korhonen. "Detection of Daily Activities and Sports With Wearable Sensors in Controlled and Uncontrolled Conditions." In: *IEEE transactions on information technology in biomedicine* 12.1 (2008), pp. 20–26.

[11] Thomas Fritz et al. "Persuasive Technology in the Real World : A Study of Long-Term Use of Activity Sensing Devices for Fitness." In: *CHI '14 Proceedings of the SIGCHI Conference on Human Factors in Computing Systemg*. 2014, pp. 487–496.

[12] Sunny Consolvo et al. "Activity Sensing in the Wild: A Field Trial of UbiFit Garden." In: *CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*. October 2017. 2008.

[13] Tony Jebara et al. "Stochasticks: Augmenting the Billiards Experience with Probabilistic Vision and Wearable Computers." In: *International Symposium on Wearable Computers*. 1997.

[14] Joseph A Paradiso and Eric Hu. "Expressive Footwear for Computer-Augmented Dance Performance." In: *International Sym*. October. 1997, pp. 20–21.

[15] Marc Bächlin, Kilian Förster, and Gerhard Tröster. "SwimMaster: A Wearable Assistant for Swimmer." In: *Ubicomp '09: Proceedings of the 11th international conference on Ubiquitous computing*. 2009, pp. 215–224.

[16] Thad Eugene Starner, Bernt Schiele, and Alex Pentland. "Visual context awareness in wearable computing." In: *International Symposium on Wearable Computers1*. 1998.

[17] Alexandros Pantelopoulos and Nikolaos G Bourbakis. "A Survey on Wearable Sensor-Based Systems for Health Monitoring and Prognosis." In: *IEEE Transactions on Systems, Man and Cybernetics - Part C: Applications and Reviews* 40.1 (2010), pp. 1–12.

[18] Eunju Kim, Sumi Helal, and Diane Cook. "Human Activity Recognition and Pattern Discovery." In: *IEEE Pervasive Computing* 9.1 (2010), pp. 1–10.

[19] T Ã Huynh. "Human Activity Recognition with Wearable Sensors." PhD thesis. 2008.

[20] Tim Van Kasteren et al. "Accurate Activity Recognition in a Home Setting." In: *UbiComp '08 Proceedings of the 10th international conference on Ubiquitous computing* (2008), pp. 1–9.

[21] Emmanuel Munguia Tapia, Stephen S Intille, and Kent Larson. "Activity Recognition in the Home Using Simple and Ubiquitous Sensors." In: *Pervasive Computing* 3001 (2004), pp. 158–175. arXiv: 9780201398298.

[22] Narayanan C. Krishnan and Diane J. Cook. "Activity recognition on streaming sensor data." In: *Pervasive and Mobile Computing* 10.PART B (2014), pp. 138–154.

[23] Jonathan Lester, Tanzeem Choudhury, and Gaetano Borriello. "A Practical Approach to Recognizing Physical Activities." In: *Pervasive Computing* 3968 (2006), pp. 1–16. URL: http://www.springerlink.com/index/7048888592382352.pdf.

[24] Stephen J. Preece et al. "A comparison of feature extraction methods for the classification of dynamic activities from accelerometer data." In: *IEEE Transactions on Biomedical Engineering* 56.3 (2009), pp. 871–879.

[25] Stephen S Intille. "A new research challenge: persuasive technology to motivate healthy aging." In: *IEEE transactions on information technology in biomedicine : a publication of the IEEE Engineering in Medicine and Biology Society* (2004). URL: http://www.ncbi.nlm.nih.gov/pubmed/15484427.

[26] Maja Stikic et al. "ADL Recognition Based on the Combination of RFID and Accelerometer Sensing." In: *Pervasive Computing Technologies for Healthcare - PervasiveHealth*. 2008.

[27] Daniele Riboni et al. "SmartFABER: Recognizing fine-grained abnormal behaviors for early detection of mild cognitive impairment." In: *Artificial Intelligence In Medicine* 67 (2015), pp. 57–74. URL: http://dx.doi.org/10.1016/j.artmed.2015.12.001.

[28] Philipp Marcel Scholl and Kristof Van Laerhoven. "A Feasibility Study of Wrist-Worn Accelerometer Based Detection of Smoking Habits." In: *International Workshop on Extending Seamlessly to the Internet of Things*. 2012.

[29] Abhinav Parate et al. "RisQ: Recognizing Smoking Gestures with Inertial Sensors on a Wristband." In: *Proceedings of the 12th ACM annual international conference on Mobile systems, applications, and services*. 2014.

[30] Edward Sazonov et al. "RF hand gesture sensor for monitoring of cigarette smoking." In: *2011 Fifth International Conference on Sensing Technology* (2011). URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6137014.

[31] Muhammad Shoaib, Hans Scholten, and Paul J M Havinga. "A Hierarchical Lazy Smoking Detection Algorithm Using Smartwatch Sensors." In: *International Conference on e-Health Networking, Applications and Services (Healthcom)*. 2016.

[32] Qu Tang et al. "Automated Detection of Puffing and Smoking with Wrist Accelerometers." In: *Pervasive Health*. 2014.

[33] Yang Qin et al. "Identifying Smoking from Smartphone Sensor Data and Multivariate Hidden Markov Models." In: *International Conference on Social Computing, Behavioral-Cultural Modeling and Prediction and Behavior Representation in Modeling and Simulation*. 2017, pp. 230–235.

[34] Xiaolong Zheng et al. "Smokey: Enabling Ubiquitous Smoking Detection with Commercial WiFi Infrastructures." In: *Infocom* Cv (2016).

[35] Casey Cole et al. "Resolving Ambiguities in Accelerometer Data Due to Location of Sensor on Wrist in Application to Detection of Smoking Gesture." In: *IEEE EMBS International Conference on Biomedical & Health Informatics (BHI)*. 2017.

[36] Oliver Amft, Holger Junker, and Gerhard Tröster. "Detection of eating and drinking arm gestures using inertial body-worn sensors." In: *Proceedings - International Symposium on Wearable Computers, ISWC* 2005 (2005), pp. 160–163.

[37] Abdelkareem Bedri et al. "EarBit: Using Wearable Sensors to Detect Eating Episodes in Unconstrained Environments." In: *ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1.3 (2017), pp. 1–20.

[38] Tracy L. Burrows, Rebecca J. Martin, and Clare E. Collins. "A Systematic Review of the Validity of Dietary Assessment Methods in Children when Compared with the Method of Doubly Labeled Water." In: *Journal of the American Dietetic Association* 110.10 (2010), pp. 1501–1510. URL: http://dx.doi.org/10.1016/j.jada.2010.07.008.

[39] Yujie Dong et al. "A new method for measuring meal intake in humans via automated wrist motion tracking." In: *Applied Psychophysiology Biofeedback* 37 (2012), pp. 205–215.

[40] Abdelkareem Bedri et al. "Detecting Mastication: A Wearable Approach." In: *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction* (2015), pp. 247–250. URL: http://doi.acm.org/10.1145/2818346.2820767.

[41] Andrea Mannini and Angelo Maria Sabatini. "Machine Learning Methods for Classifying Human Physical Activity from On-Body Accelerometers." In: *Sensors* 10 (2010), pp. 1154–1175.

[42] Edward Sazonov et al. "Non-invasive monitoring of chewing and swallowing." In: *Physiological measurement* (2008).

[43] Oliver Amft and Paul Lukowicz. "Analysis of Chewing Sounds for Dietary Monitoring." In: *UbiComp '08 Proceedings of the 7th international conference on Ubiquitous computing*. 2005, pp. 1–16.

[44] Christopher Merck et al. "Multimodality Sensing for Eating Recognition." In: *Proceedings of the 10th EAI International Conference on Pervasive Computing Technologies for Healthcare*. 2016.

[45] Ling Bao and Stephen S Intille. "Activity Recognition from User-Annotated Acceleration Data." In: *Pervasive Computing* (2004), pp. 1–17.

[46] Jennifer R Kwapisz, Gary M Weiss, and Samuel A Moore. "Activity Recognition using Cell Phone Accelerometers." In: *ACM SigKDD Explorations Newsletter* 12.2 (2010), pp. 74–82.

[47] Andrea Mannini et al. "Activity Recognition Using a Single Accelerometer Placed at the Wrist or Ankle." In: *Medicine and science in sports and exercise* (2013).

[48] Daqing Zhang and Shijian Li. "Activity Recognition on an Accelerometer Embedded Mobile Phone with Varying Positions and Orientations." In: *Ubiquitous intelligence and computing* (2010).

[49] Jens Barth et al. "Biometric and Mobile Gait Analysis for Early Diagnosis and Therapy Monitoring in Parkinson's Disease." In: *IEEE EMBS International Conference on Biomedical & Health Informatics (BHI)*. 2011, pp. 868–871.

[50] Physicians Share et al. "Predicting the Potential of Wearable Technology." In: *IEEE Engineering in Medicine and Biology Magazine* June (2003), pp. 23–27.

[51] Bruce H Dobkin and Andrew Dorsch. "The Promise of mHealth: Daily Activity Monitoring and Outcome Assessments by Wearable Sensors." In: *Neurorehabilitation and Neural Repair* (2011).

[52] Kristof van Laerhoven et al. "Medical Healthcare Monitoring with Wearable and Implantable Sensors." In: *Proc. of the 3rd International Workshop on Ubiquitous Computing for Healthcare Applications*. January. 2004.

[53] Shyamal Patel et al. "A review of wearable sensors and systems with application in rehabilitation." In: *Journal of NeuroEngineering and Rehabilitation* (2012), pp. 1–17.

[54] Atanu Roy Chowdhury, Shyamal Patel, and Paolo Bonato. "Mercury: A Wearable Sensor Network Platform for High-Fidelity Motion Analysis." In: *SenSys'09*. 2009.

[55] Venet Osmani. "Smartphone Based Recognition of States and State Changes in Bipolar Disorder Patients." In: *IEEE Journal of Biomedical and Health Informatics* (2014), pp. 1–8.

[56] Melodie Vidal et al. "Wearable Eye Tracking for Mental Health Monitoring." In: *Computer Communications* 35 (2012), pp. 1306–1311.

[57] Alireza Sahami Shirazi et al. "Already up? using mobile phones to track ampamp; share sleep behavior." In: *Journal of Human Computer Studies* 71.9 (2013), pp. 878–888. URL: http://dx.doi.org/10.1016/j.ijhcs.2013.03.001.

[58] E K Choe et al. "Opportunities for computing technologies to support healthy sleep behaviors." In: *Proceedings of the 2011 annual conference on Human factors in computing systems*. ACM. 2011, pp. 3053–3062.

[59] Akane Sano and Rosalind W Picard. "Stress Recognition using Wearable Sensors and Mobile Phones." In: *Humaine Association Conference on Affective Computing and Intelligent Interaction*. 2013.

[60] Emil Jovanov et al. "Stress Monitoring Using a Distributed Wireless Intelligent Sensor System." In: *IEEE Engineering in Medicine and Biology Magazine* June (2003), pp. 49–55.

[61] Marc Bächlin et al. "Wearable assistant for Parkinson's disease patients with the freezing of gait symptom." In: *IEEE Transactions on Information Technology in Biomedicine1* January 2010 (2010).

[62] Charles Abraham and Susan Michie. "A taxonomy of behavior change techniques used in interventions." In: *Health Psychology* 27.3 (2008), pp. 379–387. URL: http://doi.apa.org/getdoi.cfm?doi=10.1037/0278-6133.27.3.379.

[63] Jing Zhao, Becky Freeman, and Mu Li. "Can Mobile Phone Apps Influence People's Health Behavior Change? An Evidence Review." In: *Journal of Medical Internet Research* 18 (2016), pp. 1–12.

[64] Stuart J H Biddle, Trish Gorely, and David J Stensel. "Health-enhancing physical activity and sedentary behaviour in children and adolescents." In: *Journal of Sport Sciences* (2004), pp. 679–701.

[65] Janet Buckworth and Claudio Nigg. "Physical Activity, Exercise, and Sedentary Behavior in College Students." In: *Journal of American College Health* 53.1 (2004).

[66] Jack Shen-Kuen Chang et al. "The Heroes' Problems: Exploring the Potentials of Google Glass for Biohazard Handling Professionals." In: *Extended Abstracts of the ACM CHI'15 Conference on Human Factors in Computing Systems* 2 (2015), pp. 1531–1536. URL: http://dx.doi.org/10.1145/2702613.2732698.

[67] David Minnen et al. "Recognizing Soldier Activities in the Field." In: *Proceedings of International IEEE Workshop on Wearable and Implantable Body Sensor Networks (BSN)* 13 (2007), pp. 236–241.

[68] Paul Lukowicz et al. "WearITwork: Toward Real-World Industrial Wearable Computing." In: *IEEE Pervasive Computing* (2007).

[69] Shahram Jalaliniya and Thomas Pederson. "Designing Wearable Personal Assistants for Surgeons: An Egocentric Approach." In: *Pervasive Computing* (2015), pp. 22–31.

[70] Lauren Kolodzey et al. "Wearable technology in the operating room: a systematic review." In: *BMJ Innov* (2016), pp. 1–9.

[71] Bennie Lindeque et al. "Emerging Technology in Surgical Education: Combining Real-Time Augmented Reality and Wearable Computing." In: *Orthopedics* (2014).

[72] Agnes Grünerbl et al. "Monitoring and Enhancing Nurse Emergency Training with Wearable Devices." In: *UbiComp '15: Proceedings of the 17th International Conference on Ubiquitous Computing*. 2015, pp. 1261–1267.

[73] Allan Stisen and Henrik Blunck. "Handheld Versus Wearable Interaction Design for Professionals." In: *OZCHI*. 2014.

[74] Naonori Ueda and Sozo Inoue. "Mobile Activity Recognition for a Whole Day: Recognizing Real Nursing Activities with Big Dataset." In: *UbiComp '15: Proceedings of the 17th International Conference on Ubiquitous Computing*. 2015, pp. 1269–1280.

[75] Susan P Mcgrath et al. *ARTEMIS: A Vision for Remote Triage and Emergency Management Information Integration*. 2003.

[76] Shahram Jalaliniya, Thomas Pederson, and Diako Mardanbengi. "A Wearable Personal Assistant for Surgeons - Design, Evaluation, and Future Prospects." In: *EAI Endorsed Transactions on Pervasive Health and Technology* 3.1 (2017), pp. 1–13.

[77] Anind K Dey et al. "The Conference Assistant: Combining Context-Awareness with Wearable Computing." In: *International Symposium on Wearable Computers*. 1999.

[78] Werner Geyer, Heather Richter, and Gregory D Abowd. "Towards a Smarter Meeting Record—Capture and Access of Meetings Revisited." In: *Multimedia Tools and Applications* (2005), pp. 393–410.

[79] Bradley J Rhodes. "The Wearabte Remembrance Agent: A System for Augmented Hemory." In: *International Symposium on Wearable Computers*. 1997.

[80] M Iftekhar Tanveer, Emy Lin, and Mohammed Ehsan Hoque. "Rhema : A Real-Time In-Situ Intelligent Interface to Help People with Public Speaking." In: *IUI 2015: Proceedings of the 20th International Conference on Intelligent User Interfaces* (2015), pp. 286–295.

[81] Thomas Stiefmeier et al. "Wearable Activity Tracking in Car Manufacturing." In: *IEEE Pervasive Computing* 7.2 (2008), pp. 42–50. URL: `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4487087`.

[82] Gabriele Bleser et al. "Cognitive learning, monitoring and assistance of industrial workflows using egocentric sensor networks." In: *PLoS ONE* 10.6 (2015), pp. 1–41.

[83] Paul Lukowicz et al. "Recognizing Workshop Activity Using Body Worn Microphones and Accelerometers." In: *Pervasive 2004* (2004), pp. 18–32.

[84] T. Starner. "The challenges of wearable computing." In: *IEEE Micro* (2001).

[85] Thad Starner. "The challenges of wearable computing: Part 2." In: *IEEE Micro* 21.4 (2001), pp. 54–67.

[86] Albrecht Schmidt. "Implicit Human Computer Interaction Through Context." In: *Personal technologies* (2000), pp. 1–10.

[87] Mark Weiser. "The Computer of the 21st Century." In: *Scientific American* 265.3 (1999), pp. 94–105.

[88] Gregory D Abowd and Elizabeth D Mynatt. "Charting Past, Present, and Future Research in Ubiquitous Computing." In: *ACM Transactions on Computer-Human Interaction (TOCHI)* 7.1 (2000), pp. 29–58.

[89] Thad Eugene Starner. "Wearable Computing and Contextual Awareness." In: 1991 (1999).

[90] Kristof van Laerhoven and Oliver Amft. "What Will We Wear After Smartphones?" In: *Pervasive Computing* (2017).

[91] World Health Organization (WHO). *World Report on Ageing and Health*. Tech. rep. 2015.

[92] World Health Organization (WHO). *World Health Report: Reducing Risks, Promoting Healthy Life*. Tech. rep. 2002.

[93] Nancy F Butte et al. "Assessing Physical Activity Using Wearable Monitors: Measures of Physical Activity." In: *Medicine and science in sports and exercise* (2012), pp. 5–12.

[94] Urs Anliker et al. "AMON: A Wearable Multiparameter Medical Monitoring and Alert System." In: *IEEE Transactions on Information Technology in Biomedicine* 8.4 (2004), pp. 415–427.

[95] Meir Plotnik et al. "Wearable Assistant for Parkinson's Disease Patients With the Freezing of Gait Symptom." In: *IEEE Transactions on Information Technology in Biomedicine* 14.2 (2010), pp. 436–446.

[96] Delsey M Sherrill et al. "Using hierarchical clustering methods to classify motor activities of COPD patients from wearable sensor data." In: *Journal of NeuroEngineering and Rehabilitation* 14 (2005), pp. 1–14.

[97] Nicky Kern et al. "Wearable Sensing to Annotate Meeting Recordings." In: *Personal and Ubiquitous Computing* (2003).

[98] Michael Beigl. "MemoClip : A Location based Remembrance Appliance MemoClip : A Remembrance Appliance." In: *Personal and Ubiquitous Computing* Figure 1 (2000), pp. 2–6.

[99] Abigail Sellen and Stece Whittacker. "Beyond Total Capture: A Constructive Critique of Lifelogging." In: *Communications of the ACM* 54.5 (2010), p. 70.

[100] Susan Dumais et al. "Stuff I've seen: a system for personal information retrieval and re-use." In: *Proc. 26th annual international ACM SIGIR conference on Research and development in informaion retrieval* 49.2 (2003), pp. 72–79.

[101] Kent Lyons. "Improving support of conversations by enhancing mobile computer input." In: August (2005).

[102] Neil C. Rowe et al. *Automated Assessment of Physical-Motion Tasks for Military Integrative Training*. 2009.

[103] Teesid Leelasawassuk, Dima Damen, and Walterio Mayol-Cuevas. "Automated capture and delivery of assistive task guidance with an eyewear computer: The GlaciAR system." In: (2016). arXiv: 1701.02586. URL: http://arxiv.org/abs/1701.02586.

[104] Markus Aleksy and Mikko J Rissanen. "Utilizing Wearable Computing in Industrial Service Applications." In: *Ambient Intelligence and Humanized Computing (JAIHC)* (2012).

[105] Katrin Jonsson and Ulrika H Westergren. "Technologies for value creation: An exploration of remote diagnostics systems in the manufacturing industry." In: *Information Systems Journal* (2008), pp. 107–131.

[106] Lawrence J Najjar, J Christopher Thompson, and Jennifer J Ockerman. "A Wearable Computer for Quality Assurance Inspectors in." In: *International Symposium on Wearable Computers*. 1997, pp. 163–164.

[107] Eisa Zarepour et al. "Applications and Challenges of Wearable Visual Lifeloggers." In: *IEEE Computer* (2017).

[108] Matthias Lampe, Martin Strassner, and Elgar Fleisch. "A Ubiquitous Computing Environment for Aircraft Maintenance." In: 2004.

[109] J.J. Ockerman and A.R. Pritchett. "Preliminary investigation of wearable computers for task guidance in aircraft inspection." In: *International Symposium on Wearable Computers* (1999), pp. 33–40. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=729527.

[110] Steven Feiner and Steven J Henderson. "Evaluating the Benefits of Augmented Reality for Task Localization in Maintenance of an Armored Personnel Carrier Turret." In: *IEEE International Symposium on Mixed and Augmented Reality*. 2009, pp. 135–144.

[111] Kai Kunze et al. "Does Context Matter ? - A Quantitative Evaluation in a Real World Maintenance Scenario." In: *International Conference on Pervasive Computing*. 2009.

[112] Thomas Stiefmeier et al. "Combining Motion Sensors and Ultrasonic Hands Tracking for Continuous Activity Recognition in a Maintenance Scenario." In: *2006 10th IEEE International Symposium on Wearable Computers* (2006), pp. 97–104. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4067733.

[113] Dan Siewiorek et al. "Adtranz: A Mobile Computing System for Maintenance and Collaboration." In: *International Symposium on Wearable Computers*.

[114] J Rantanen et al. "Smart Clothing Prototype for the Arctic Environment." In: *Personal and Ubiquitous Computing* (2002), pp. 3–16.

[115] Eric Foxlin. "Pedestrian Tracking with Shoe-Mounted Inertial Sensors." In: *IEEE Computer Graphics and Applications* December (2005), pp. 38–46.

[116] C. Fischer and H. Gellersen. "Location and Navigation Support for Emergency Responders: A Survey." In: *IEEE Pervasive Computing* 9 (2010), pp. 38–47.

[117] Christopher E Carr, Steven J Schwartz, and Ilia Rosenberg. "A Wearable Computer for Support of Astronaut Extravehicular Activity." In: *International Symposium on Wearable Computers*. 2002.

[118] Carsten W Mundt et al. "A Multiparameter Wearable Physiologic Monitoring System for Space and Terrestrial Applications." In: *IEEE Transactions on Information Technology in Biomedicine* 9.3 (2005), pp. 382–391.

[119] D Bannach, P Lukowicz, and O Amft. "Rapid Prototyping of Activity Recognition Applications." In: *Pervasive Computing, IEEE* 7.2 (2008), pp. 22–31.

[120] Jamie a. Ward et al. "Activity recognition of assembly tasks using body-worn microphones and accelerometers." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28.X (2006), pp. 1553–1566.

[121] Ulf Blanke and Bernt Schiele. "Remember and Transfer what you have Learned – Recognizing Composite Activities based on Activity Spotting." In: *ISWC '10: International Symposium on Wearable Computing*. 2010.

[122] Jason Pascoe and United Kingdom. "Adding Generic Contextual Capabilities to Wearable Computers." In: *International Symposium on Wearable Computers*. Vol. 44. 0. 1998.

[123] Ron B Yeh et al. "ButterflyNet: A Mobile Capture and Access System for Field Biology Research." In: *Human Factors in Computing Systems*. ACM SIGCHI, 2006, pp. 1–10.

[124] Nuria Oliver, Eric Horvitz, and Adaptive Systems. "Layered Representations for Human Activity Recognition." In: (2002).

[125] Tia Gao et al. "Vital Signs Monitoring and Patient Tracking Over a Wireless Network." In: *Johns Hopkins APL Technical Digest*, 27.1 (2006).

[126] Thomas Wagner et al. "An Application View of COORDINATORS Coordination Managers for First Responders." In: *AAI Emerging Applications* (2004), pp. 908–915.

[127] Joseph A Paradiso and Thad Starner. "Energy Scavenging for Mobile and Wireless Electronics." In: *Pervasive Computing* (2005).

[128] Lia Kvalilashvili and Judi Ellis. "Ecological validity and the real-life/laboratory controversy in memory research: A critical (and historical) review." In: *Ecological Psychology* (2007).

[129] Kenneth Hammond. *Ecological Validity: Then and Now*. URL: `http://www.albany.edu/cpr/brunswik/notes/essay2.html`.

[130] Christian Monrad Nielsen et al. "It's Worth the Hassle! The Added Value of Evaluating the Usability of Mobile Systems in the Field." In: *NordiChi*. October. 2006, pp. 14–18.

[131] Colin F Camerer. "The promise and success of lab-field generalizability in experimental economics: A critical reply to Levitt and List." In: (2011).

[132] Scott Carter et al. "Exiting the Cleanroom: On Ecological Validity and Ubiquitous Computing." In: *Human-Computer Interaction*. October. 1999.

[133] Attila Reiss, Didier Stricker, and Gustaf Hendeby. "Towards robust activity recognition for everyday life: Methods and evaluation." In: *Pervasive Computing Technologies for Healthcare* May (2013), pp. 25–32.

[134] Nils Y Hammerla, Shane Halloran, and Thomas Ploetz. "Deep, Convolutional, and Recurrent Models for Human Activity Recognition using Wearables." In: *Ijcai* (2016), pp. 1533–1540. arXiv: 1604.08880. URL: http://arxiv.org/abs/1604.08880.

[135] Daniel Ashbrook, Kent Lyons, and Thad Starner. "Methods of Evaluation for Wearable Computing." In: *Smart Clothing* (2010), pp. 229–248.

[136] Jesper Kjeldskov and Connor Graham. "A Review of Mobile HCI Research Methods." In: *5th International Mobile HCI conference*. 2003.

[137] Mark Mirtchouk et al. "Recognizing Eating from Body-Worn Sensors: Combining Free-living and Laboratory Data." In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT)* 1.3 (2017), pp. 1–20.

[138] Nathalie Japkowicz and Shaju Stephen. "The class imbalance problem: A systematic study." In: *Intelligent data analysis*. IOS Press (Amsterdam, The Netherlands), 2002.

[139] Jamie Ward, Paul Lukowicz, and Hans Gellersen. "Performance metrics for activity recognition." In: 2.1 (2011), pp. 1–23. URL: http://dx.doi.org/10.1145/1889681.1889687.

[140] David Minnen et al. "Performance Metrics and Evaluation Issues for Continuous Activity Recognition." In: *Proc. Int. Workshop on Performance Metrics for Intelligent Systems* (2006), pp. 141–148.

[141] Nathalie Japkowicz and Mohak Shah. "Evaluating Learning Algorithms." In: (2011), p. 423. URL: http://books.google.com/books?id=VoWIIOKVzR4C%7B%5C%%7D5Cnhttp://ebooks.cambridge.org/ref/id/CBO9780511921803.

[142] J K Aggarwal and M S Ryoo. "Human Activity Analysis: A Review." In: *ACM Computing Surveys* (2011).

[143] Tracy Westeyn et al. "A naive technique correcting time-series data for recognition applications." In: *Proceedings - International Symposium on Wearable Computers, ISWC* (2009), pp. 159–160.

[144] Thomas Plötz et al. "Automatic synchronization of wearable sensors and video-cameras for ground truth annotation - A practical approach." In: *Proceedings - International Symposium on Wearable Computers, ISWC* (2012), pp. 100–103.

[145] Jessica Lin et al. "A Symbolic Representation of Time Series, with Implications for Streaming Algorithms." In: *8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, (2003), pp. 2–11.

[146] S G Mallat. *A wavelet tour of signal processing [electronic resource] : the Sparse way / Stephane Mallat*. 2009.

[147] Maximilian Christ, Andreas W Kempa-liehr, and Michael Feindt. "Distributed and parallel time series feature extraction for industrial big data applications $." In: *arxiv* (2017). arXiv: `arXiv:1610.07717v3`.

[148] Andreas Zinnen, Ulf Blanke, and Bernt Schiele. "An analysis of sensor-oriented vs. model-based activity recognition." In: *Proceedings - International Symposium on Wearable Computers, ISWC* (2009), pp. 93–100.

[149] Alexandru Niculescu-mizil and Rich Caruana. "Predicting Good Probabilities With Supervised Learning." In: *22nd international conference on Machine learning*. 1999. 2005.

[150] Holger Junker et al. "Gesture spotting with body-worn inertial sensors to detect user activities." In: *Pattern Recognition* 41.6 (2008), pp. 2010–2024.

[151] Oliver Amft and Gerhard Tröster. "Recognition of dietary activity events using on-body sensors." In: *Artificial intelligence in medicine* 42.2 (2008), pp. 121–36.

[152] Eamonn Keogh and Chotirat Ann Ratanamahatana. "Exact indexing of dynamic time warping." In: *Knowledge and information systems* February 2003 (2005), pp. 358–386.

[153] Daniel Roggen and Hristijan Gjoreski. "Unsupervised Online Activity Discovery Using Temporal Behaviour Assumption." In: *International Symposium on Wearable Computers*. 2017, pp. 42–49.

[154] David Heckerman, Dan Geiger, and David M. Chickering. "Learning Bayesian Networks: The Combination of Knowledge and Statistical Data." In: *Machine Learning* 243 (1995), pp. 197–243.

[155] Christophe Andrieu et al. "An Introduction to MCMC for Machine Learning." In: *Machine Learning* (2003), pp. 5–43.

[156] Nils Y Hammerla and Thomas Plötz. "Let's (not) Stick Together: Pairwise Similarity Biases Cross-Validation in Activity Recognition." In: *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing* (2015), pp. 1041–1051.

[157] Andreas Bulling et al. "Eye movement analysis for activity recognition using electrooculography." In: *IEEE transactions on pattern analysis and machine intelligence* 33.4 (2011), pp. 741–53. URL: `http://www.ncbi.nlm.nih.gov/pubmed/20421675`.

[158] Marc Tonsen et al. "InvisibleEye: Mobile Eye Tracking Using Multiple Low-Resolution Cameras and Learning-Based Gaze Estimation." In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 1.3 (2017), pp. 1–21.

[159] Jun Rekimoto. "GestureWrist and GesturePad: unobtrusive wearable interactiondevices." In: *Wearable Computers, 2001. Proceedings. Fifth International Symposium on*. October. 2001.

[160] Jingyuan Cheng, Oliver Amft, and Paul Lukowicz. "Active capacitive sensing: exploring a new wearable sensing modality for activity recognition." In: *UbiComp '08 Proceedings of the 10th international conference on Ubiquitous computing*. 2010.

[161] Georg Ogris, Matthias Kreil, and Paul Lukowicz. "Using FSR based muscule activity monitoring to recognize manipulative arm gestures." In: *International Symposium on Wearable Computers*. 2007, pp. 5–8.

[162] Maria E Niessen, Tim L M Van Kasteren, and Andreas Merentitis. "IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events." In: *IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events*. 2013, pp. 1–3.

[163] Tim R Mullen et al. "Real - time Neuroimaging and Cognitive Monitoring Using Wearable Dry EEG." In: *EEE Transactions on Biomedical Engineering* (2015), pp. 1–17.

[164] Eric Elenko, Lindsay Underwood, and Daphne Zohar. "Defining digital medicine." In: *Nature* 33.5 (2015).

[165] Jess Mcintosh, Asier Marzo, and Mike Fraser. "EchoFlex: Hand Gesture Recognition using Ultrasound Imaging." In: *CHI '17: Proceedings of the SIGCHI conference on Human factors in computing systems*. 2017, pp. 1923–1934.

[166] Thalmic Labs. *Myo*. 2014. URL: http://www.myo.com.

[167] Christoph Amma, Thomas Krings, and Tanja Schultz. "Advancing Muscle-Computer Interfaces with High-Density Electromyography." In: *CHI '15: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2015, pp. 929–938.

[168] Michael T Wolf et al. "Gesture-Based Robot Control with Variable Autonomy from the JPL BioSleeve." In: *IEEE Conference on Robotics and Automation (ICRA)*. 2013.

[169] Michael Beigl and Holger Krull. "The MediaCup: Awareness Technology embedded in an Everyday Object." In: *Handheld and Ubiquitous Computing*. 1999, pp. 1–3.

[170] Christian Floerkemeier, Electronics Engineers, and Electronics Engineers. "RFID tag antenna based sensing: Does your beverage glass need a refill?" In: *IEEE International Conference on RFID*. 2010.

[171] Michael Beigl and Hans Gellersen. "Smart-Its: An Embedded Platform for Smart Objects." In: *Smart Objects Conference (sOc)*. 2003.

[172] Kristof Van Laerhoven et al. "Pin&Play: The Surface as Network Medium." In: *IEEE Communications Magazine* April (2003), pp. 90–95.

[173] Christian Floerkemeier, Matthias Lampe, and Thomas Schoch. "The Smart Box Concept for Ubiquitous Computing Environments." In: *Smart Objects Conference*. 2003.

[174] DJ Patterson, D Fox, and H Kautz. "Fine-grained activity recognition by aggregating abstract object usage." In: *Wearable Computers, 2005. Proceedings. Ninth IEEE International Symposium on*. 2005. URL: http://ieeexplore.ieee.org/xpls/abs%7B%5C_%7Dall.jsp?arnumber=1550785.

[175] Abdallah El Ali et al. "VapeTracker: Tracking Vapor Consumption to Help E-cigarette Users Quit." In: *CHI '16 Proceedings of the SIGCHI Conference on Human Factors in Computing Systemg*. 2016. arXiv: arXiv:1603.09533v1.

[176] *LowIEE - Smart Cigarette Case*. URL: http://www.lowiee.com/.

[177] Robyn Whittaker et al. "Mobile phone-based interventions for smoking cessation." In: *The Cochrane database of systematic reviews* (2012).

[178] Bethany Raiff et al. "Laboratory Validation of Inertial Body Sensors to Detect Cigarette Smoking Arm Movements." In: *Electronics* (2014). URL: http://www.mdpi.com/2079-9292/3/1/87/.

[179] Amin Ahsan Ali et al. "mPuff: Automated Detection of Cigarette Smoking Puffs from Respiration Measurements." In: *IPSN*. 2012, pp. 269–280.

[180] Nazir Saleheen et al. "puffMarker: A Multi-Sensor Approach for Pinpointing the Timing of First Lapse in Smoking Cessation." In: *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing - UbiComp '15*. 2015, pp. 999–1010. URL: http://dl.acm.org/citation.cfm?id=2750858.2806897.

[181] Paulo Lopez-Meyer et al. "Detection of Hand-to-Mouth Gestures Using a RF Operated Proximity Sensor for Monitoring Cigarette Smoking." In: *The open biomedical engineering journal* (2013).

[182] Jinqi Cui et al. "An Audio-based Hierarchical Smoking Behavior Detection System Based on A Smart Neckband Platform." In: *Mobiquitous 2016*. 2016.

[183] Syed Anas Imtiaz, Mingxu Peng, and Esther Rodriguez-villegas. "Monitoring Smoking Behaviour Using a Wearable Acoustic Sensor." In: *IEEE International Conference on Engineering in Medicine and Biology Society (EMBC)*. 2017.

[184] Artem Dementyev. *Detection and analysis of smoking events with wrist-worn sensors*. Tech. rep. MIT, 2013.

[185] Michael C. Fiore, Carlos Roberto Jaèn, and Timothy B. Baker. "Treating Tobacco Use and Dependence." In: *U.S. Dept. of Health and Human Services, Public Health Service, 2008.* (2008). URL: http://www.ncbi.nlm.nih.gov/books/NBK63952/.

[186] L. E. Wagenknecht et al. "Misclassification of smoking status in the CARDIA study: A comparison of self-report with serum cotinine levels." In: *American Journal of Public Health* 82.1 (1992), pp. 33–36.

[187] Sarah Connor Gorber et al. "The accuracy of self-reported smoking: a systematic review of the relationship between self-reported and cotinine-assessed smoking status." In: *Nicotine & tobacco research : official journal of the Society for Research on Nicotine and Tobacco* 11.1 (2009), pp. 12–24. URL: http://www.ncbi.nlm.nih.gov/pubmed/19246437.

[188] Steven E Meredith et al. "A mobile-phone-based breath carbon monoxide meter to detect cigarette smoking." In: *Nicotine & tobacco research : official journal of the Society for Research on Nicotine and Tobacco* 16.6 (2014), pp. 766–73. URL: http://www.ncbi.nlm.nih.gov/pubmed/24470633.

[189] *CReSS Device*. URL: http://borgwaldt.hauni.com/en/instruments/smoking-machines/smoking-topography-devices/cress-pocket.html.

[190] Bedfont Technologies. *Smokerlyzer Produktreihe ®*. 2015.

[191] *QuitKey Device*. URL: https://www.quitkey.com/.

[192] Roy J Adams, Dunn Hall, and Benjamin M Marlin. "Hierarchical Span-Based Conditional Random Fields for Labeling and Segmenting Events in Wearable Sensor Data Streams." In: *Icml* 48 (2016).

[193] Aurélien Tabard, Evvelyn Eastmond, and Wendy E Mackay. "From Individual to Collaborative: The Evolution of Prism, a Hybrid Laboratory Notebook." In: *Computer Supported Cooperative Work*. ACM, 2008.

[194] Colin L Bird, Cerys Willoughby, and Jeremy G Frey. "Laboratory notebooks in the digital era: the role of ELNs in record keeping for chemistry and other sciences." In: *Chemical Society reviews* 42 (2013), pp. 8157–75. URL: http://www.ncbi.nlm.nih.gov/pubmed/23864106.

[195] Francois Roubert and Mark Perry. "Putting the Lab in the Lab Book : Supporting Coordination in Large , Multi-site Research." In: *Proceedings of the 27th International BCS Human Computer Interaction Conference* (2002), pp. 1–10.

[196] Jim Giles. "Going paperless: The digital lab." In: *Nature* 481.7382 (2012), pp. 430–431. URL: http://www.nature.com/doifinder/10.1038/481430a.

[197] Clemens Nylandsted Klokmose and Pär-Ola Zander. "Rethinking Laboratory Notebooks." In: (2010). Ed. by Myriam Lewkowicz et al., pp. 119–140. URL: http://link.springer.com/10.1007/978-1-84996-211-7.

[198] Stacey Kuznetsov et al. "Open Source Biology Tools as Platforms for Hybrid Knowledge Production and Scientific Participation." In: *CHI '15 Proceedings of the SIGCHI Conference on Human Factors in Computing Systemg*. 2015.

[199] James D Myers. "Collaborative Electronic Notebooks as Electronic Records : Design Issues for the Secure Electronic Laboratory Notebook (ELN)." In: *Simulation Series* (2003).

[200] Lea A I Vaas et al. "Electronic laboratory notebooks in a public–private partnership." In: *PeerJ Computing* (2016), pp. 1–22.

[201] Gerard Oleksik. "Study of an Electronic Lab Notebook Design and Practices that Emerged in a Collaborative Scientific Environment." In: *17th ACM conference on Computer supported cooperative work & social computing*. 2014, pp. 120–133.

[202] Samantha Kanza et al. "Electronic lab notebooks: can they replace paper?" In: *Journal of Cheminformatics* 9 (2017), pp. 1–15.

[203] Michael Bernstein et al. "Information Scraps: How and Why Information Eludes our Personal Information Management Tools." In: *ACM Transactions on Information Systems* 26.4 (2008), p. 24.

[204] Wendy E Mackay, Guillaume Pothier, and Catherine Letondal. "The Missing Link: Augmenting Biology Laboratory Notebooks." In: *Symposium on User Interface Software and Technology*. Vol. 15. 2. 2002.

[205] Jim Gemmell et al. "MyLifeBits: Fulfilling the Memex Vision." In: *International Conference on Multimedia*. ACM, 2002.

[206] Larry Arnstein et al. "Labscape: a smart environment for the cell biology laboratory." In: *IEEE Pervasive Computing* 1.3 (2002), pp. 13–21. URL: `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1037717`.

[207] Gaetano Borriello. "Invisible computing: automatically using the many bits of data we create." In: *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences* 366.1881 (2008), pp. 3669–3683.

[208] Gareth Hughes et al. "The semantic smart laboratory: a system for supporting the chemical eScientist." In: *Organic & Biomolecular Chemistry* (2004). URL: `http://www.ncbi.nlm.nih.gov/pubmed/15534706`.

[209] Simon J Coles et al. "First steps towards semantic descriptions of electronic laboratory notebook records." In: *Journal of cheminformatics* 5 (2013), p. 52.

[210] Aurélien Tabard et al. "The eLabBench in the wild: supporting exploration in a molecular biology lab." In: *Human Factors in Computing Systems*. ACM, 2012. URL: http://dl.acm.org/citation.cfm?id=2208718.

[211] Florian Echtler et al. "BioTISCH: the interactive molecular biology lab bench." In: *CHI Extended Abstracts on Human Factors in Computing Systems*. ACM, 10, pp. 5–10.

[212] Tom Nicolai, Thomas Sindt, and Hendrik Witt. "Wearable computing for aircraft maintenance: Simplifying the user interface." In: *Applied Wearable Computing (IFAWC)*. 2006. URL: http://ieeexplore.ieee.org/xpls/abs%7B%5C_%7Dall.jsp?arnumber=5758272.

[213] J.J. Ockerman and a.R. Pritchett. "Preliminary investigation of wearable computers for task guidance in aircraft inspection." In: *International Symposium on Wearable Computers (ISWC)* (1998), pp. 33–40. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=729527.

[214] Grace Hu et al. "Exploring the Use of Google Glass in Wet Laboratories." In: *Extended Abstracts of the ACM CHI'15 Conference on Human Factors in Computing Systems* 2 (2015), pp. 2103–2108. URL: http://dx.doi.org/10.1145/2702613.2732794.

[215] Walter Bell et al. "Best Practices for Repositories I: Collection, Storage, and Retrieval of Human Biological Materials for Research." In: *Cell Preservation Technology* 3.1 (2005), pp. 5–48.

[216] Jerry J Lou et al. "A review of radio frequency identification technology for the anatomic pathology or biorepository laboratory: Much promise, some progress, and more work needed." In: *Journal of pathology informatics* 2 (2011), p. 34.

[217] Amitava Dasgupta and Jorge Sepulveda. *Accurate Results in the Clinical Laboratory - A Guide to Error Detection and Correction*. 2013.

[218] Frank R. Ihmig et al. "RFID for anonymous biological samples and pseudonyms." In: *2011 IEEE International Conference on RFID-Technologies and Applications, RFID-TA 2011* (2011), pp. 376–380.

[219] Atieh Zarabzadeh, R. William G Watson, and Jane Grimson. "The use of radio frequency identification to track samples in bio- Repositories." In: *Proceedings of the 2008 1st International Conference on Information Technology, IT 2008* May (2008).

[220] Hanan Davidowitz. "Use of Radio Frequency Identification (RFID) for Sample Tracking." In: *American Laboratory* (2012).

[221] By Joshua R Smith et al. "RFID-Based Technologies for Human-Activity Detection." In: *Communications of the ACM* 48.9 (2005), pp. 39–44.

[222] Kenneth P. Fishkin, Matthai Philipose, and Adam Rea. "Hands-on RFID: Wireless wearables for detecting use of objects." In: *Proceedings - International Symposium on Wearable Computers, ISWC* 2005 (2005), pp. 38–41.

[223] Eugen Berlin et al. "Coming to Grips with the Objects We Grasp: Detecting Interactions with Efficient Wrist-Worn Sensors." In: *Tangible, Embedded, and Embodied Interaction*. 2010.

[224] Michael Buettner et al. "Recognizing Daily Activities with RFID-Based Sensors." In: *Ubicomp 2009* (2009), pp. 51–60.

[225] Gaetano Borriello. "The inivisble assistant." In: *ACM Queue* 4.6 (2006), p. 44.

[226] Eric Freeman and Scoott Fertig. "Lifestreams: Organizing your Electronic." In: *Conference companion on Human factors in computing systems*. 1996.

[227] Scott Deerwester et al. "Indexing by Latent Semantic Analysis." In: *Journal of the American Society for Information Science* (1990).

[228] C.A.R. Hoare. "A Model for Communicating Sequential Processes." In: *The origin of concurrent programming*. 1978.

[229] David Bannach et al. "Integrated tool chain for recording and handling large, multimodal context recognition data sets." In: *Proceedings of the 12th ACM international conference adjunct papers on Ubiquitous computing - Ubicomp '10* (2010), p. 357. URL: `http://doi.acm.org/10.1145/1864431.1864434`.

[230] G. Spina et al. "CRNTC+: A smartphone-based sensor processing framework for prototyping personal healthcare applications." In: *Pervasive Computing Technologies for Healthcare (PervasiveHealth)*. May. 2013.

[231] Kent Lyons, Helene Brashear, and Tracy Westeyn. "GART : The Gesture and Activity Recognition Toolkit." In: *Electronics* 4552 (2007), pp. 718–727. URL: `http://portal.acm.org/citation.cfm?id=1769671`.

[232] Non-Profit Organization Matroska. *The Matroska File Format*. 2016. URL: https://www.matroska.org/.

[233] Nobuo Kawaguchi, Nobuhiro Ogawa, and Yohei Iwasaki. "Hasc challenge: gathering large scale human activity corpus for the real-world activity understandings." In: *Proceedings of the 2nd Augmented Human International Conference* (2011), p. 27. URL: http://dl.acm.org/citation.cfm?id=1959853.

[234] Stephen S Intille et al. "Using a Live-In Laboratory for Ubiquitous Computing Research." In: *Pervasive Computing* (2006), pp. 349–365.

[235] T L M Van Kasteren, G Englebienne, and B J A Kr. "Human Activity Recognition from Wireless Sensor Network Data : Benchmark and Software." In: *Activity Recognition in Pervasive Intelligent Environments*. 2010.

[236] Marko Borazio et al. "Towards Benchmarked Sleep Detection with Inertial Wrist-worn Sensing Units." In: *Healthcare Informatics (ICHI)*. 2014.

[237] Mark Hall et al. "The WEKA data mining software." In: *SIGKDD Explorations Newsletter* 11.1 (2009), p. 10. URL: http://portal.acm.org/citation.cfm?doid=1656274.1656278.

[238] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: A Library for Support Vector Machines*. 2011.

[239] Thomas Stiefmeier et al. "Wearable activity tracking in car manufacturing." In: *IEEE Pervasive Computing* 7.2 (2008), pp. 42–50.

[240] Kristof van Laerhoven. *Longitudinal Wrist Motion Dataset*. URL: https://es.informatik.uni-freiburg.de/application/files/hhg%7B%5C_%7Dlogs/0089/index.html (visited on 06/06/2016).

[241] Foundation Xiph.org. *The Free Lossless Audio Codec (FLAC)*. URL: https://www.xiph.org/flac/.

[242] David Bryant. *The WavPack Codec*. URL: https://www.wavpack.org/.

[243] David Lamparter. *Advanced Sub Station Alpha*. URL: http://fileformats.wikia.com/wiki/SubStation%7B%5C_%7DAlpha (visited on 06/06/2016).

[244] Consortium WebM. *The WebM File Format*. URL: http://www.webmproject.org/ (visited on 06/06/2016).

[245] Suramya Tomar. "Converting video formats with FFmpeg." In: *Linux Journal 2006* (2006).

[246] Google Inc. *Google Developer Guides to Dataset Curaton*. 2017. URL: `https://developers.google.com/search/docs/data-types/datasets`.

[247] Marc Kurz et al. "The OPPORTUNITY Framework and Data Processing Ecosystem for Opportunistic Activity and Context Recognition." In: *International Journal of Sensors Wireless Communications and Control* 1.2 (2012), pp. 102–125.

[248] Kai Kunze and Paul Lukowicz. "Sensor placement variations in wearable activity recognition." In: *IEEE Pervasive Computing* 13.4 (2014), pp. 32–41.

[249] Ny Hammerla and Reuben Kirkham. "On Preserving Statistical Characteristics of Accelerometry Data using their Empirical Cumulative Distribution." In: *... International Symposium on ...* (2013), pp. 65–68. URL: `http://dl.acm.org/citation.cfm?id=2494353`.

[250] Melanie Hartmann, Alexander Bauer, and Ulf Blanke. *Method and system for sensor classification*. US Patent 7062320. 2013.

[251] De la Torre et.al. "Guide to the Carnegie Mellon University Multimodal Activity (cmu-mmac) Database." In: *Robotics Institute*. 2008.

[252] Katerina et.al. Karagiannaki. "A benchmark study on feature selection for human activity recognition." In: *ACM Ubicomp: Adjunct*. ACM. 2016.

[253] Attila Reiss, Gustaf Hendeby, and Didier Stricker. "Confidence-based multiclass AdaBoost for physical activity monitoring." In: *International Symposium on Wearable Computers*. ACM. 2013.

[254] Daniel Roggen et al. "Collecting complex activity datasets in highly rich networked sensor environments." In: *International Conference on Networked Sensing Systems*. 2010.

[255] Oresti et.al. Banos. "mHealthDroid: a novel framework for agile development of mobile health applications." In: *International Workshop on Ambient Assisted Living*. Springer. 2014.

[256] Jamie A Ward et al. *Continuous activity recognition in the kitchen using miniaturised sensor button*. 2002.

[257] Eric Guenterberg et al. "An Automatic Segmentation Technique in Body Sensor Networks based on Signal Energy." In: *Proceedings of the Fourth International Conference on Body Area Networks*. 2009, pp. 1–7.

[258] Daniel Lemire. "A Better Alternative to Piecewise Linear Time Series Segmentation." In: *ARXIV* (2007). arXiv: 0605103 [cs]. URL: http://arxiv.org/abs/cs/0605103.

[259] Samaneh Aminikhanghahi and Diane J. Cook. "Using Change Point Detection to Automate Daily Activity Segmentation." In: *Workshop on Context and Activity Modeling and Recognition Using*. 2017.

[260] Nils Yannick Hammerla. "Activity recognition in naturalistic environments using body-worn sensors." PhD thesis. 2015.

[261] Rehan Akbani, Stephen Kwek, and Nathalie Japkowicz. "Applying Support Vector Machines to Imbalanced Datasets." In: *Lnai* 3201 (2004), pp. 39–50.

[262] Laurens Van Der Maaten and Geoffrey Hinton. "Visualizing Data using t-SNE." In: *Journal of Machine Learning Research* 9 (2008), pp. 2579–2605.

[263] Christophe Viau et al. "The FlowVizMenu and parallel scatterplot matrix: Hybrid multidimensional visualizations for network exploration." In: *IEEE Transactions on Visualization and Computer Graphics* 16.6 (2010), pp. 1100–1108.

[264] Ron Bekkerman, Mikhail Bilenko, and John Langford. *Scaling up Machine Learning: Parallel and Distributed Approaches*. 2012.

[265] Ole Tange. "GNU Parallel, the command-line power tool." In: *The USENIX Magazine* 36.1 (2011).

[266] Philipp Marcel Scholl and Kristof Van Laerhoven. "A Multi-Media Exchange Format for Time-Series Dataset Curation." In: *UbiComp 2016 Adjunct - Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. 2016.

[267] Philipp Marcel Scholl and Kristof van Laerhoven. "On the Statistical Properties of Body-Worn Inertial Motion Sensor Data for Identifying Sensor Modality." In: *International Symposium on Wearable Computers*. 2017.

[268] World Health Organization. "Global status report on noncommunicable diseases 2010." In: *World Health* (2010), p. 176. URL: http://whqlibdoc.who.int/publications/2011/9789240686458%7B%5C_%7Deng.pdf.

[269] Marie Ng et al. "Smoking prevalence and cigarette consumption in 187 countries, 1980-2012." In: *Jama, American Medical Association* 311 (2014).

[270] European Commission. *Attitudes of Europeans towards tobacco and electronic cigarettes*. Vol. 429. May. 2015, p. 214.

[271] Public Health Service. "The health consequences of smoking—50 years of progress: A report of the Surgeon General." In: *Smoking and Health* (2014). URL: http://ash.org/wp-content/uploads/2014/01/full-report.pdf%7B%5C%%7D5Cnhttp://www.legacyforhealth.org/content/download/4428/62627/file/Abrams.SurGenReport.50thAnniv.2.5.14.FIN.pdf.

[272] Andrew Jarvis et al. *A Study On Liability And The Health Costs Of Smoking*. 2009.

[273] Cancer Facts and American Cancer Society. "Cancer Facts & Figures." In: (2012).

[274] Ian Li, Anind K Dey, and Jodi Forlizzi. "Understanding My Data, Myself: Supporting Self-Reflection with Ubicomp Technologies." In: *Discovery* (2011).

[275] Stefanie De Jesus et al. "A systematic review and analysis of data reduction techniques for the CReSS smoking topography device." In: *Journal of Smoking Cessation* 10.1 (2015), pp. 12–28.

[276] Philipp Marcel Scholl, Nagihan Kücükyildiz, and Kristof Van Laerhoven. "Bridging the Last Gap: LedTX - Optical Data Transmission of Sensor Data for Web-Services." In: *Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication*. 2013.

[277] Zentri Limited. *Data sheet - AMSoox BLE Module*. 2015.

[278] Joseph a Paradiso and Mark Feldmeier. "A Compact , Wireless , Self-Powered Pushbutton Controller." In: *Ubicomp 2001: Ubiquitous Computing*, (2001), pp. 299–304.

[279] Alexander Rudmann. "Evaluuierung verschiedener Energiequellen für ein intelligentes Feuerzeug." Bachelor Thesis. University of Freiburg, 2016.

[280] PicoTech. *PicoScope 3206 MSO*. URL: https://www.picotech.com/products.

[281] David Jones. *The uCurrent - a professional precision current adapter for Multimeters*. 2010. URL: http://www.eevblog.com/files/uCurrentArticle.pdf.

[282] Philipp Marcel Scholl et al. "When Do You Light a Fire? Capturing Tobacco Use with Situated, Wearable Sensors." In: *Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication*. 2013.

[283] Kai Kunze. "Compensating for On-Body Placement Effects in Activity Recognition." In: June (2011), pp. 1–61.

[284] Stephen S Intille et al. "A Living Laboratory for the Design and Evaluation of Ubiquitous Computing Technologies." In: *CHI'05 extended abstracts on Human factors in computing systems*. 2005, pp. 1–4.

[285] Sarah Connor Gorber et al. "The accuracy of self-reported smoking: A systematic review of the relationship between self-reported and cotinine-assessed smoking status." In: *Nicotine and Tobacco Research* 11.1 (2009), pp. 12–24.

[286] C a Patten and J E Martin. "Measuring tobacco withdrawal: a review of self-report questionnaires." In: *Journal of substance abuse* 8.1 (1996), pp. 93–113. URL: http://www.ncbi.nlm.nih.gov/pubmed/8743771.

[287] Stéphanie a Prince et al. "A comparison of direct versus self-report measures for assessing physical activity in adults: a systematic review." In: *The international journal of behavioral nutrition and physical activity* 5 (2008), p. 56.

[288] Daniel Roggen et al. "Gestures are Strings: Efficient online gesture spotting and classification using string matching." In: *International Conference on Body Area Networks (BodyNets)*. May 2014. 2007.

[289] J J Arnett. "Optimistic bias in adolescent and adult smokers and nonsmokers." In: *Addictive Behaviors* 25.4 (2000), pp. 625–632.

[290] John Paul Varkey, Dario Pompili, and Theodore a. Walls. "Human motion recognition using a wireless sensor-based wearable system." In: *Personal and Ubiquitous Computing* 16.7 (2011), pp. 897–910. URL: http://link.springer.com/10.1007/s00779-011-0455-4.

[291] Philipp M Scholl and Kristof Van Laerhoven. "A Feasibility Study of Wrist-Worn Accelerometer Based Detection of Smoking Habits." In: *2012 Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*. 2012.

[292] Takumi Kobayashi, Koiti Hasida, and Nobuyuki Otsu. "Rotation Invariant Feature Extraction From 3-D Acceleration Signals." In: *Acoustics, Speech and Signal Processing (ICASSP)*. 2011, pp. 3684–3687.

[293] *Gluco Wise Device*. URL: `http://www.gluco-wise.com/`.

[294] Bosch Sensortec. *BMX160 9-axis Absolute Orientation Sensor Datasheet*. Tech. rep. 2017.

[295] TDK InvenS. *ICM-20948 Datasheet*. Tech. rep. 2017, pp. 1–89.

[296] Nordic Semiconductor. *NRF51822 Product Specification v3.3*. Tech. rep. 2014.

[297] Philipp Marcel Scholl and Kristof van Laerhoven. "Lessons Learned From Designing an Instrumented Lighter for Assessing Smoking Status." In: *ACM International Joint Conference on Pervasive and Ubiquitous Computing*. S, 2017.

[298] ProGlove GmBH. *ProGlove*. URL: `https://www.proglove.de`.

[299] PanMobil GmBH. *PanMobil*. URL: `https://www.panmobil.de`.

[300] John Gruber. *Markdown: Syntax*. URL: `http://daringfireball.net/projects/markdown/syntax`.

[301] Philipp Marcel Scholl and Kristof Van Laerhoven. "Wearable digitization of life science experiments." In: *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing Adjunct Publication - UbiComp '14 Adjunct* (2014).

[302] Philipp Marcel Scholl, Matthias Wille, and Kristof Van Laerhoven. "Wearables in the Wet Lab: A Laboratory System for Capturing and Guiding Experiments." In: *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. 2015.

[303] Philipp Marcel Scholl and Kristof Van Laerhoven. "RFID-Based Compound Identification in Wet Laboratories With Google Glass." In: *WOAR '15 Proceedings of the 2nd international Workshop on Sensor-based Activity Recognition and Interaction*. 2015.

[304] Philipp Marcel Scholl and Kristof Van Laerhoven. "Remind - Towards a Personal Remembrance Search Engine for Motion Augmented Multi-Media Recordings." In: *MUM 2016 - Proceedings of the international 2016 Conference on Multimedia and Ubiquitous Systems*. 2016.