Textile Pressure Mapping (TPM) for Pervasive and Wearable Activity Recognition: Sensing, Framework and Applications

Thesis approved by
the Department of Computer Science
Technische Universität Kaiserslautern
for the award of the Doctoral Degree
Doctor of Engineering (Dr.-Ing.)

to

Bo Zhou

Date of Defense: 29th Oct 2019
Dean: Prof. Dr. Stefan Deßloch
Reviewer: Prof. Dr. Paul Lukowicz
Reviewer: Prof. Dr. Albrecht Schmidt

D 386

SUMMARY

Planar force or pressure is a fundamental physical aspect during any people-vs-people and people-vs-environment activities and interactions. It is as significant as the more established linear and angular acceleration (usually acquired by inertial measurement units). There have been several studies involving planar pressure in the discipline of activity recognition, as reviewed in the first chapter. These studies have shown that planar pressure is a promising sensing modality for activity recognition. However, they still take a niche part in the entire discipline, using ad hoc systems and data analysis methods. Mostly these studies were not followed by further elaborative works. The situation calls for a general framework that can help push planar pressure sensing into the mainstream.

This dissertation systematically investigates using planar pressure distribution sensing technology for ubiquitous and wearable activity recognition purposes. We propose a generic Textile Pressure Mapping (TPM) Framework, which encapsulates (1) design knowledge and guidelines, (2) a multi-layered tool including hardware, software and algorithms, and (3) an ensemble of empirical study examples. Through validation with various empirical studies, the unified TPM framework covers the full scope of application recognition, including the ambient, object, and wearable subspaces.

The hardware part constructs a general architecture and implementations in the large-scale and mobile directions separately. The software toolkit consists of four heterogeneous tiers: driver, data processing, machine learning, visualization/feedback. The algorithm chapter describes generic data processing techniques and a unified TPM feature set. The TPM framework offers a universal solution for other researchers and developers to evaluate TPM sensing modality in their application scenarios.

The significant findings from the empirical studies have shown that TPM is a versatile sensing modality. Specifically, in the ambient subspace, a sports mat or carpet with TPM sensors embedded underneath can distinguish different sports activities or different people's gait based on the dynamic change of body-print; a pressure sensitive tablecloth can detect various dining actions by the force propagated from the cutlery through the plates to the tabletop. In the object subspace, swirl office chairs with TPM sensors under the cover can be used to detect the seater's real-time posture; TPM can be used to detect emotion-related touch interactions for smart objects, toys or robots. In the wearable subspace, TPM sensors can be used to perform pressure-based mechanomyography to detect muscle and body movement; it can also be tailored to cover the surface of a soccer shoe to distinguish different kicking angles and intensities.

All the empirical evaluations have resulted in accuracies well-above the chance level of the corresponding number of classes, e.g., the 'swirl chair' study has classification accuracy of 79.5% out of 10 posture classes and in the 'soccer shoe' study the accuracy is 98.8% among 17 combinations of angle and intensity.

My sincere gratitudes to:

Prof. Paul Lukowicz, for giving me the opportunity, resources and guidance for this work. There are so much I have learnt from Paul: how to analytically and efficiently approach a problem, how to write strong arguments and how to deliver on point speeches.

Prof. Jingyuan Cheng, for introducing me to the topic, and her tireless spirit and attention to details have been a great inspiration to me.

Prof. Albrecht Schmidt, for giving me valuable opinions on the thesis writing and for constantly motivating me to finish my dissertation.

Mathias Sundholm, for working together on the Smart-Mat and many other projects during the first year.

Heber Zurian Cruz, for helping me with implementing the FTDI protocol and other demos.

Monit Shah Singh, for implementing the transfer learning with deep neural network part.

And all other co-authors in my publications, especially but not only:
Prof. Stefan Schneegaß, Prof. Fernando Seoane, Prof. Björn Eskofier, Prof. Oliver Amft, Attila Reiss, Markus Wirth, Christine Martindale

I would also like to thank all the students and research assistants who have helped me:
Muhammet Yildrim, Marius Laval, Ivi Prifti, Alejandro Baucells Costa, Carlos Velez, Yorman Munoz, Jakub Jakubowski, Jyothish James.

As well as all my colleagues that created this inspiring working environment:
Jane Bensch, Agnes Grünerbl, Hamraz Javaheri, Hymalai Bello, Orkhan Amiraslanov, Gernot Bahle, Tobias Franke, Jens Weppner, Peter Hevesi, Marco Hirsch, Vitor Fortes Rey, George Kampis, Sizhen Bian, Gerald Pirkl, Junaid Younas

Also thanks to other friends who have given me interesting ideas on my dissertation:
Prof. Tandra Ghose, Robert Gressnich, Miguel Sousa.

At last, special thanks to my family, friends, and Snoopy. Without their support, I would not have come this far.

Save the Planet with a Smarter Template

One thing I have learned from working in a research institute, is that to make new discoveries and exceed the boundary of our knowledge, we need to question and challenge existing knowledge, rules and habits that people take for granted. It is common to gauge the progress of a thesis by the page count. When I was writing my bachelor or master thesis, at some point I was hoping to have more pages, and it gave me the feeling that with more pages done, I was getting closer to the finishing line. But now I have come to realize that feeling was nothing but an illusion, a thesis is done when the work is done, when the author and the supervisors all agree it is done. A doctorate dissertation (or any academic thesis in general) should not be valued by its superficial page count or the thickness of the printed book - it is merely an irrelevant feature, not a determinant variable for the content quality. Nobody knows how many pages is enough, faculties also do not have a rule on the page count. More pages means more trees cut down, more carbon footprint, and more paper to flip while reading the dissertation, and nothing more. I would not want my dissertation that I have put so much heart and effort on to be judged by this criteria. Thus I've designed this template based on the double-column IEEEtran scientific paper format with the majority fontsize of $10pt$. Compared to the popular single-column thesis template with bigger fontsize and row spacing, on the one hand, it is a format that everyone in this discipline is very familiar with, and I personally find it easier and faster to read; on the other, it roughly cuts the page count to half. Small figures, tables, equations and bullet points take much less space. If the thesis is printed double sidedly, as the template is intended, it would take just about one quarter of the pages with traditional template. To make it easier for the readers to go through this dissertation, every chapter starts with a single-column summary and introduction page, the remaining texts are then of the double-column format.

Table of Contents

List of Figures

List of Tables

Chapter 1 Introduction

The discipline of pervasive and wearable computing, combined with artificial intelligence algorithms, have made great contributions to the improvement of life quality, or simply changing our lifestyles in the past years, bringing automation, feedback, novel human-computer interactions and statistical analysis into our everyday activities. With multiple embedded sensors in people's digital devices, not only the software can self-adjust to the users' activity and use it as non-invasive input methods, but people can also use life-logging and activity tracking to improve their lifestyle. Most recently, smart home has also become a popular topic, not just among researchers, but also global leading technology companies, e.g., smart lighting systems can adjust different tones to the occupant's activity and mood, or automated by schedules or sensors. This chapter introduces the background and applicable area of this dissertation. Every main chapter will start with such a cover page to briefly introduce and summarize its content.

Activity recognition and pattern discovery is an important branch of the artificial intelligence discipline [1]. The general goal of activity recognition is to provide the user personalized and context-customized support, by information feedback or environmental changes that are unique to different activities of the user. Fig. 1 shows several possibilities of how activity recognition can help not only users with their daily activities but also the experts to understand the users' activities through different aspects and discover patterns from a large collection of data.

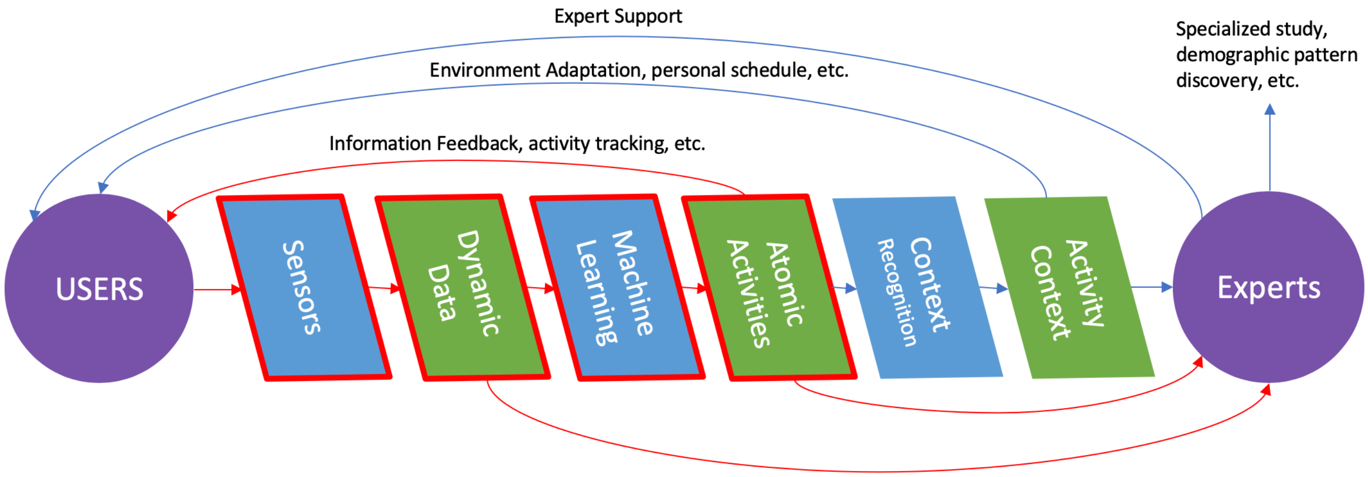

Expert Support
Environment Adaptation, personal schedule, etc.
Information Feedback, activity tracking, etc.
Specialized study, demographic pattern discovery, etc.
USERS
Sensors
Dynamic Data
Machine Learning
Atomic Activities
Context Recognition
Activity Context
Experts

Fig. 1. Main blocks of activity recognition. Red color outlines the scope of this dissertation.

Currently, motion-based sensors (inertial measurement units, IMUs) are widely available in smartphones, smartwatches, and smartglasses; therefore a lot of activity recognition research is based on IMUs, and IMUs have become a mature sensing method. [2] IMUs typically measure the movement of the users' body or the objects which they are using. [3]

In this thesis, instead of the motion, force is evaluated for detecting activities, since usually force is the cause or the consequence of motions. More specifically, this dissertation builds a framework for textile pressure mapping (TPM) sensing technology, and uses it to look at the dynamic change and distribution of planar surface pressure force in various applications.

The work of this dissertation within the activity recognition structure is marked with red in Fig. 1. It starts from the *Sensors* block, with building a hardware architecture for TPM sensing technology. Then *Machine Learning* algorithms that specifically for the pressure mapping data to extract information that represent the users' activities. The hardware, software and the related algorithm parts of the framework is evaluated through various empirical studies. These studies, in turn, inspire the improvement and evolution of the framework.

The structure of this chapter:

- First, the background of this work, pervasive and wearable activity recognition is introduced in more details;
- Then the motivation of this work and where it lies within the activity recognition background is explained;
- A review of the state of the art follows, especially in the scope of planar pressure force mapping technologies;
- Finally, with a comparison of this work and the state of the art, the chapter concludes with the contribution of this work.

1.1 Background

1.1.1 Ubiquitous Sensing and Computing

As computing and sensing devices grow smaller and more affordable during the past two decades, visions of how to utilize them in daily environments of people's normal life have emerged. [4] [5]

Some direct and simple examples of how ubiquitous sensing is helping our daily life are sound activated light or proximity sensitive automated doors. In these systems, simple sensors act as a switch based on certain physical variables such as the sound volume or proximity to control other electronics and automated devices. Fall detection for elderly and patients [6] is another more advanced ubiquitous sensing and computing applications that deal with more sophisticated sensor signal processing and data mining techniques to discover the patterns of certain events. Regardless of the complexity, those ubiquitous sensing and computing applications share a similar core structure as from Fig. 1: sensors pick up signals that are related to the people's activity and the physical world, the digitized signals are processed by a computing unit to determine the activity, then support the people by trigger certain changes in the environment or notifying related experts. [7]

Ubiquitous sensors are the first step of these applications. [7] By the sensing principle, sensors can be categorized as passive and active sensors. Passive sensors' outputs are caused directly by the physical variable they are sensitive to, such as inertial measurement units (IMUs) or optical sensors; active sensors are transceivers, they emit stimuli into the world and measure the change of the signal that is reflected back to its receivers, such as proximity sensors, ultrasonic systems or 3D cameras. There are also other types of devices that are widely used in ubiquitous sensing and computing that do not belong to either passive or active sensors, such as Bluetooth beacons and RFID tags that only functions when a user carries certain devices (such as smartphones) in the proximity.

In general, there are three categories in terms of the sensor installation **spaces**:

- **Ambient**: Sensors are placed in the surrounding environment where the targeted activity happens. Active sensors, such as cameras [8] and Kinect [9], can directly track the occupants movement; passive sensors, such as indoor seismic sensor [10], ambient atmosphere sensor [11], may also be used for monitoring the changes of the environment itself caused by occupants' activities. Beacons and RFID tags can be installed in the environment to assist indoor localization and occupant tracking.
- **Object**: Objects that people interact with can be equipped with sensors to track the objects' motion or the interaction between the people and the object such as touch or force. The separation principle of the *object* and *ambient* in this work is defined that *objects* have more mobility and typically smaller scale. For example, a smart couch would be categorized as *ambient* while an office swivel chair is *object*.
- **Wearable**: Miniaturized sensors can be directly worn on-body, and move with the attached body parts. IMUs have been maturely developed [2] and most current consumer devices (smartphones, smartwatches) are equipped with them. Head-mounted cameras and eye trackers are also trending sensing methods. [12]

While in some studies these spaces or domains are not clearly separated, for example, in [13] or [14], non-wearable sensors are all categorized as ambient sensors. The space categorization is defined mainly from the sensor platform design point of view, which are in the later chapters of this work. The *object* space is necessary to represent a specific hardware implementation between *wearable* and *ambient*.

1.1.2 Activity Recognition

The basic principle driving the research of human activity recognition is finding the correlation between certain activities and the data from the sensing systems. The activities can be of different scenarios such as elderly assist, home automation, sports performance and interactions, infrastructure management, security, etc. The sensing systems can consist of either a single type of sensors or a combination of different sensors in different installation categories. A combination of sensors are typically interesting since different installations and sensing modalities can compensate each others' weak points or blind spots and the challenge of combining their data in the machine learning algorithms.

The activities can have meaning on two different levels: First is the **atomic activity**, (also known as the transient activity or the activity state), which is typically the direct classification result of a temporal window of the sensor data. The temporal sequences of sensor data are processed to extract features that can uniquely represent the data, and then using supervised machine learning, classifiers can be trained to recognize different activities. [15] They may represent simple actions such as raising a hand, running or sitting down. The higher level is **activity context**, which is a combination of atomic activities. [1] For example, if a person is (1) in the kitchen (from location sensing) and (2) moving arms (from wearable motion sensing) and (3) making sharp metal noise (from ambient microphone), they may be cooking. [16] Such scenarios combine multiple sensor inputs, sensor fusion techniques are therefore proposed, either dealing with multiple sensors from the same domain [17] or cross-domain sensors that both are in the surroundings and on the users [14].

1.1.3 Sensors

In Section 1.1.1, ubiquitous sensors are categorized into three different spaces from the system design point of view. Yet the same type of sensors can be installed in different scenarios.

IMUs, the combination of accelerometers, gyroscopes, and magnetometers, are typically the primary choice for pervasive activity recognition, because they directly measure the physical movement. Yang *et al.* [18] has summarized the capabilities of using IMUs as wearable activity recognition sensor, including posture and movement classification, energy expenditure estimation and balance control evaluation. Embedded IMU sensors are available in very small integrated circuits (IC) packages such as the Bosch BMI055 or InvenSense 9250, which are both a mere millimeter in size. To utilize the sensors, embedded devices are needed to support operating the IMU ICs, such as the ready-to-use solutions offered by Xsens. As most recently, the emerging of the smartphone and smartwatch market, with integrated IMUs is also helping the study of IMU-based activity recognition, since they can be easily programmed and contains plenty of connectivity and output interfaces. [19] [20]

In the computer vision discipline, human motion analysis is a classic central topic. [21] offers a comprehensive overview of different methods: Vision-based motion analysis can either identify body parts or track the human as a cluster of pixels; some methods also involve building a joint-connected volumetric 3D model of the human body and fit the model on the 2D images. State-space approaches [22] [23] and template matching are both used to recognize human activities. In recent years, the development of depth cameras (or RGB-D cameras) such as the Microsoft Kinect, has also broadened the potential of computer vision based applications.

Object-based binary sensors such as Radio-Frequency Identification (RFID) tags are a low-cost solution when detecting activity based on localized objects usages. Such methods typically focus on the activity context and rather than atomic activities, for example: shaking a coffee container and opening it can be the same from the sensor's output. In [24], users wear an RFID reader glove and 60 household objects are tagged with RFID chips such as the coffee container or toilet lid. Using an aggregate model, 11 activity contexts such as 'using the bathroom' or 'making oatmeal' can be recognized by up to 81.2% accuracy.

Microphones are another popular sensor, thanks to their small footprint and established signal processing techniques such as spectrum analysis. More importantly, sound can be picked up even from a distance and typically won't be obstructed compared to computer vision based methods. In [25], microphone and IMUs are worn on the hand while participants are doing carpenter workshop activities, two sensing modalities are compared and also combined together. From 21 activities, two sensing modalities have on-par results for user-dependent validations, and the microphone has superior accuracy than IMUs for user-independent cases. In [26], the sound picked up from the neck can be used to classify 12 activities including laughing, eating, etc.

1.1.4 Issues in Selecting the Right Sensors

Examining the sensing modalities introduced above, there are several common issues when selecting sensors for activity recognition:

Sensitivity: some sensors such as IMUs, microphones, and capacitive sensors are very sensitive to even small atomic activities, while some sensors such as RFID only has binary sensitivity. High sensitivity, however, usually also makes the sensor prone to noise (e.g. for IMUs, random movements that are not interested for the target activities; or for microphones environment noise or sound from other users that are not relevant). For cameras, extreme lighting condition changes is also a significant problem. While TPM can be placed at the locations where the activity happens, thus it generally suffers less motion artifact.

Location dependency: some wearable sensors are typically fit at specific locations. In [27], Kunze, et al. have pointed out the problem that sensor placement variations will greatly influence the activity recognition result while investigating microphones and IMUs. Typical lab experiments the sensors are fit on fixed locations, but in the real world the sensors may shift over time and displaced completely based on the users (e.g. putting a smart glass off from the head and into the pocket). As a flexible textile matrix, it can cover more area easily than rigid sensors; and thus TPM sensors are less influenced by sensor location shifting.

User acceptance: comfort, consent of information and privacy are determinant factors for a sensing modality to be accepted in people's daily activity. TPM can detect activities from the surrounding instead of directly on the users, which makes it very unobtrusive. It also gathers less personal, private information such as people's face, words, biometric information that can be possibly used for identification or testimony. However, from this work, it is shown that identification is possible through people's force signature or pressure body-print. Hence it is important to know what level of information this sensing modality can reveal, which would help with improving awareness and regulations in this aspect.

1.2 State of the Art

By the start of this work, thin-film based, single force sensitive resistors (FSRs) have already become widely available [32] [33]. FSRs typically utilize the property of carbonated polymers, that being pressed, its local electrical resistance will change, due to the deformation caused by external pressure force. When combined with compression garments, FSRs can be used to detect muscle activities such as the work in [34] [35].

FSRs can be constructed into a matrix format to map the distribution of the force. Tekscan also offers a range of pressure mapping sensors based on their thin-film based FSRs that has a matrix structure. The products are mainly sold for medical or industrial evaluation and diagnostics.

Fig. 2 shows several works from the academic community of ubiquitous computing, using planar pressure sensing technologies to detect human activities [28] [29] [30] [31]. Dementyev, et al. used a wrist-worn FSR (force sensitive resistor) array to detect hand gestures [28]. [29] proposed a system to detect tongue control gestures with a face-worn TPM patch. Pressure mat placed on the chair surfaces to detect seating postures have also been studied in [30] [36]. Sundholm, et al. have demonstrated that sports exercises can be recognized from a sports mat which sense the pressure distribution in [37]. Schneegass, et al. investigated using a pressure matrix as a sleeve for the forearm to recognize writing gestures in [38].

These projects are all conducted on an ad hoc basis. From the apparatus including hardware and software, to the data mining algorithms, every project is developed from the scratch level. If a unified framework covering the hardware, software and data analysis for planar pressure mapping sensing technology exists, which can be used for a broad spectrum of activity recognition and human-computer interaction studies, these said projects and many

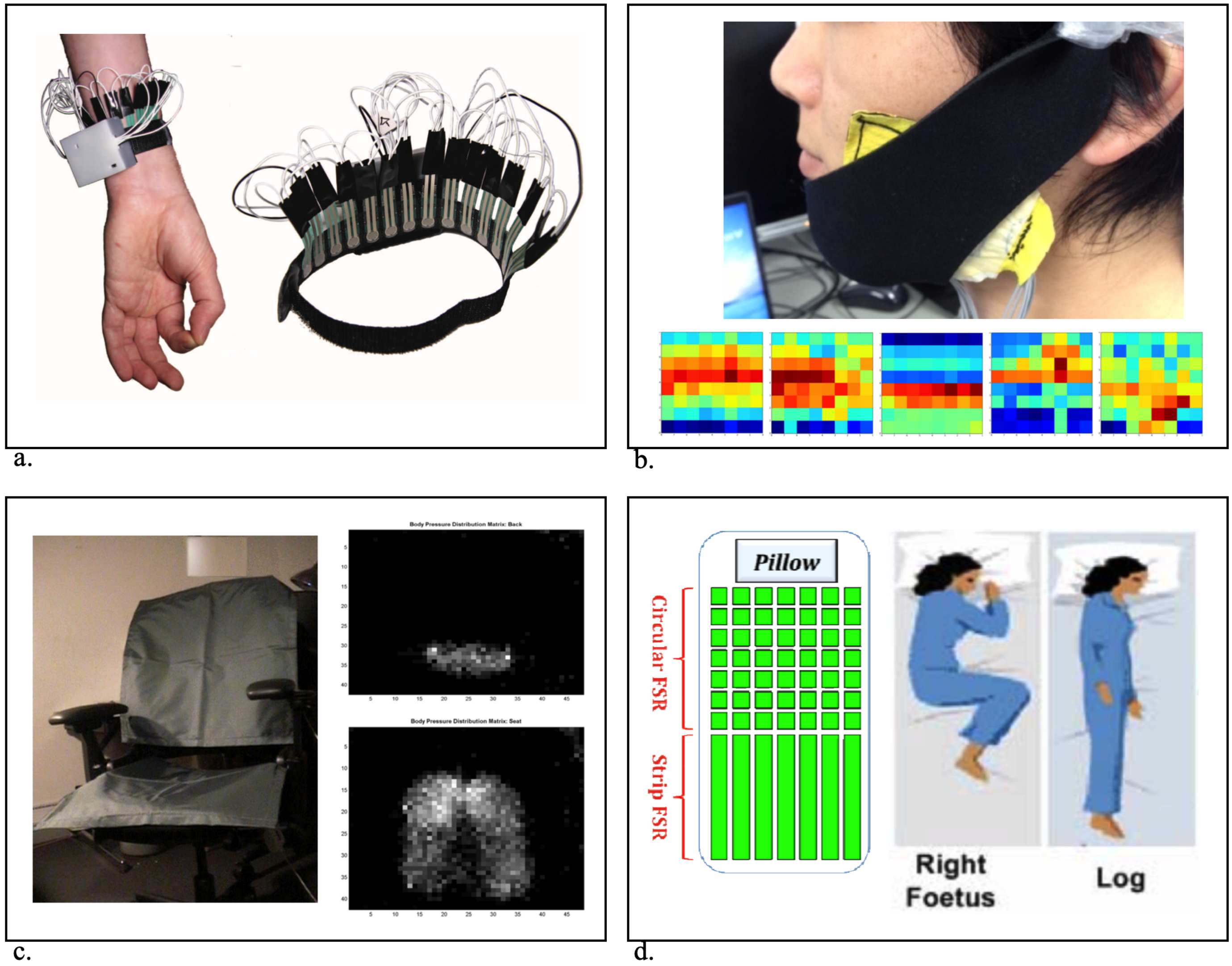

a.
b.
Body Pressure Distribution Matrix: Back
Body Pressure Distribution Matrix: Seat
c.
Pillow
Circular FSR
Strip FSR
Right
Foetus
Log
d.

Fig. 2. Academic related work with planar pressure sensors with ad hoc systems. a. detect hand gestures from a wrist worn FSR array [28]. b. tongue control gestures with a face worn TPM prototype [29]. c. detect seating posture with a commercial pressure mat [30]. d. evaluate sleeping postures with FSR matrix [31].

more ideas can be developed with much less effort. And their findings, especially in how to process the data for specific applications, can also compliment each other.

Most of the state of the art commercial pressure mapping systems have a closed loop driving and data logging system and high cost, which is not suitable as smart objects, furniture and wearable garments for everyday use. And they are typically engineered for specific purposes, with limited adaptation flexibility and lacking an open, modular structure that can be easily modified for a wide range of pervasive and wearable applications. Recently, startup companies such as Sensing Tex, S.L. is also offering development kits up to 48 × 24 matrix for a $90 \times 180cm^2$ area at the cost of 1999 Euros.

For example, if a computer science research team plans to implement a real-time recognition system with their own algorithms, based on above-mentioned systems, they need to start from the undocumented raw data transmission format, because the included software is a closed-loop system designed for experts from other disciplines to inspect the pressure mapping sensor. [39] In this work, the entire textile sensor matrix, data acquisition electronics, software structure is an open, heterogeneous, reconfigurable, hackable framework, therefore it can be easily adapted to any novel applications.

The off-the-shelve systems are also not easily scalable. Developers are restricted to the resolution offered by the manufacturers. For example, if developers want to use two pieces of a 32-by-32 matrix together as a longer matrix, they have to use two separated hardware, and hack the data receiving program to accommodate the double input channels. The framework in this work has a strong emphasis on scalability on multiple levels:

- The individual sensors can be cut and tailored, and even connected together to meet the need of sophisticated sensing shapes such as a glove or a shoe surface. This is also both supported on the hardware and software level, and has been tested through applications. In a specific application of a smart-belt, the ADC input pins are also reprogrammed as output

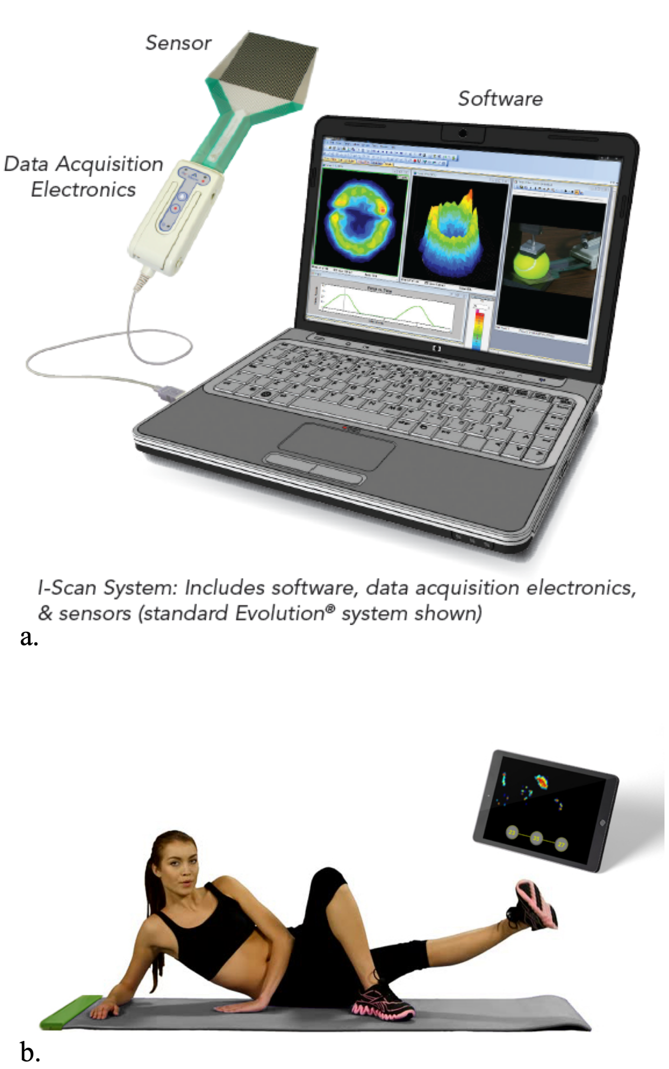

Sensor
Software
Data Acquisition Electronics
I-Scan System: Includes software, data acquisition electronics, & sensors (standard Evolution® system shown)
a.
b.

Fig. 3. The advertising materials from a.Tekscan and b. SensingTex.

pins so that the hardware that was designed to support a 32-by-32 pseudo-square-space is actually driving a 60-by-4, long matrix which exceeds the pseudo-square-space but meets the pin counts of the actual hardware.

- Heterogeneity and parallelism guide the framework development, which makes it easy to scale up the sensing area with connecting multiple hardware together. The software takes care of combining parallel input on the *Tier 1* level and the rest of the framework on the higher abstract level can be unchanged.

Most of the off-the-shelf solutions also lack wireless capabilities, whereas, in this work, stable Bluetooth transmission, power management, and optimization as a mobile Li-Po battery powered device have enabled a range of applications with the mobility that only wireless systems can provide, especially in scenarios out of the labs.

Capacitive principles can detect very small motions due to its high sensitivity [40] [41]. Capacitive sensing also does not require the material's property to change with force, instead, it measures the distance change or dielectric change between two plates of the capacitor. Thus only metallic conductive material is required. And the spacer material requires only mechanical properties that would change its thickness with the force, instead of electrical properties. However, measuring a single capacitive node normally requires sine signal sweep excitation, which results in relatively longer sampling periods. This therefore makes it more difficult to scale up a capacitive matrix system because the sampling rate will suffer exponential degradation as dimension increases. [42]. In [43], a capacitive textile matrix is demonstrated to measure seating posture. Every node is sensed at 90Hz; however, with multiplexing, the total 240 nodes requires 3 seconds to sample (3Hz refresh rate).

1.3 Contribution of This Work

The contributions of this work on textile pressure mapping (TPM) technology are two-fold:

1) A full-stack framework is constructed to develop TPM systems from the essential hardware to the upper-level software and machine learning. The framework is an encapsulation of design knowledge and guidelines, a multi-layered tool, and also an ensemble of examples. Unlike other ad-hoc studies using pressure sensing in application recognition, this work uses a unified framework to cover the full scope of application recognition including the ambient, object and wearable subspaces.
2) TPM is extensively evaluated for activity recognition in three scopes: smart environment, smart objects, and smart wearables. Empirical knowledge of utilizing TPM is gained throughout these scopes.

The TPM framework covers from multiple angles to develop TPM systems.

- Through design space exploration, five design dimensions (propagation, intention, coverage, mobility, and sampling) and three sub-spaces (ambient, object and wearable) are outlined to help specify and direct the designing process of TPM systems.
- A general hardware architecture to drive the TPM sensors is proposed, with the emphasis on the flexible scalability of the sensing surface area. Multiple versions of the architecture are implemented, either with parallel hardware such as field-programmable gate arrays (FPGAs) or power efficient microcontrollers, according to the need of different design space requirements.
- The framework includes a heterogeneous software stack, which is consisted of four tiers to cover the needs of driver, signal processing, machine learning, output and feedback.
- The framework also evaluates different algorithms for data processing and machine learning with TPM sensor outputs. A preset of features, the TPM feature set, is presented specifically for data mining with TPM's spatial-temporal data format, including spatial domain features (8x10), temporal and frequency domain features (17x39), and deep neural network features.
- The effectiveness of feature subsets is analyzed using neighborhood component analysis to evaluate their

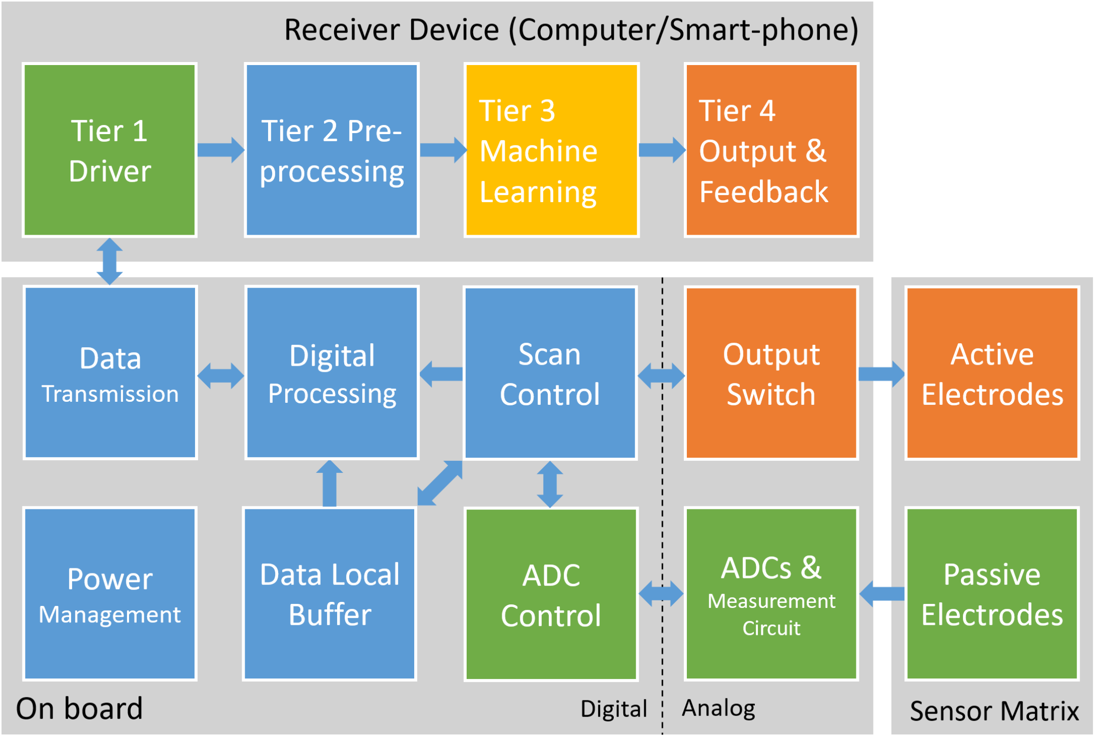


Fig. 4. Brief overall structure of the TPM framework

contribution with a unified process on datasets from different design subspaces. Thus a new dataset with a TPM system can be evaluated directly using the TPM feature set, and further optimized by feature selection.

The TPM framework is designed to be easy to scale and developer friendly. Fig. 4 shows the brief structure of the TPM framework's actual tool chain part, which consists of three sectors: Sensor Matrix (Textile), Hardware (Electronics), Software (Receiver Device). Developers can first directly use this framework to design and validate a TPM system in novel activity recognition scenarios, and then make their own customizations for further evaluation and deployment. The TPM framework is developed iteratively with a series of empirical studies, where the TPM sensing modality is validated for activity recognition tasks.

Force is an essential physical interaction that causes all of the change of motions. TPM sensors can be utilized in activity recognition tasks by measuring the planar pressure (force applied on surfaces). Through the empirical studies, significant findings prove that TPM technology has potential in various application scenarios of the field of ubiquitous activity recognition.

TPM can detect activity from the pressure body-print on various surfaces, since earth's gravity is constantly pulling us onto supporting surfaces. TPM sensors can cover not only flat surfaces, but also soft, curved and cushioned surfaces.

Touch is a fundamental interaction and most often carries emotions through the minor changes in force patterns. TPM can be unobtrusively installed on surfaces and objects for not only gesture interface, but also sensing emotional related physical interactions.

TPM can be integrated into comfortable fitting garments. Muscle activity can be detected through the planar pressure change on the garment that is caused by the wearer's body motions. This method is also called pressure mechanomyography (MMG) in the biomedical discipline. Compared to the electrically coupled electromyography (EMG), TPM MMG has the advantages of no skin-electrode contact, robust against ambient electromagnetic interference, easy to form a mesh to over bigger area with fine spatial granularity, etc.

Specifically, the following significant findings of to what extent TPM technology can be used to detect activities are discovered:

In the ambient subspace:

- A table cloth made of TPM matrix can detect 8 classes of micro-interactions of acquiring food, which is propagated from the cutlery through the plates to the table, with up to 0.946 accuracy (0.125 chance level).
- A 2-by-1 meter sports mat with a 128-by-64 TPM matrix under the surface can detect 9 exercise categories from the dynamic time sequence of body-prints with 0.804 accuracy (0.111 chance level), and also distinguish 47 trivial variations of these exercise categories with 0.387 accuracy (0.021 chance level).
- A carpet with TPM under the surface (the same from the said sports mat) can identify the person from 13 user groups by the force signature of their gaits with up to 0.877 accuracy (0.077 chance level).

In the object subspace:

- An office chair with a 32-by-32 TPM matrix (1.5cm pitch) covering the back surface can detect 10 different postures from the back-print with 0.795 accuracy (0.100 chance level).

- A pressure sensitive garment for objects such as robot arms can detect people's emotional interactions such as poking and stroking. Using a 20-by-20 TPM matrix (1cm pitch), with 7 categories of defined interactions, the classification accuracy is up to 0.933 (0.143 chance level).

In the wearable subspace:

- An 8-by-16 TPM matrix (1cm pitch) fixed on wearer's thigh with a stretchable band can detect 4 workout plus 3 non-workout activities inside the gym with 0.817 accuracy (0.143 chance level).
- A soccer shoe covered by TPM sensors across its surface, with 33 sensing points sampling at 550Hz, can detect the angle and intensity of shooting a ball from 17 variations at 0.988 accuracy (0.059 chance level).

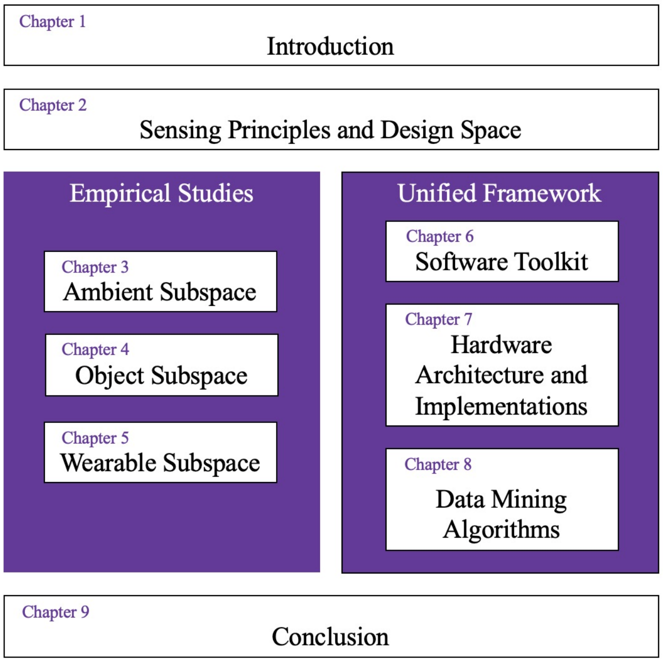


Fig. 5. Dissertation structure

1.4 Dissertation Structure

This dissertation first introduces the basic knowledge, including the sensing principle and the design space, of using TPM sensors for activity recognition systems. From a bottom-up approach, individual empirical studies are carried out in different scenarios covering the ambient, object and wearable scopes. Then the general hardware, software and algorithm framework is constructed by comparing and summarizing the empirical studies from a top-down approach.

Chapter 2 first introduces the sensing principle. The textile pressure mapping matrix emphasizes the integration of electrically conducting polymer composite fabrics to extract useful pressure information. The functioning principle of interconnected sensing matrix is explained via simulation. Then the design space of utilizing TPM sensors is discussed, which includes five dimensions: *Propagation*, *Intention*, *Coverage*, *Mobility*, and *Sampling*.

As introduced in Chapter 1, the design space can also be divided into three sub-spaces: *Ambient*, *Object* and *Wearable*. In every sub-space, several empirical studies are carried out to evaluate what sorts of information TPM sensors can provide. Chapter 3 explores the *Ambient* space. After a discussion of the ambient space, three empirical studies is introduced including TPM augmented table cloth, sport mat and ID carpet.

In Chapter 4, the *Object* space is explored. Smart objects are equipped with TPM sensors in different studies to detect user activities, including a smart seat cover that detect the user's sitting posture, robot skin patches that recognizes emotional related touch interactions with human. This chapter also contains a student workshop, in which students use the TPM framework to develop and evaluate their own smart object projects.

The *Wearable* space is discussed in Chapter 5, introducing how TPM can be used as smart garments. In particular, TPM can be used as means of performing mechanomyography to measure muscle activities. A smart soccer shoe can also detect the ball impact by the TPM sensor on its upper surface.

The empirical studies from Chapter 3 Chapter 4 and Chapter 5 all use different hardware, software and algorithms. If an hierarchy of those aspects is to be extracted from these studies, it would be a bottom-up structure. Chapter 7 looks at the hardware aspect from top-down. This chapter fist proposes a general hardware architecture, then discuss the problems and solutions of implementing the driver electronics for data acquisition. To achieve the goal that the framework should be flexibly scaled up and down, two branches (portable and high performance) of electronics are finalized.

The general software implementation is explained from top-down in Chapter 6. Implementation mainly considers the requirements for meeting real-time data acquisition, classification and visualization during development and deployment. A heterogeneous software system framework, as well as an all-in-one independent C++ executable application are developed as examples.

Chapter 8 discusses the data processing and machine learning algorithms on the theoretical level without concerns of the trade-off of different software and hardware implementation environments (e.g. sampling rate, programming languages, libraries, dependencies, etc.). After all the empirical studies are explained in details in previous chapters, this chapter looks at the data mining processes from a top-down view and focuses on a generic algorithm workflow and investigating which part of the information are more important in specific applications.

Finally, Chapter 9 concludes the entire dissertation with a summary of key contributions, and discussions on limitations and outlook.

Chapter 2 Sensing Principles and Design Space Exploration

This chapter first introduces the textile sensing principles and the types of materials that can be used for constructing TPM sensors. An overall view of the design space is discussed thereafter, regarding how textile pressure mapping can be deployed to acquire information about user activities.

The primary sensing principle of planar pressure mapping can be generalized as: an electrically measurable variance factor that is related to the pressing force onto the sensitive material. The variance can be electrical conductivity (e.g., the resistance of flexible resistor materials), electrical capacitance (e.g., two electrode plates separated by a flexible spacer), optical luminance (e.g., light leakage of two knotted optical fibers), etc. The work of this dissertation is focused on resistance based fabric material, a fabric CarboTex produced by Sefar AG, together with their fabric with metallic stripes developed during the EU project SimpleSkin. Apart from those fabrics, low-cost materials can also be used such as electrostatic discharge protection materials.

To use TPM for activity recognition, the designing of the system should consider aspects including:

- How the force is propagated from the source to the sensing fabrics.
- Whether the captured information is intended or unintended performed by the users.
- The coverage relationship between the TPM sensor area and the user(s).
- The mobility requirements of the TPM sensor hardware.
- The sampling rate, which can be limited by many factors such as activity nature, hardware and software constraints, algorithm requirements, etc.

The design space is explored regarding those aspects. Concerning the form of implementation, the design space can be divided into three sub-spaces: ambient space, object space, and wearable space.

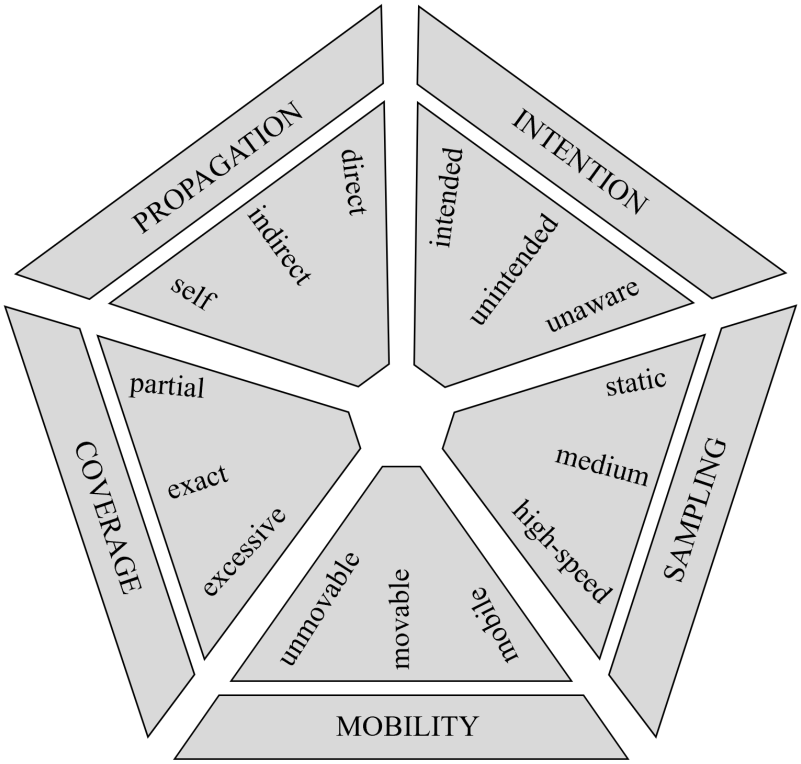


Fig. 6. TPM design space with its five dimensions

The structure of this chapter:

- The sensing principles, materials, and their relative characteristics are first introduced;
- A brief but in-depth simulation is presented (more in-detailed simulations are already documented in [44]);
- The design space is analysed in five dimensions: propagation, intention, coverage, mobility, and sampling.

2.1 Sensing principles

The core principle of electrical resistance based pressure mapping is the special property of electrically conducting polymer composites (ECPCs), that their deformation, which could be caused by either tension or pressure, will cause its electrical impedance in the vicinity of the deformation to change. [45] ECPCs are typically a colloidal form of conductive particles (e.g., carbon black, metal and metal oxide particles) dispersed into a polymer matrix. [46] [47] Deformation will change the polymer matrix's molecular structure so that the suspended conductive particles (filler particles) form new conductive paths, which in turn changes the local electrical impedance on the supermolecular level. [48] Under 20-70 °C, the temperature dependency of carbon black polymer is found to be very weak; therefore the measured resistance variation of the composite can be practically directly linked to the pressure and tension. [49]

The pressure force can be measured by the volume resistance, that is, the electrical resistance from the top to the bottom side of a thin film ECPC material. In order to practically measure the resistance with analog-digital converters (ADC), a voltage divider is used as shown in Fig. 7(a detailed discussion of why Wheatstone bridge is not superior to a voltage divider, in this case, can be found in [50]). The signal source impedance (R0) should be matching the ADC input impedance, which in reality varies based on models and manufactures, and is typically not specified except for high-performance units. Thus R0 is usually determined empirically. It is also important that the source impedance stays relatively stable during the sampling, so the configuration in Fig. 7 with the reference resistor R0 connected to ground is a better choice compared to exchanging Rx and R0, even though both configurations are theoretically possible.

To map the pressure force of the material, we can measure the volume resistance distribution across the area as illustrated in Fig. 7. The volume resistance can be measured by placing parallel stripe electrodes on both sides of the material; each side is perpendicular to one another. Name one side of the parallel electrodes as *active electrodes* (offers voltage stimuli) and the other side as *passive electrodes* (each connected to an ADC input with a voltage divider). Two layers of operation loops complete the scanning cycle:

- turn on one active electrode, while connecting the other active electrodes to the ground or high impedance.
- measure the voltage at the passive electrodes one by one.

However, the matrix structure is not simply a 2D array of the individual volume resistors from each cross-point: between two neighboring electrodes on the same side, there is a measurable electrical resistance (parasitic resistance Rp in Fig. 7); even only considering a 2D array of the individual volume resistors, the electrical stimuli and ADC input are at specific points of an interconnected network, which involves all the resistors between this point and the ground.

In [51], a MEMS (micro-electro-mechanical systems) sensor is proposed with a similar strategy, using strain gauge arrays on a microscopic membrane to detect microscopic 3D objects.

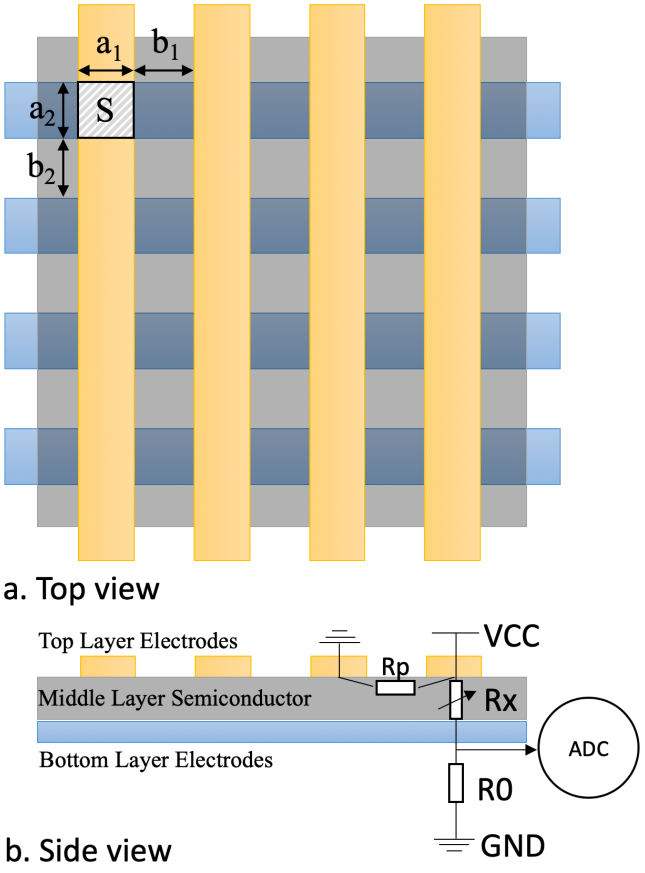


Fig. 7. The matrix structure of measuring the resistance distribution of a semi-conductive sheet

Detailed simulation considering the matrix network can be found in [44], Chapter 2. To summarize the findings from the simulation:

- The scanning cycle with two layers of operation loops mentioned above generates a voltage mapping that can represent the sensor resistor values.
- The electrical property of the active electrodes is relevant to the imagery result, especially an influence to the range and contrast; in this regard, grounding the inactive active electrodes is better than connecting them to high impedance.
- The parasitic resistance acts similar to a spatial average filter with a small blurring effect; therefore special treatment is not necessary.

2.2 Low cost ECPC materials

One of the goals of this work is to promote pressure mapping application on a broader scale in everyday environment, therefore, low cost materials is crucial. ECPC materials are quite ubiquitous even though their force-sensitive properties are usually not exploited. For example, the plastic package material Velostat for ESD (electrostatic discharge) protection for electronics are typically made of carbon polymer foils. To dissipate the electrostatic charge without short-circuiting the electronics it's protecting, the ESD-safe bags usually have a few kilo- to a few megaohms volume resistance per square centimeter when the electrodes are slightly fixed on the material to maintain full contact. As external force press on the electrodes, the resistance decreases down to mere 1 to 5 % of the slack state. Some early prototypes are made using ESD safe foils and foams as shown in Figure 8, 9. One major drawback of these cheap ESD protection materials is that they are not produced for sensing purpose. So the chemical mixture process has a big range of tolerance, possibly resulting in varying levels of conductivity from different batches even of the same supplier article number.

The CARBOTEX from Sefar AG is a series of anti-static fabrics, that can also be classified as an ECPC material. [52] It is originally produced for filtration in flammable gas and liquid substances, as a fabric sheet of carbon-polymer. Except for the fabric level flexibility, CARBOTEX is also washable and air-permeable.

Figure 11 shows the volumetric resistance per $7 \times 7mm^2$ to load weight relationship. The test method is: a sensor patch of 7-by-7 with 1.5cm electrode pitch is placed on a flat weight scale, while a manual force is placed on five different points of the matrix (one at the center and one at each corner) through a probe, one point at a time. At the meantime, the resistance of the pressed point is measured by a multimeter. However, manual force is not ideal in terms of precision and stability; a more comprehensive testing method is described in [53] for measuring the repeatability of force sensors, which requires more sophisticated equipment that applies highly repeatable force on the same area. Nonetheless, such tests only show the ideal situation where the sensors are always fixed in the test-bench to study the absolute properties of the sensing material.

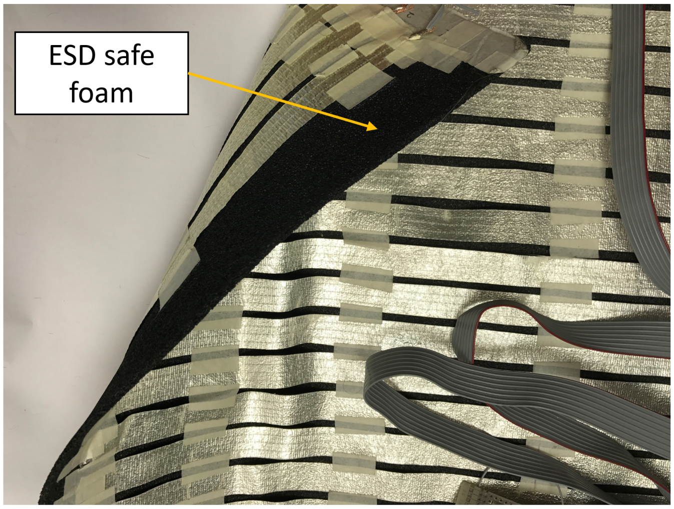


Fig. 8. Early prototype with foam ESD safe package material

Most of the empirical studies in this dissertation use CARBOTEX as the sensing material. It has been proven to be very suitable for activity recognition and touch interaction tasks. The absolute sensor value is usually not needed, since normalization is typically a standard pre-processing operation. As the dynamic change and sensitivity are highly related to the activity, the pressure mapping result can resemble the real-time movement and state of the interaction between the people and sensor-equipped objects or garments. In those applications, the fabric structure is usually being moved: the fabric layers may shift or form wrinkles, and different layers may move separately, especially in some application that the fabrics are not attached to a sturdy surface and being frequently bent. However, what is important for activity, gait or gesture recognition is the repeatability of the dynamic change in pressure, that after standard normalization, data from different instances of the same activity would be similar to each other, and contains unique features for different activities. The training data can also offer the machine learning algorithms sufficient reference for classification in repeatable experiments.

2.3 Textile Integration of the TPM Fabrics

The general structure of TPM is a matrix structure formed by two perpendicular groups of parallel conductive stripes as shown in Fig. 7. The conductive stripes are both electrodes and the connection paths. Most of the budget flexible conductive material relies on the non-conductive side (usually a layer of plastic film laminated with the metal film layer)

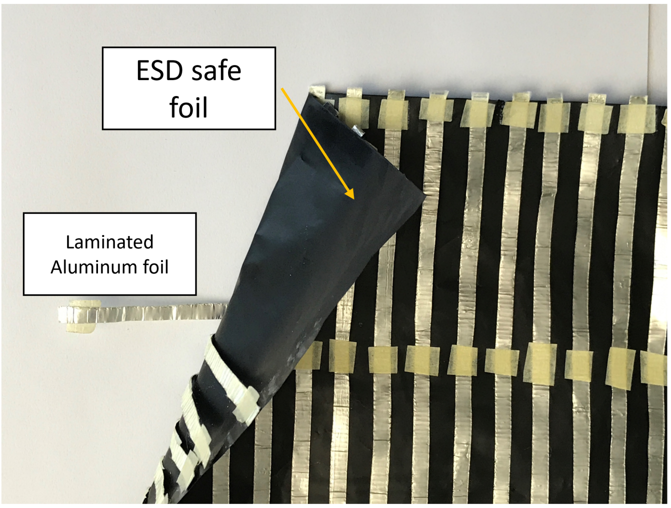


Fig. 9. Early prototype with thin film ESD safe package material

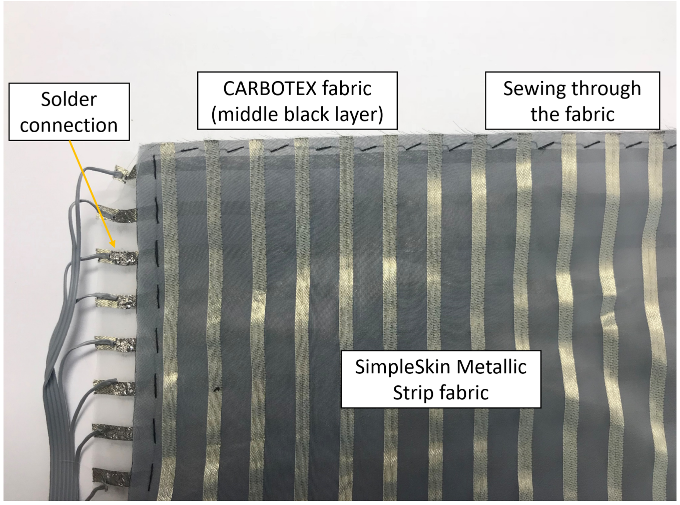

Solder connection
CARBOTEX fabric (middle black layer)
Sewing through the fabric
SimpleSkin Metallic Strip fabric

Fig. 10. Full fabric sensor with SimpleSkin fabrics produced by Sefar.

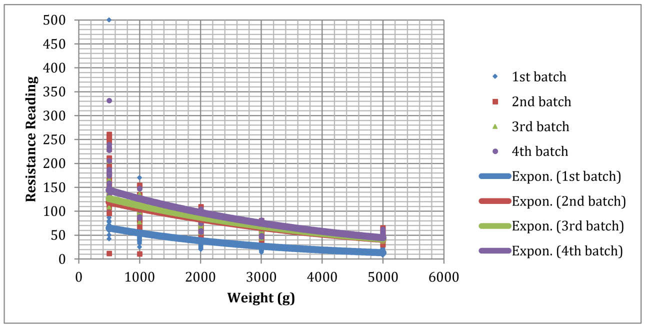


Fig. 11. Volumetric resistance and load characteristic test of the CARBOTEX fabric.

for the flexibility; thus the conductive side should face the ECPC material.

The surface area of the intersection between two stripes directly relates to the effective resistance. The sequential resistor in the measurement circuit should match the effective resistance. The effective resistance also decides the power consumption when a DC stimulus powers the active electrodes.

A budget method to construct such a matrix is using conductive (typically copper or aluminum) tapes and carbonated film material (normally for ESD protection). Conductive tapes can be fixed on a conventional non-conductive plastic or even paper sheet as substrate, and cover the ECPC material in the center. Examples are shown in Fig. 8 and Fig. 9.

Sefar's SimpleSkin metallic stripe fabric is a mass-producible fabric with the metallic stripes exposed on the one side of the fabric and covered by a non-conductive polymer substrate on the other. Together with CARBOTEX, these two types of fabrics can construct whole fabric pressure sensors. The fabric sensors can also be sewn with conventional tailoring methods to fix on different surfaces. During the EU project SimpleSkin, a washing test is carried out by DFKI and iTV. As the result shown in Fig. 11, the TPM sensors are still sensitive to force changes after 40 washing cycles (with electronics disconnected), only with magnitude shifts of the response curve after being washed.

The integration methods do not have to be restricted by the perpendicular matrix. For example, in 5.2, the soccer shoe surface is made of three diamond shape patches to accommodate the shape of the shoe. It can also be tailored into smaller patches to fit on irregularly shaped objects which a single matrix would not fit, such as a glove. This aspect is further elaborated in the virtual surface concept in Chapter 7.

The fabrics used in the TPM sensors in this work are not stretchable, although a stretchable sensor structure can be more versatile. Thus when an elastic garment is required, the sensing fabrics are placed on elastic fabrics as a substrate, the sensing region is then not stretchable, but the rest of the substrate fabric is. On the one hand, the direct pressure interaction can be measured; on the other, if the circumference of the closed stretchable garment increases, the garment will have increased tension, which will also increase the initial pressure on the sensing area. This will be further discussed in the *propagation* design space dimension.

The connection from the fabric to the electronics is usually implemented by standard electronic ribbon cables. For the time being during this work, it is the most efficient and stable way to connect big numbers of connections. The wires can be soldered directly onto the metallic stripes from the Sefar SimpleSkin fabric. The other end of the cables can be fixed on clamp-on style headers, which can be interfaced with matching headers on the electronics. During the SimpleSkin project, iTV has also developed a prototype of embroidery wire connection, with solder bumps for connection. However, the method was still in its developing stage and thus is not suitable for the various applications this work involves. We believe one of the major future challenges in smart fabric integration is easy and cheap to access wiring solutions.

2.3.1 Matrix Simulation

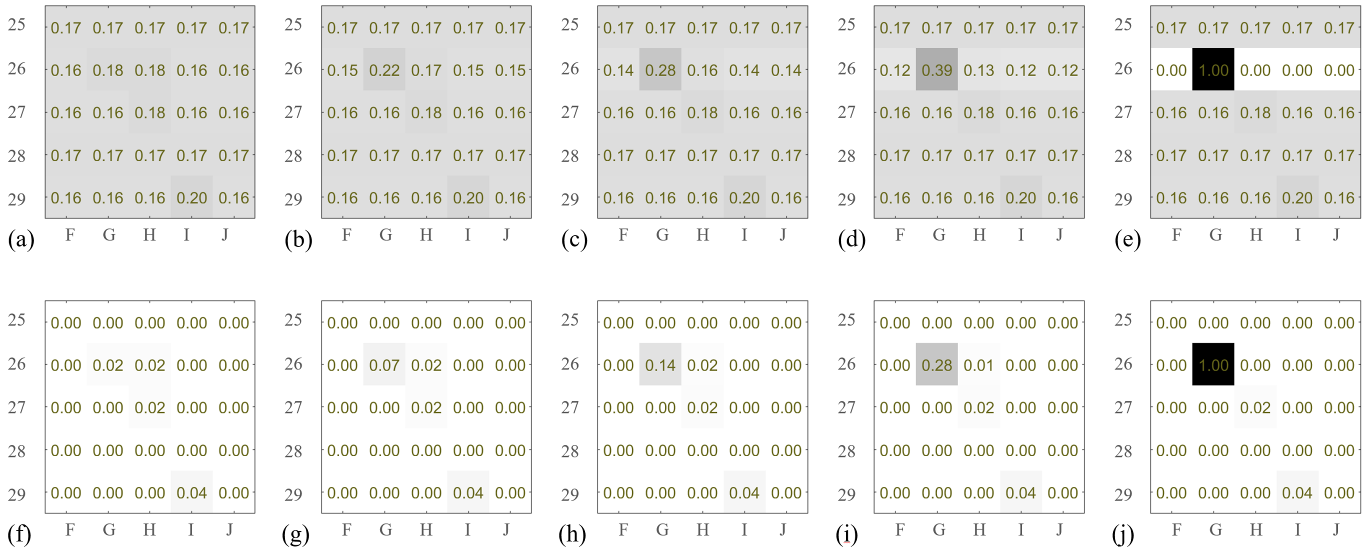
Next, we consider the matrix structure as a circuit network. Take a 5-by-5 matrix as an example. The equivalent resistor matrix is drawn in Fig. 13. Assume a situation that the whole matrix is evenly pressed to a default level ($V_{default} = 1k\Omega$), then several points are differently triggered. Next one node $G26$ is changed from $0.9k\Omega$ to $0.1k\Omega$ with $0.2k\Omega$ steps; eventually, it is decreased to an infinitely small value (near Zero). The results in Fig. 12 a-e are reconstructed imagery of the resistors' matrix. The values in every cell are the normalized voltage valuesV_{sense}/V_{CC}. First, the result values increase as the resistance decreases. From Fig. 11, the resistance decreases as the applied pressure increases. Thus the voltage value overall can be interpreted as related to the pressure value. Second, we can observe that the changed resistor influences the remaining nodes connected to the same passive electrode (rows in Fig. 12), resulting in smaller V_{sense} in those remaining nodes than without this changed resistor. This phenomenon is most evident in the second row (connected to passive electrode 26). To compensate for this, we can use the same concept of the relative contrast introduced before. For every row, the values are subtracted by the minimum value of the row:

$$\frac{V_{sense} - V_{row\ minimum}}{V_{CC}}$$

Fig. 12 f-j shows the result after removing row minimums.

In the real application, this effect exists only in some versions of the hardware. It is due to the physical characteristics of different component devices. Specifically, the row influence is noticeable in hardware version F1, F2, F3 mentioned in Table 17, but not observable in the remaining versions. By ruling out variables from hardware iterations, the reasons could be:

1) Typically one ADC is connected to multiple passive electrodes via analog switches or multiplexers. When the passive electrodes are not being sampled by the ADC, how they are handled is typically not specified by the manufacturers. They can have an impedance that is influencing the remaining of the circuit.
2) In hardware version F3, the active electrodes are driven by an analog multiplexer. When it is not the current stimuli, the output of the channel is also



(a)

	F	G	H	I	J
25	0.17	0.17	0.17	0.17	0.17
26	0.16	0.18	0.18	0.16	0.16
27	0.16	0.16	0.18	0.16	0.16
28	0.17	0.17	0.17	0.17	0.17
29	0.16	0.16	0.16	0.20	0.16

(b)

	F	G	H	I	J
25	0.17	0.17	0.17	0.17	0.17
26	0.15	0.22	0.17	0.15	0.15
27	0.16	0.16	0.18	0.16	0.16
28	0.17	0.17	0.17	0.17	0.17
29	0.16	0.16	0.16	0.20	0.16

(c)

	F	G	H	I	J
25	0.17	0.17	0.17	0.17	0.17
26	0.14	0.28	0.16	0.14	0.14
27	0.16	0.16	0.18	0.16	0.16
28	0.17	0.17	0.17	0.17	0.17
29	0.16	0.16	0.16	0.20	0.16

(d)

	F	G	H	I	J
25	0.17	0.17	0.17	0.17	0.17
26	0.12	0.39	0.13	0.12	0.12
27	0.16	0.16	0.18	0.16	0.16
28	0.17	0.17	0.17	0.17	0.17
29	0.16	0.16	0.16	0.20	0.16

(e)

	F	G	H	I	J
25	0.17	0.17	0.17	0.17	0.17
26	0.00	1.00	0.00	0.00	0.00
27	0.16	0.16	0.18	0.16	0.16
28	0.17	0.17	0.17	0.17	0.17
29	0.16	0.16	0.16	0.20	0.16

(f)

	F	G	H	I	J
25	0.00	0.00	0.00	0.00	0.00
26	0.00	0.02	0.02	0.00	0.00
27	0.00	0.00	0.02	0.00	0.00
28	0.00	0.00	0.00	0.00	0.00
29	0.00	0.00	0.00	0.04	0.00

(g)

	F	G	H	I	J
25	0.00	0.00	0.00	0.00	0.00
26	0.00	0.07	0.02	0.00	0.00
27	0.00	0.00	0.02	0.00	0.00
28	0.00	0.00	0.00	0.00	0.00
29	0.00	0.00	0.00	0.04	0.00

(h)

	F	G	H	I	J
25	0.00	0.00	0.00	0.00	0.00
26	0.00	0.14	0.02	0.00	0.00
27	0.00	0.00	0.02	0.00	0.00
28	0.00	0.00	0.00	0.00	0.00
29	0.00	0.00	0.00	0.04	0.00

(i)

	F	G	H	I	J
25	0.00	0.00	0.00	0.00	0.00
26	0.00	0.28	0.01	0.00	0.00
27	0.00	0.00	0.02	0.00	0.00
28	0.00	0.00	0.00	0.00	0.00
29	0.00	0.00	0.00	0.04	0.00

(j)

	F	G	H	I	J
25	0.00	0.00	0.00	0.00	0.00
26	0.00	1.00	0.00	0.00	0.00
27	0.00	0.00	0.02	0.00	0.00
28	0.00	0.00	0.00	0.00	0.00
29	0.00	0.00	0.00	0.04	0.00

Fig. 12. Matrix circuit simulation result.

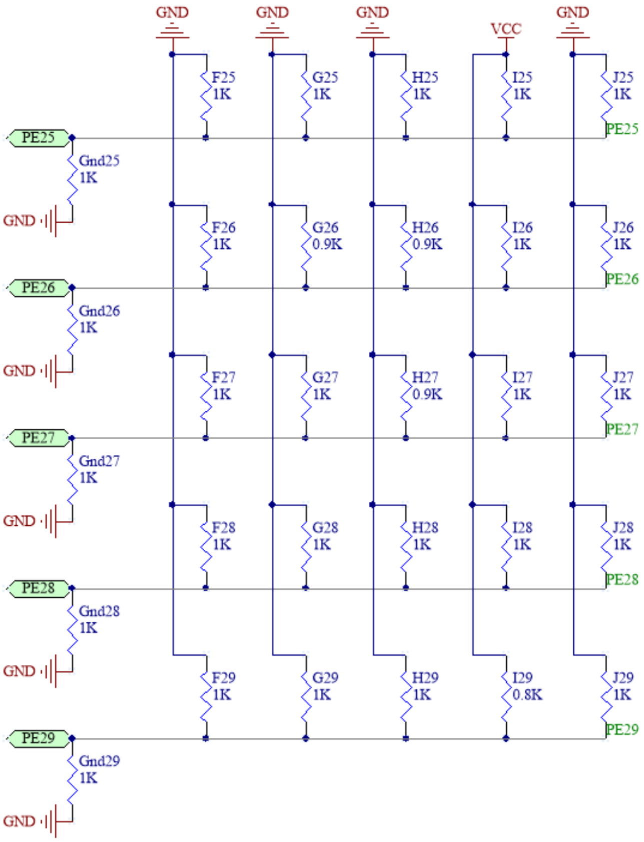


Fig. 13. Matrix circuit simulation schematic.

unknown. Through experience, the best practice is connecting the idle active electrodes to the ground.

2.4 Design Space Exploration

Designing a sensing system with TPM for activity recognition is a multi-dimensional problem. Unlike integrated circuit (IC) sensors such as IMU sensors, textile pressure sensors rely on a piece or pieces of sensing fabrics. Pieces of fabrics are more expansive and mobile than single IC sensors, which increases the complexity of the design process. Where and how to install those fabrics to retrieve activity information is decided by many factors. Fig. 14 shows several examples of use cases and design concerns in three subspaces: ambient, object and wearable. One should consider: (1) the *propagation* of force from the source of the activity to the sensing fabrics; (2) the *intention* of the user when the activity or interaction happens; (3) the *coverage* relationship between the sensing fabrics and the targeted user or objects; (4) the *mobility* of the fabric sensor during the activity and interaction; (5) and the *sampling* rate of the fabric surface.

2.4.1 propagation

For user activity recognition, there are generally three sectors as shown in Fig. 16: (1) target users, (2) objects or other users that the target user is interacting with, and (3) environment or static objects surrounding the target user. Additionally, target users may also wear smart garments. TPM sensors acquire activity information through the force from the target users to the other sectors in the scope. During the design process, how the force is propagated is a differentiating factor.

1) Surface force can be a *direct* cause or result of the activity - it can initiate a motion or deformation, or the consequence of contact. In this case, the TPM sensors are generally placed on the surface where the direct interaction happens, such as on objects' surfaces to detect touch gestures.
2) It can also be *indirectly* propagated from the activity source to the sensor surface through a medium. Such as on a tabletop of a dining table, the actions of cutting, acquiring food with cutlery on the plate will propagate through the plates to the table surface.
3) The sensor fabric can be part of smart garments, and the motion of the surface of the garments would

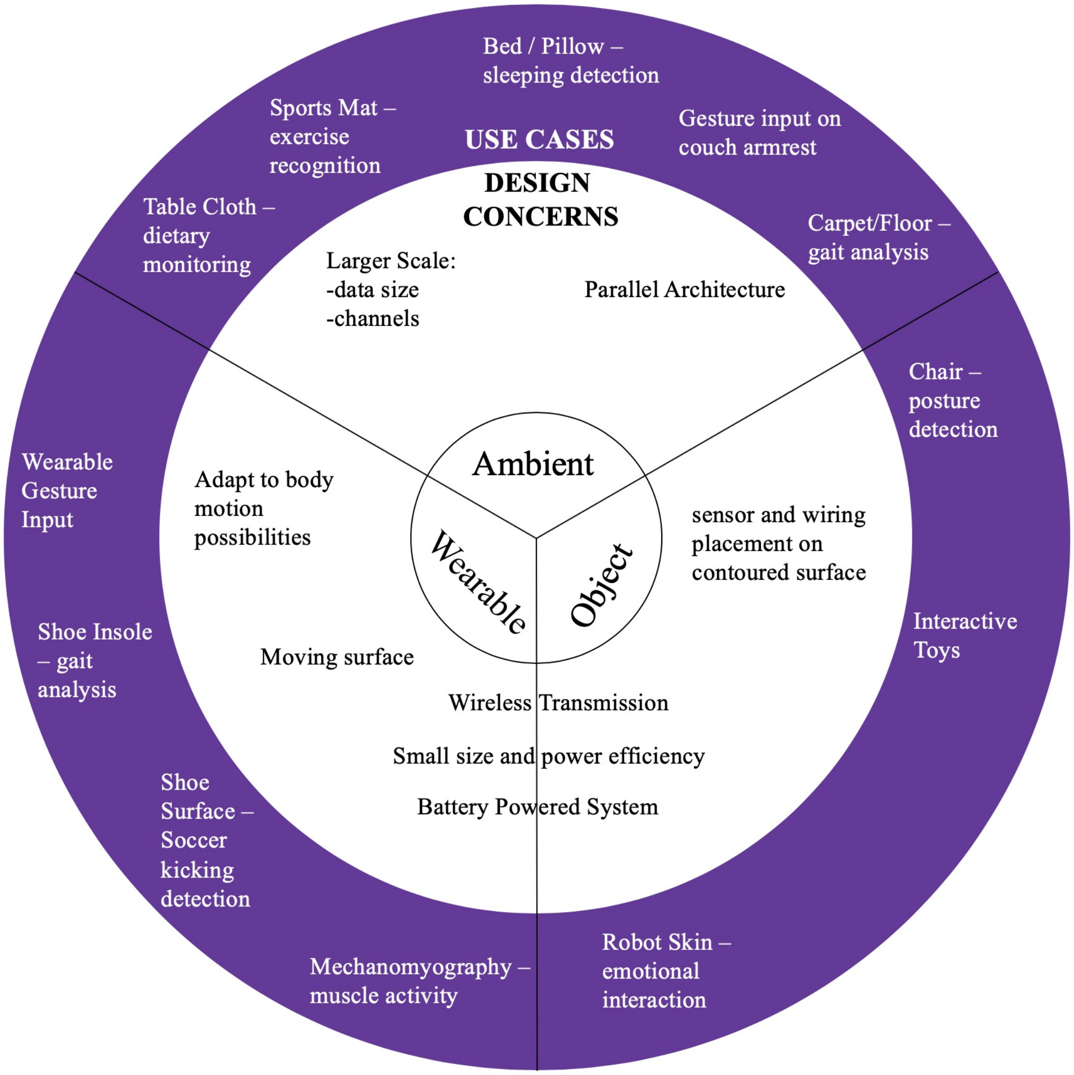

USE CASES
DESIGN CONCERNS
Ambient
Wearable
Object
Bed / Pillow – sleeping detection
Sports Mat – exercise recognition
Gesture input on couch armrest
Table Cloth – dietary monitoring
Carpet/Floor – gait analysis
Larger Scale: -data size -channels
Parallel Architecture
Chair – posture detection
Wearable Gesture Input
Adapt to body motion possibilities
sensor and wiring placement on contoured surface
Interactive Toys
Shoe Insole – gait analysis
Moving surface
Wireless Transmission
Small size and power efficiency
Battery Powered System
Shoe Surface – Soccer kicking detection
Mechanomyography – muscle activity
Robot Skin – emotional interaction

Fig. 14. The design concerns and example use cases in the ambient, object and wearable subspaces.

cause pressure force changes to the sensor fabric. In this case, the target users do not interact with any external objects, users or environment, but with the sensing fabric itself (*self*). An example is that when users wear fitted garments, as they move their body, they would feel the pressure from the garments - the garments can then measure this pressure force.

In the first and second propagation cases, the TPM sensor can either be installed on the target user or the objects and environments. Moreover, for the third propagation method, the sensor is attached to the target user, typically with a fitted garment that covers the moving parts of the body during the activity under investigation.

2.4.2 Intention

In an activity recognition system design with TPM, the pressure force can be *intended* by the user or *unintended* as the user is aware of the sensor; while the user can also be *unaware* of the sensor system. Intended interactions are generally human-machine input or control gestures, such as a textile force-sensitive trackpad or sleeve [54] [38]; or directly picking up and manipulate an object [55]. Unintended interactions are usually 'side-effects' of the targeted interaction, for example, the change of the body print while a person is doing exercise on a sports mat [37] or the change of fabric surface pressure of a fitted garment caused by the user's body motion [56]. Unaware intentions are naturally unintended; however, with the unawareness, the sensor is usually disguised as ordinary objects, and the users are not necessarily notice their existence. This can be the force signature detected from under a chair while the person is doing different actions [57]; or detecting a person's identity from their footprints' pressure profile [58].

The intention dimension is thus on the information level, not the physical level. The same TPM sensor can be used to recognize intended, unintended or unaware interactions. For example, a mat for detecting different sports postures can also be used to identify the person's footstep; it can also be used as a foot gesture input [59]. Thus for ethical and privacy reasons, it is essential to identify what kind of information the TPM sensors can acquire from the users. Because some of the activities that can be detected from

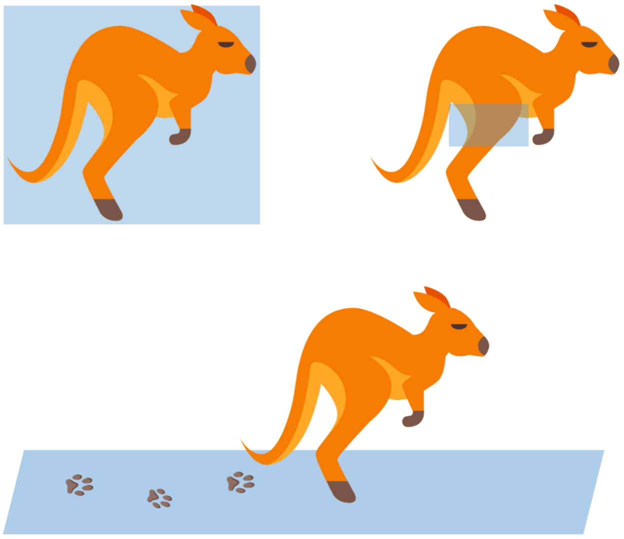


Fig. 15. In general the sensor can be placed in three fashions: exact coverage, partial coverage and excessive coverage.

analyzing the sensor data may not be what the users are aware of, or what the users are willing to disclose.

2.4.3 Coverage

In general, there are three categories of the relationship between a TPM sensor and the target object/user, as shown in Figure 15: the sensor exactly matches the target object/user (*exact*), only partially covers the target (*partial*) and significantly larger than the target (*excessive*). This categorization is made because for different type of relationship a special pre-processing step may be required: for partially covering, it needs to be taken into consideration that the sensor may move during action; for a canvas style, segmentation, tracking and tracing of the moving target may be required.

The relationship is usually defined as soon as the sensor installation is designated. However it also changes according to specific applications: a sportmat for mostly lying-down exercises can be considered as an exact match, since the user generally stays in the middle of a mat such as in [37]; while for mostly footstep-only activities, a mat is an excessive area for the users' footsteps such as in [60]. Or a wearable sensor patch placed on a limb can be considered as partially cover, if it is used to detect the covered muscle activity of the wearer such as in [61]; but if the sensor is used to detect external events, such as emotional touch towards the wearer user (or possibly a toy or robot), it can be considered an exact match or canvas such as the work in [62].

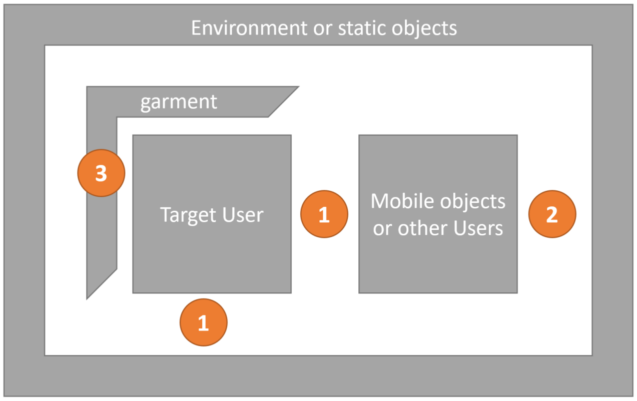


Fig. 16. The propagation dimension of the design space.

Overall *coverage* describes the spatial characteristics of the design. It should also include the granularity of the placement of the sensing nodes. If the sensing nodes are constructed by a matrix, the distance between lines of electrodes (pitch) and the width of the electrodes should be the concerns in this aspect. In some occasions the sensing nodes are individual sensors, the placement and the size of the sensors need to be designed in this dimension. The coverage of a TPM sensing system may not be a rectangular area. In Chapter 8 the virtual surface concept is introduced to cope with arbitrary shape and asymmetric granularity.

2.4.4 Mobility

The TPM sensors can have different mobility installation methods.

1) The sensor is completely *static* during operation. Normally such sensors are installed on the surfaces of the environment or static objects such as the floor, a static furniture piece, etc.
2) The sensor is *movable*, but within a restricted space. Usually they are tethered with a cable to a computing device (including both computers via standard USB or Android smartphones via USB-OTG) either due to high data bandwidth requirement or power supply. Such scenarios can be on a chair surface, or under a mouse pad. Although movable installations can also be wireless, but the application does not require the sensors to be frequently moved from a limited space.
3) The sensor is *mobile*, with wireless communication and battery power. Most wearable or smart objects require mobile level mobility, so that the users can carry them around.

Sometimes, a system may only have static and movable installations, due to the necessity of wired connection, if the sensor has large amount of sensing nodes which exceeds wireless transmission's bandwidth. Or the application does not require the system to be wireless, static or movable installation is sufficient. It is apparent that whether the TPM sensor system can be implemented in a wireless configuration with the current popular standards is a major threshold to divide the overall design space.

2.4.5 Sampling Rate

In activity recognition, the requirement of sampling rates varies based on the application's specific demands. For example, sports applications need higher sampling rates to capture the fast movements, such as left $<$ 20Hz for body motion or left $<$ 100Hz for the impact of a soccer ball; while applications such as posture recognition only need slower sampling rates such as 10Hz or less. These numbers are not hard defined, and shall be determined empirically based on the actual requirement of the application. Video is a proper analogue of this case: in regular Internet video websites, the standard frame rate is around 25Hz (or frames per second). With this frame rate, most activities are visually distinguishable. However, for sports videos such as a person kicking a soccer ball, only a few frames can be captured during the kicking moment. Thus high frame rate is needed, such as 120Hz or higher, so that the action details can be distinguished from the slow motion video.

As will be introduced later in Chapter 7, with the current popular wireless standard Bluetooth serial profile, an embedded system can stream to the host computer or smartphone with up to 30KB/s bandwidth. This becomes the design envelop of balancing between the number of sensing nodes and the sampling rate for mobile mobility systems. While with a wired data connection, the bandwidth is up to 30MB/s, which is virtually not a bottleneck.

In the software aspect, the sampling rate can also be restricted by many factors. For logging data, the writing speed of the storage media, and whether it occupies the processor during the writing cycle can both limit the sampling rate. If the data needs to be processed and visualized per frame, the computational capability of the processor and the efficiency of the implemented programming language may not be sufficient to process all the frames. This problem can even be present when the data rate is the same, but the hardware is generating a higher frame rate with a smaller matrix. If those limiting factors are challenged with the sampling rate, the typical result is data loss. More details on how this topic is discussed in Chapter 6.

Thus while designing a sensor matrix, the sampling rate should be considered in two aspects: what range of the sampling rate is needed and what sampling rate can be achieved, considering other design space dimensions, especially the mobility and coverage dimensions. For example, if in the mobility dimension, wireless data transmission is a hard requirement, then the data bandwidth may be a bottleneck that limits combinations of relatively large sensing node counts and a fast sampling rate. Then compromises need to be found between the sampling and coverage dimensions, such as the necessity of certain sampling rate or sensing node density.

2.5 Three Design Subspaces

These dimensions above can be used to define and guide the design specifications of TPM sensing in different use cases prior to implementing the hardware and software of a sensing system. The five dimensions can be categorized into two groups:

- Core dimensions or abstract dimensions are *Propagation* and *Intention*. Because they define the functionality of the system.
- Practical dimensions or physical dimensions are the remaining *Coverage*, *Mobility* and *Sampling* dimensions.

The general approach is:

- First specify the design space according to the five dimensions.
- Then design initial prototypes to cover the core dimensions: propagation and intention.
- Validate the idea of the system with the prototypes via empirical studies.
- Improve the hardware and software to cover the remaining physical dimensions.
- More thorough validation and user studies with the improved system.

The last two steps can well be iterative for further improvement.

While there are multiple dimensions as mentioned above, based on the use cases, as introduced in Chapter 1, the design space can be divided into three subspaces: ambient, object and wearable.

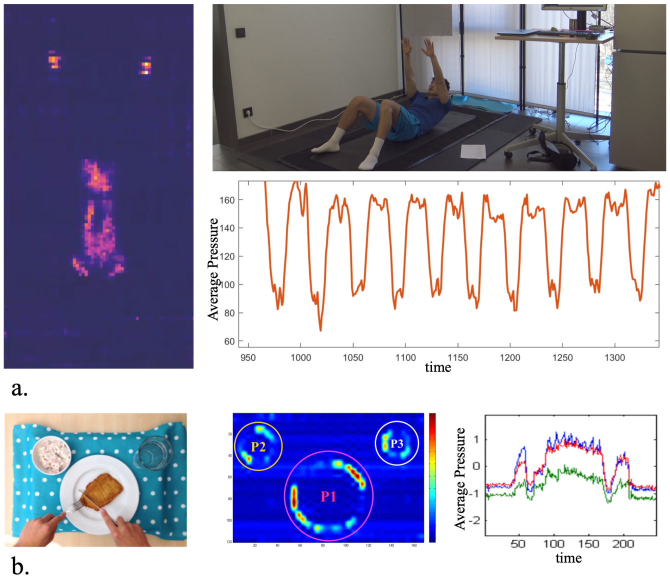


Fig. 17. Ambient application examples: a. smart mat - a person is doing abdominal crunches; the left picture shows the pressure mapping at the moment of the action in the top right, with feet facing upwards of the pressure map; bottom right is the change of the average value of the whole mapping over time during this set of repetitions. b. smart table cloth - left shows a person pocking a steak with a fork, then cutting a part off, and take the part away; middle is the pressure mapping at a moment of this process; and right is the change of average value, and center of mass coordinate of the plate's pressure profile during this process.

2.5.1 Ambient Subspace

Ambient applications are usually immobile surfaces or large pieces of furniture in the environment of people's daily activities, such as the floor or table. Due to gravity, people are eventually in contact with a supporting surface. Also, all the atomic activities will propagate onto the supporting surface even with small details. Take a person seated on a four-legged chair for example, when they are nodding; the head motion can be detected by the force between the chair legs and the floor. [57] This work evaluates TPM sensing in the ambient subspace through the following empirical studies:

Smart mat is a soft sports mat that detects sports activities based on the pressure 'body print' from the users. A variance of the sport mat in the form of a carpet is also useful for everyday pervasive tasks such as gait analysis.

Smart Table cloth is a tabletop cover that detects users' hand motions while dining that is propagated through cutlery and plates to the table surface.

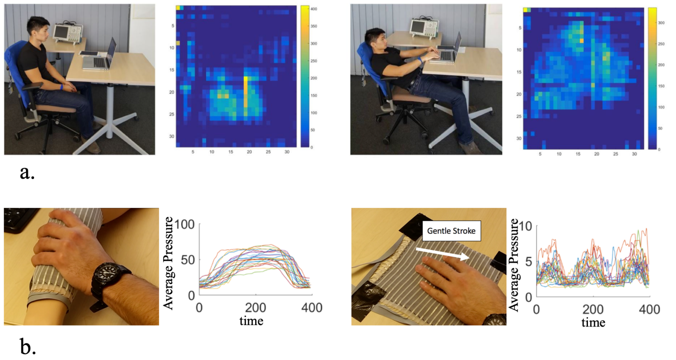


Fig. 18. Object application examples: a. Seat Cover - left two pictures show a person sitting upright with a slouch posture and the subsequent pressure mapping on the back of the chair; right two pictures show the person lying very flat on the chair and the back pressure mapping. b. Robot Skin - left two pictures show a person is grabbing a dummy arm and the temporal change of the average pressure of many of such repeated actions; right two pictures show a person stroking gently on a surface.

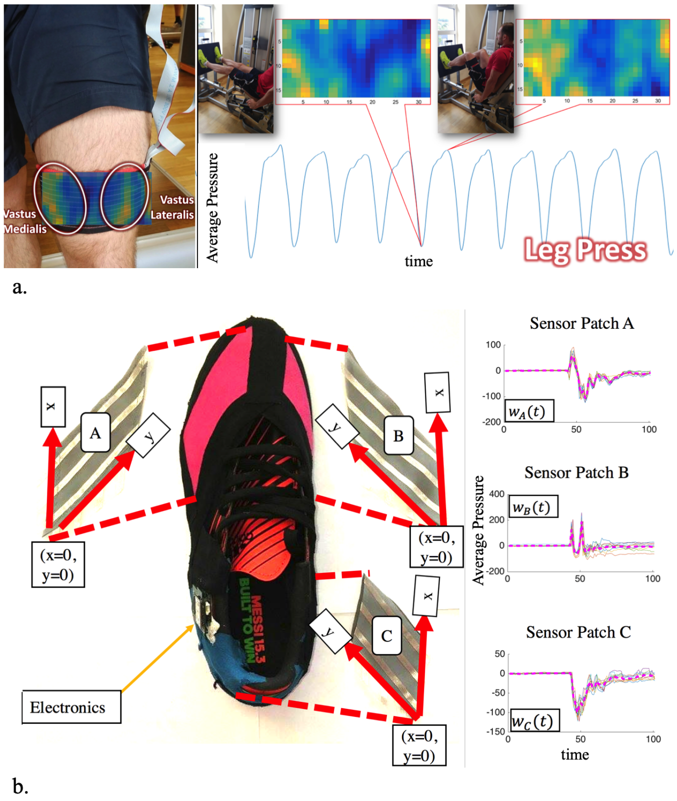


Fig. 19. Wearable application examples: a. Leg Band - left shows the placement of the pressure sensor on a person's thigh, it is fitted with a stretchable band; right shows the person doing leg press, the change of the average pressure values and the pressure mapping at the moments of the minimum and maximum average pressure. b. Soccer Shoe - the left shows a show with pressure sensors installed at the upper face; the right shows the change of average pressure on every patch for a certain type of ball kick.

2.5.2 Object Subspace

Object applications typically look at the force from people's body (hand, body) to small or medium-sized objects (such as a plush toy or a chair). They are usually mobile or movable in the *mobility* dimension. Through the distinct shapes of people's body print and the force signatures, various activities or people's identity can be recognized.

Seat Cover is a furniture cover with TPM under conventional fabrics. It is placed at the back of a chair to detect seating postures. The empirical study showcases explicitly how the software environment is built to give real-time feedback from classification results.

Robot Skin is a TPM sensing patch that is meant to be the skin or clothing of robots with which people would interact. This application evaluates how emotion-related touch gestures can be detected for human-robot interaction studies.

2.5.3 Wearable Subspace

Wearable applications are mainly based on the contact between the body and the fabrics. When people are moving, the skin is essentially pressing against the covering clothes. Therefore, the pressure on the clothing fabric can be related to people's activities. The TPM can be placed at either the pressed surface when people are interacting with the outside world (e.g., at the palm side of the glove, shoe front surface) or pressed onto the body with a compression garment that applies initial pressure.

Leg Band and related *Trainwear* shirt utilizes the idea that the pressure on a fitting wearable garment from our body is related to the body and muscle activities. TPM can be used to perform planar pressure mechanomyography.

Soccer Shoe elaborates the advantages of the versatile tailoring, fast sampling speed and wireless data transmission to cover the top layer of a soccer shoe to measure the impact of kicking a soccer ball, which happens between 10 to 20 ms.

The subspace separations appear to be similar to the mobility dimension, but the subspaces are a combination of all the design space dimensions: the wearable subspace most likely are mobile devices, but mobile devices can also be in the object or ambient domain, such as beacons. Alternatively, a design for the wearable subspace can be movable in the mobility dimension during the initial prototype stage, and develop to a fully mobile device in later refining stages. The design space dimensions and subspaces are thus general guidelines based on existing studies. The aim of this work is to offer a universal starting platform for elaborating existing researches or exploring novel applications with TPM technology.

Chapter 3 Ambient Subspace Empirical Studies

Smart environment systems typically use sensors integrated into the physical world to detect users' specific activities to adapt to the activity contexts, gather information about the environment usage for infrastructure management, etc.

There are a wide range of ambient sensing categories including (1) ranged sensors, such as cameras, infrared arrays, proximity sensors. (2) contact sensors, such as RFIDs, capacitive floor mats, individual FSR sensors, the TPM sensing systems in the ambient space also fall in this category. (3) physical parameter sensors, such as smart utility meters, ambient light sensors or atmosphere sensors.

Ranged ambient sensor and physical parameter sensor systems typically cover bigger space in the physical environment with single points of sensors. These ranged sensors can also be installed at the ceiling. These two features give them better independence and unobtrusiveness. However, the ranged sensor systems suffer from several problems. The obstruction of the target users or activities can result in invalid detection outputs, or undermine tracking processes. Complex background and especially for cameras changing lighting also challenge object detection and filtering algorithms as well as computational cost.

Contact sensor systems compliment the disadvantages of the ranged sensor systems at obstruction and complex background problems. They usually are more specific in the activity detection tasks they carry. For example, RFID tags are usually only able to detect the usage or presence of the user at specific objects and locations; while ranged systems such as a computer vision system, can detect various activities with a single camera input.

TPM sensor systems physically are in the contact sensor systems' scope. However, the information they offer is more similar to some ranged computer vision sensors since the TPM sensors offer pressure imagery. Typically the data TPM offers can be processed with computer vision techniques. Moreover, since the contact pressure is close to zero at those sensing nodes where there are no other objects, TPM imagery has the unique advantage that the background is usually clean of any complexity, because where there is no object or user, there is no external pressure. Also, there are usually no irregular textures in TPM imagery comparing to real camera images, such as texture patterns on clothing. As a result, adapting computer vision techniques in TPM imagery typically deal with the contour, shape and simple profile. As computer vision techniques are developed to process much more complex images, they have proven to be quite effective with the much cleaner and simpler TPM imagery. Artificial neural networks trained with images can also be used to extract features and classify TPM imagery in the form of transfer learning.

From the five design dimensions, ambient space normally spans over *direct* and *indirect* in the *propagation* dimension. Ambient space can cover the whole *Intention* and *Coverage* dimensions. In the *Mobility* dimension, ambient space stays mainly in the unmovable part; while some designs can be movable or mobile, with relatively smaller sensor patches attached to movable parts of the environment such as a door or dining tray. The *Sampling* dimension of the ambient space is focused on the medium and static range.

This chapter discusses three empirical studies:

- Smart Table Cloth: this study evaluates the idea that activity can be detected indirectly in the *Propagation* dimension, from the source of the action which is users' hand holding cutlery, propagated through the plate onto the tabletop.
- Sports Mat: in this study, a TPM matrix is implemented inside a sports mat to discover differences in the dynamic body print from various exercise activities.
- ID Carpet: a TPM matrix similar to that in the *Sport Mat* study is implemented under a carpet to detect a person's identity from the dynamic pressure change of single footprints.

3.1 Table cloth

TPM sensors do not necessarily need to be directly placed with the source of activity, which is the *indirect* method in the *Propagation* dimension of the design space. In particular, the force propagated from the activity source, through the object, to the supporting structure would represent distinct patterns of the activity source. To study such a method, a dining scenario is designed: a TPM sensor takes the form of a piece of table cloth, placed under dining plates, to detect and distinguish different micro-actions of acquiring food.

From the design space point of view, the core dimensions *Propagation* and *Intention* of the system should be indirect and unintended/unaware. In the physical dimensions, in *Coverage* it should be excessive so that the surface can enclose the dining containers. In *Mobility*, since a table cloth is typically fixed at the dining area; therefore, unmovable or movable is sufficient. Yet it is possible that a dining tablet needs to be carried around a bigger area such as dining halls or hospitals. Therefore mobile can also be a possible range in the *Mobility* dimension. In *Sampling*, the actions of cutting and acquiring food are typically moderate speed activities; therefore the medium sampling rate is sufficient.

This study includes 10 participants, each having consumed a total of 8 meals chosen from 17 possible main dishes (divided into four classes according to the predominant cutlery action involved) combined with 6 possible side dishes (divided into 3 classes). [1]

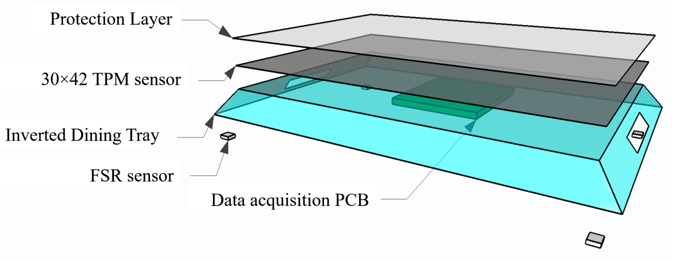


Fig. 20. Table cloth study: prototype hardware illustration.

3.1.1 Empirical Case Introduction

A wide range of global medical studies have shown that dietary habits are related to risks of various health problem from digestive problems, obesity [65], diabetes [66] [67] to coronary heart disease (CHD) [68] and even gastric cancer [69]. Food intake patterns are also an important issue in elderly care, in healthy children development and in general physical and mental well-being. In summary, it can be said that nutrition habits are at least as important for health and well being as is physical exercise. As a consequence, in the past decade, much research in pervasive computing has been devoted to using various sensor systems to automatically monitor eating habits to improve the traditional methods for dietary tracking such as self-recalls, surveys, caretakers, etc. Recently, first consumer products such as sensor-equipped cups, forks, in-ear microphones have also emerged. However, while automated monitoring of physical exercise has become commonplace, automatic nutrition monitoring has so far had very little impact on real-life applications. Instead, state of the art in electronic nutrition monitoring support is smartphone apps that help users note down their daily dietary intake. Such apps make self-reporting easier; however, they cannot fully eliminate problems such as compliance and reliability that are inherent in manual reporting.

The main challenge of automatic dietary monitoring is finding sensors that are (1) unobtrusive, cheap and suitable for long term every day use while (2) providing sufficient information on nutrition related behaviour. An overview nutrition monitoring approaches can be found in [70]. Examples of approaches that were investigated to date range from spotting nutrition related gestures with motion sensors [71] through detecting chewing/swallowing with in-ear or neck-worn microphones [72] [73], textile capacitive collar sensors [74], and neck mounted EMG electrodes [75], to instrumented cutlery [76] and camera based food analysis [77]. On the other hand, ambient sensing mostly focus on event level activity recognition (e.g. eating schedule) by tracking food items being placed/removed in the fridge and cupboards [78] [79] [80].

For big meals such as lunch, dinner, or elaborate breakfasts, food intake is primarily determined by cutlery use. This includes not only the actual intake (putting food into the mouth), but also acquisition operations such as cutting, poking, and scooping. This work is motivated by the observation that such operations and the way they are performed are closely related to the type of food being consumed. Thus, food such as steak requires cutting before eating. Chopped meat, such as goulash, might require only “poking” with a fork. Rice, mashed potatoes, and so on are “scooped”, while noodles are often swirled onto the fork. If such different actions can be recognized, they can provide information that can be used to augment other sensing modalities towards more detailed dietary monitoring. Further information, which we obtain by detecting the plate on which an operation has been performed, can reveal how much main dish and how much side dish has been consumed. Finally, the frequency and number of cutlery actions are obviously related to eating speed and the amount of consumed food. These considerations have led to the development of “smart cutlery”. Here, we investigate the value of a different sensing modality that can be used either instead of or in addition to such augmented cutlery: pressure-sensitive eating surfaces that can detect and analyze subtle pressure signals caused by the application of cutlery on the plate. The two sensing modalities we’ve selected are largely unobtrusive. Our textile pressure-sensing matrix is covered by a water-resistant layer topped by a traditional tablecloth (that can be washed or exchanged); as such, it can be permanently kept on a family table with no relevant disturbing effect. Furthermore, our simple FSRs can be installed under any firm surface, such as a dining tray or personal eating surface.

[1] This section is based on the publications [63] and [64]. As the leading author of the publications, I have only included my own work; unless other authors’ contribution is an integral part of explaining my work, in which case it will be explicitly stated. Since the published work has gone through extensive wording, some sentences and passages have been quoted verbatim from the said sources.

3.1.2 Hardware Requirement

A simple meal tray is used as the base structure (see Figure 1). The tray is inverted so that the space below its rim can be used for housing the wirings and electronics. As Fig. 20 shows, a layer of the TPM fabric is placed on top of the tray as a smart tablecloth. The force sensitive resistors (FSRs) are placed underneath the four corners. Custom spacers are added to focus all the weight from the tray platform into the sensitive area of the FSRs.

The electronics is hardware version F-2 from Table 17 in Chapter 2. The matrix size is 30-by-42 with a pitch of 1 cm. Every pixel of the TPM is scanned at 24-bit and 40Hz. The FSRs are scanned at 5120Hz, but every 128 samples are calculated as average to strongly suppress noise and match the TPM's sampling rate of 40Hz. Instead of soldering wires right at the edge of the sensing area, the metallic stripe layer of fabric is used to extend the connection so that it can be folded underneath the platform. Therefore, the users only feel a continuous, fabric-only surface. This also highlights how a textile integration technique can make the smart fabrics more unobtrusive.

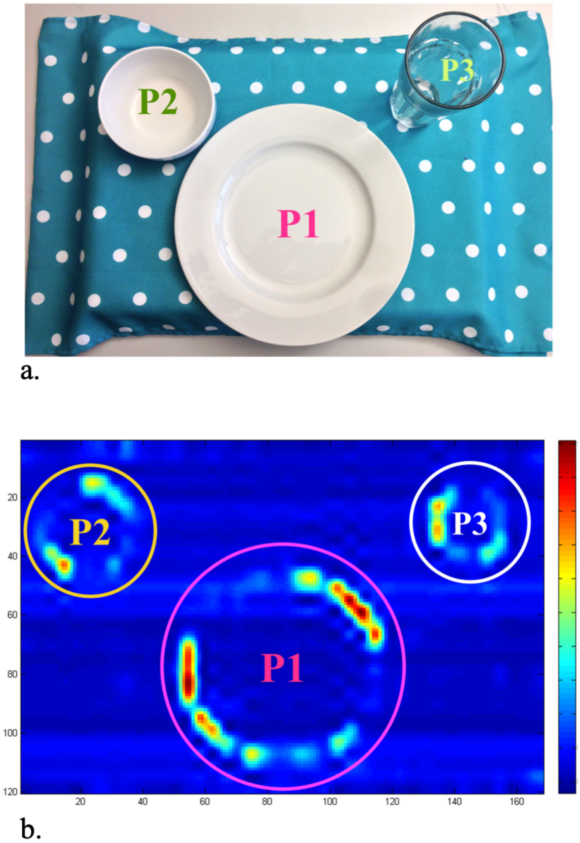


Fig. 21. a. object placement, b. frame and object detection after signal conditioning.

3.1.3 Experiment

For data collection, we conducted an experiment with a group of 10 users: seven males and three females who ranged in age from 22 to 27 years old. Group members included vegetarians and people who were left-handed, and members came from various cultural backgrounds across Eurasia. Our primary goal was to distinguish between different actions as listed in Table 1. During the study, each volunteer chose two dishes from each class in the provided menu (see Table 1). Overall, volunteers consumed eight meals (lunches) over a two-week period. Soup or salad was chosen to accompany the main dishes in equal numbers, but only one of the two was served as a side dish in each meal. The order of meals was decided based on the convenience of the cook, availability of the persons each day, and overall scheduling issues.

To simulate a canteen/cafeteria/restaurant environment, shift the volunteers' attention away from their eating actions, and introduce realistic noise, the volunteers were encouraged to watch TV and socialize with other people during the experiment. Also, we arranged to have people walk around in the same room because a pressure sensor together with a mass constructs a simple accelerometer, which can pick up vibrations from the floor. The volunteers were not restricted to use specific tools for specific actions. Some people, for example, used a spoon for scooping risotto/rice or gathering salad, while others used a fork; and some used only a fork for stirring noodles, while others used a knife or spoon to assist the fork. Although the sensor matrix is capable of tracking multiple objects, it was not the focus of this case study, so the volunteers were not allowed to rearrange the plates. The containers were weighed before and after the meal to determine the consumption ground truth.

3.1.4 Dataset and Features

The data analysis schemes include the following steps: processing the raw data from the sensors, extracting features, verifying the sensing modalities by cross-validation from labelled data, and gesture spotting from data recordings without labels.

The raw data stream of the matrix *(Source 1)* is a standard TPM *Stream* introduced in Chapter 8, while the raw data of the four FSRs *(Source 2)* is four-channel. Every frame in Source 1 is up-scaled four times on each dimension to 120 × 168 using bicubic interpolation. The background (where there is no object placed) is filtered out using a dynamic threshold, calculated by the histogram of the 2D frame. Because the focus is not dynamic tracking, the object detection and segmentation part of data processing is simplified. The 2D frames within a second are accumulated per pixel basis, then Atherton and Kerbyson's phase coding method [81] is used to locate the three objects (plate, bowl and glass). Because in this experiment the food containers' placement is not altered, the majority of the positions and radii of the 3 objects are used as the static location. The original data stream is then segmented into streams of 3 subframes (P1, P2, and P3), which is the pressure mapping changes over time of the 3 containers. An example of the signal conditioning result is shown in Figure 2. The four FSRs are mean-filtered and the data stream named as R4.

Because the food containers are rigid objects, the overall shape of each pressure mapping is stable. However, the strength and focus of the force changes as people use cutlery to handle the food. Therefore, for every subframe, we calculate the spatial average of pixels (weight) and the weighted center on each dimension (center(x, y)). Then,

TABLE 1
Table Cloth Study: Action Classes and Menu Options

Class Index	Action Class	Menu Options
Main Plate (P1)		
C1	Stir	Spaghetti with meat sauce, Asian noodles, instant noodles, spaghetti with vegetarian sauce
C2	Scoop	Chilli con carne with rice, salmon with rice, chao fan (fried rice), risotto
C3	Cut	Schnitzel, steak, burger steak, vegetarian soya schnitzel, vegetarian soya steak
C4	Poke	Chicken nuggets, curry wurst, mini-frikadeller, vegetarian soya nuggets
Side Plate (P2)		
C5	Scoop(liquid)	Chunky vegetable soup, creamy soup
C6	Collect	Garden salad (green leaves), bean salad, wurst salad
Glass (P3)		
C7	Remove/Replace	Any drink

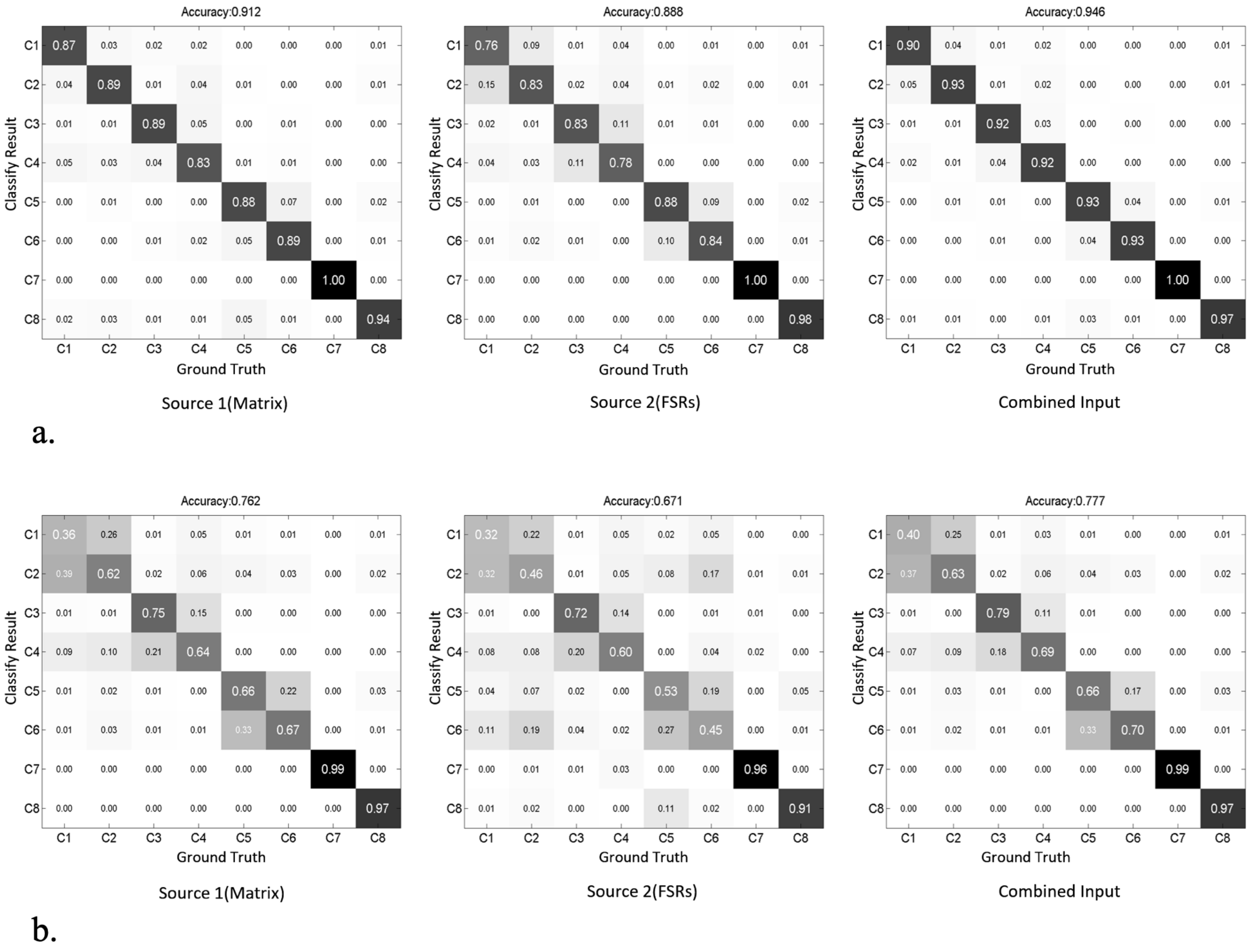


Fig. 22. Table cloth study: confusion matrix of table cloth and four FSR sensors.

within every time window containing an action event (segmented from the ground truth), the sequences of weights (w(t)) and centers (c(t)) are normalized according to the mean and standard deviation values. Next, we extract 16 overall features that describe the signal's statistic and dynamic properties, calculating the mean values of w(t) and c(t) before normalizing: The features used during the study are:

- mean of $w(t)$
- mean of $c(t)$
- std of $w(t)$
- std of $c(t)$
- max and min of $w(t)$
- range of $w(t)$
- max and min of $w'(t)$
- mean of $\mid w'(t) \mid$
- max and min of $w''(t)$
- mean of $\mid w''(t) \mid$
- central frequency of the FFT spectrum of $w(t)$

For the four FSRs, information containing only individual sensors should not be used since they are biased by specific container placement. Therefore, the 4 channels of

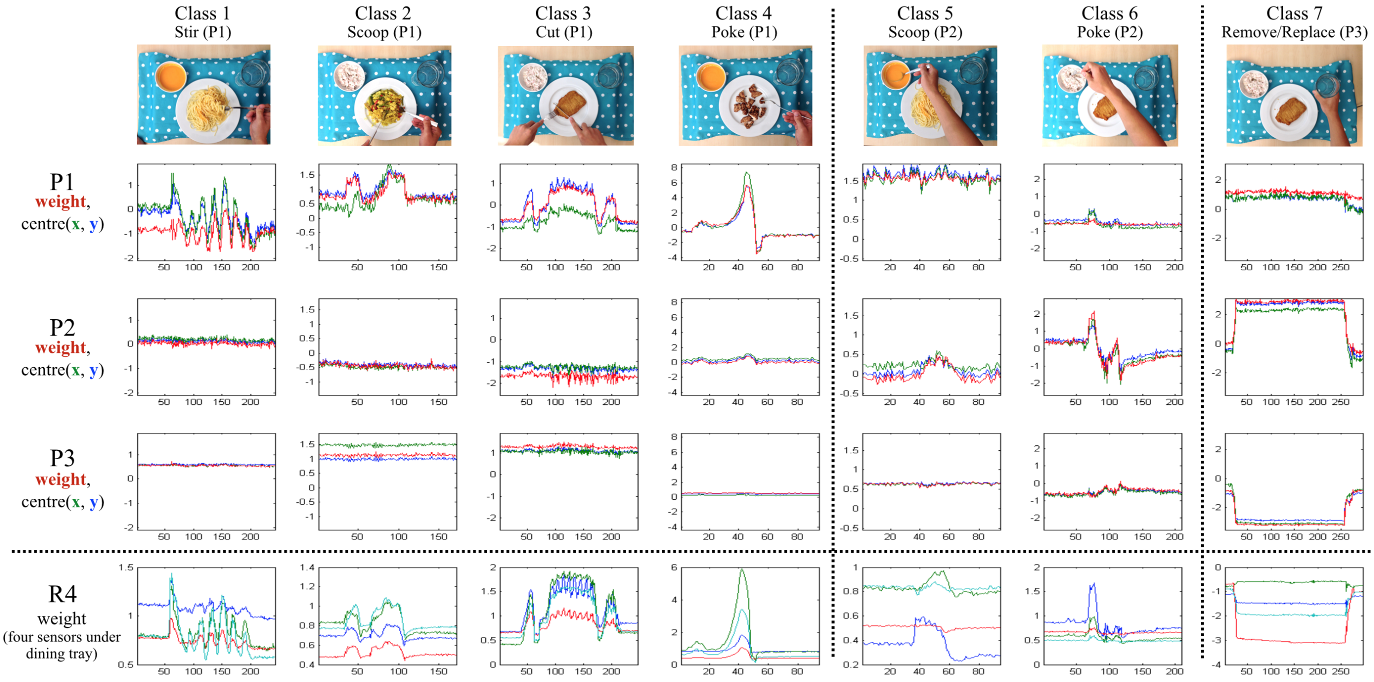


Fig. 23. Table cloth study: signal example of the table cloth and four FSR sensors.

data are converted into a sequence of the average of four channels ($f(t)$). Then the following set of 12 features is chosen for each event:

- mean, std, max, min and range of $f(t)$
- max and min of $f'(t)$
- mean of $|\ f'(t)\ |$
- max and min of $f''(t)$
- mean of $|\ f''(t)\ |$
- central frequency of the FFT spectrum of $f(t)$

For Source 1 (TPM), the classification process is 2-staged: firstly a KNN classifier takes only the range and std of $w(t)$ and $c(t)$ of all three objects to decide on which plate the action is happening. Thus the non-action class (C8) and the objects where only one possible action is defined (in this case, remove then replace the glass - C7); classes of the same objects are also grouped (C1-C4 and C5-C6). Then for each object, a separate confidence-based AdaBoost algorithm (ConfAdaBoost.M1) with decision trees as the base classifier is used. Each classifier takes only the 16 features and is trained with only the respective classes from the object. For Source 2 (FSRs), again the same ConfAdaBoost.M1 algorithm is used. Since it is also interesting to investigate how the two sources can compensate for each other, a sensor fusion process is done by using the matrix to decide the object; then separate ConfAdaBoost.M1 based classifiers take corresponding features from the matrix and FSRs.

The spotting approach is based on the observation that, in general, the only changes in the weight distribution on the plate come from the handling of food. Thus all we need to do is to analyze the dynamic change in the data stream (in the pressure matrix case, the data stream is summed weight w(t) and weighted center c(t) of each, and in the four FSRs case, the average of the four channels f(t)). For actions when the person is acquiring food from the plates, an adapted threshold is applied to the standard deviation of the data stream. This results in a binary mask along the time domain, where logic '1' marks a possible action. For actions such as removing/replacing the object (e.g., drinking with the glass), only the transient state of the removal and replacement can be registered by the previous method, and the state when the object is not on difference between data stream and the corresponding polynomial curve fitting at every sample point. The use of polynomial curve fitting instead of linear approximation is due to the fact that the overall data stream of the complete meal is a decreasing curve instead of having a stable average value, which is caused by the decreasing of food and the sensor material's settling effect.

Through combining the two masks, removing small instances (less than half a second) and grouping adjacent instances (less than one second), events can be spotted. The events are then classified by the same method mentioned above.

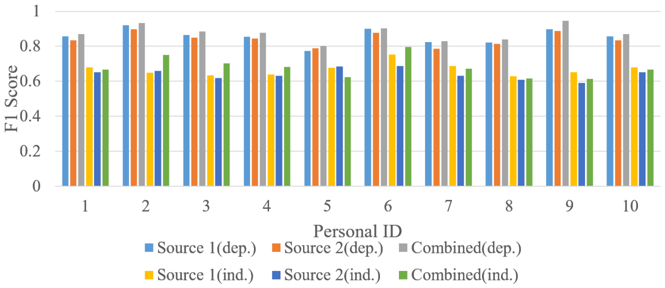


Fig. 24. Table Cloth Study: average Spotting F1-score from every participants with different input source(es) in person-dependent/independent cases.

3.1.5 Classifying Action Types

The first evaluation step is the classification of individual cutlery actions (e.g., single poke, single cut, etc.) on pre-segmented data. Note that the data also includes segments from the NULL class, which encompasses the setting of the table and times when the user applies the cutlery without actually taking food (such as placing them on a plate without eating). In the person-dependent case, ten-fold cross-validation with balanced training was performed. For the person-independent case, leave-one-participant-out cross-validation was conducted.

Fig. 22 shows the average results of both person-dependent and -independent cases as confusion matrices, and the detailed spotting results of each person are displayed in Fig. 24 as F1-scores. As the figure shows, on the person-dependent case, approximately 90% of individual cutlery actions are correctly recognized by all sensing modalities, with the combined classification going up to 94% and the FSR-only case being just below 90%. It is also important to note that the NULL class recognition goes up to 97 percent. The primary source of confusion came (as expected) from C1 and C2, and C3 and C4. This is possibly because the food content of each pair has a similar texture and thus requires similar force to acquire (risotto and noodles are both soft and scattered, while steak and nuggets are more rigid and have distinguishable pieces).

The accuracy of the person-independent case is significantly lower; the best result achieved by combining both sensing modalities is 77%. Given the individual variance of eating styles, the diversity of our participants and menu selections, and the small sample size, this is not surprising. During the study, diversities such as people using different cutlery for the same action are observed(both forks and spoons were used to scoop risotto, for example). Also, participants have different uses of the same fork for the same dish, as when some participants stirred noodles with their fork at a flat angle, while others stirred with the fork almost horizontal. In addition, some participants first did a scoop-like action to locate a big bulk of noodle threads, and then started slowly flipping the fork, while others merely pierced the fork into the food and the major action was to swiftly rotate the fork. In such cases, the former could be classified as scooping (C2).

The average accuracy values for different containers in the combined modality case were 92%, 93%, and 100% for P1 (C1-C4), P2 (C5-C6), and P3 (C7), respectively, for the person-dependent case; and 63%, 68%, and 99% for the same containers, respectively, in the person-independent case. The localized accuracy seems to decrease as more action classes are defined.

3.1.6 Spotting Results

As Fig. 25 shows, the classifier is trained by the other meals of the same participant, and leave-one-meal-out validation is performed. The classified instances are then compared with the ground truth. A match is defined by a significant ($> 50\%$) overlapping between the classified instance and the ground truth in the time domain, with the two labels having the same class. Thus, precision and recall are calculated, and the F1-score is used to evaluate the spotting result.

For the person-dependent cases, the F1-scores range from 0.8 to 0.9; for the user-independent cases, they range from 0.6 to 0.8; and the combined system mostly scored the best by a few percentage points. Overall, this is consistent with the pre-segmented classification results.

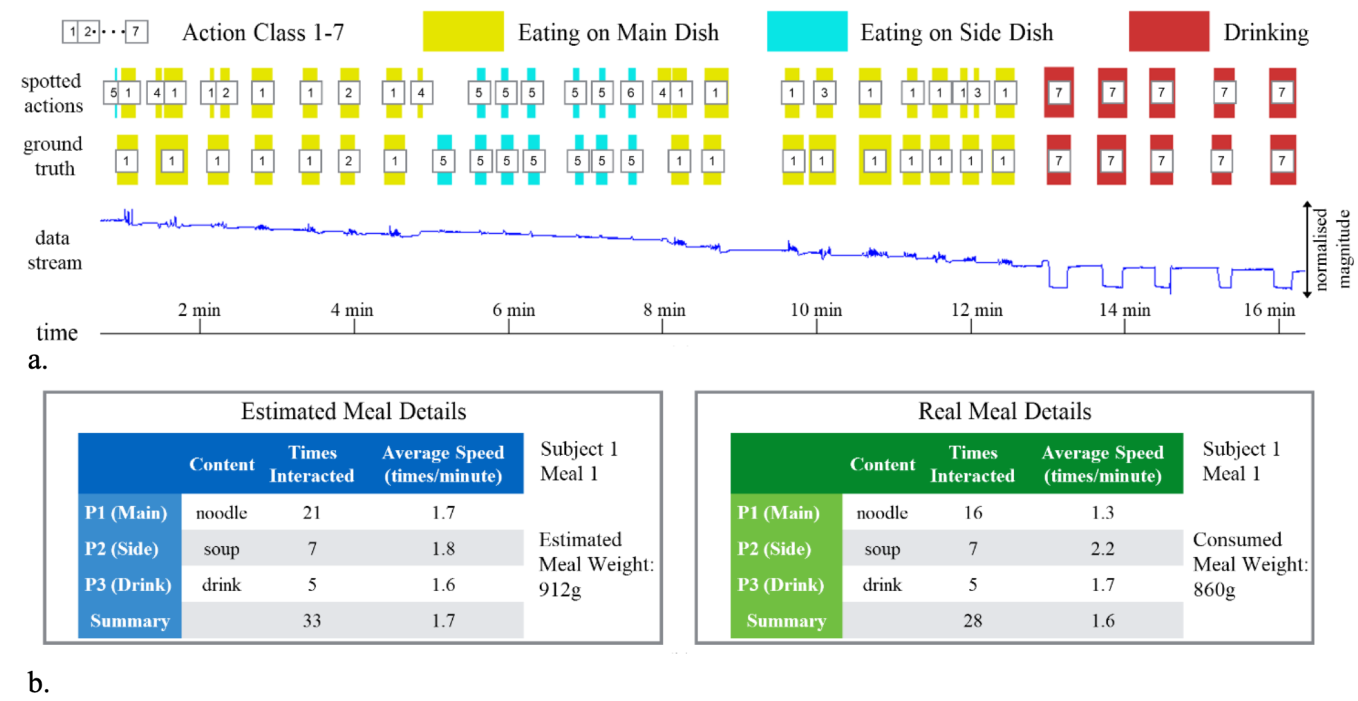
3.1.7 Estimating Food Content and Weight

The motivation for tracking and recognizing cutlery actions lies in their relevance to various aspects of food monitoring. The spotting results indicate that the estimation of food intake speed is possible at least at a qualitative level. The next aspect is the detection of food type. As described, cutlery usage is suitable to distinguish between certain broad categories related to different eating techniques. Essentially, it is the assumption that consuming a meal belonging to a particular category implies that the majority of the cutlery actions performed will fall within the specific class (for example, when eating a rice meal, the majority of actions will be scooping).

Majority analysis is applied to the recognition data. In the user-dependent case, 78 out of the 80 consumed main dishes and 74 out of the 80 side dishes were classified correctly. In the user-independent case, 67 out of the 80 main dishes, and 68 out of the 80 sides were correct. Obviously, such performance is based on fixed menu categories in a study; however, this core idea can help shrink the number of possible food types in real-life applications.

The final high-level analysis aspect is food weight and volume estimation. Obviously, high-precision weight sensors exist that could be placed in the tray. However, the question is how well the estimation As mentioned in Chapter 2, compared to rigid force sensors and weight scales, flexible sensors are more sensitive to dynamic transient changes, but less likely to return to the exact state with the same weight, yielding poor repeatability. We compared the food weight difference and the difference of the four FSRs' average output before and after a meal. As shown in Fig. 26a, between the weight range of 0.3 and 1.2 kilograms, the root mean square of the variation takes approximately 16 percent of the range. This, however, can be further improved by stacking rigid scale sensors with the FSRs.

Another possibility for estimating the amount of eaten food is to count the individual cutlery operations. The general idea is that, at least statistically, people take similarly sized bites. The advantage of such an approach is that it would allow the system to break down the amount into different classes of food as given by the different types of food acquisition actions and different plates (main dish, side dish) from which they were taken. To see how feasible this approach is, the ground truth of bite times and food weight differences are summarized to derive the average weight-per-bite value of each meal in Fig. 26b. As the figure shows, although the average bite size varied among participants (for example, the female participants usually had a smaller bite size), for a fixed participant, the fluctuation range was limited. By comparing the main dish (four categories of meals) and the side dish (2 categories), the range of variation in average bite size tended to increase as the food diversified.



Estimated Meal Details

	Content	Times Interacted	Average Speed (times/minute)
P1 (Main)	noodle	21	1.7
P2 (Side)	soup	7	1.8
P3 (Drink)	drink	5	1.6
Summary		33	1.7

Subject 1 Meal 1

Estimated Meal Weight: 912g

Real Meal Details

	Content	Times Interacted	Average Speed (times/minute)
P1 (Main)	noodle	16	1.3
P2 (Side)	soup	7	2.2
P3 (Drink)	drink	5	1.7
Summary		28	1.6

Subject 1 Meal 1

Consumed Meal Weight: 860g

Fig. 25. Table cloth study: dietary details extracted from a meal (a) data (average pressure) and the spotting result (b) comparison of dietary details between the ground truth and machine learning result.

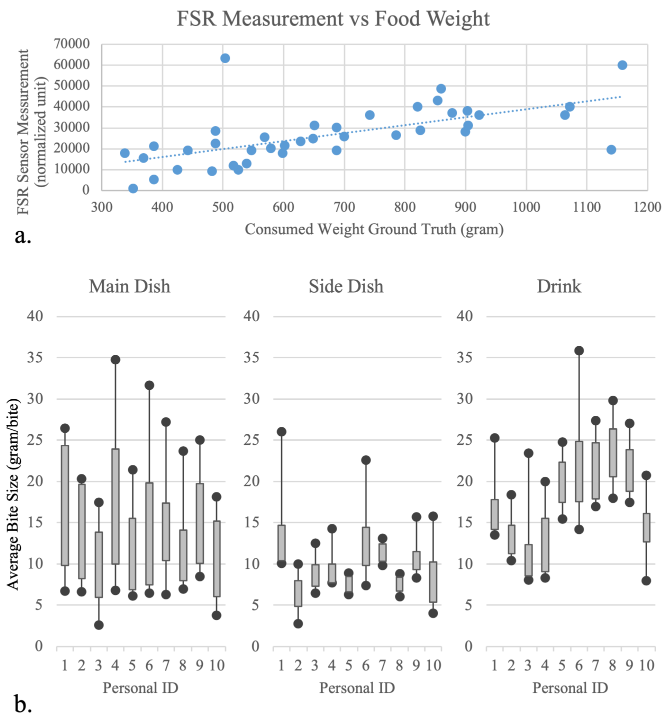


Fig. 26. Table cloth study: the link to food weight approximation from the study result. (a) FSR measurement compared with real consumed weight. (b) average bite size fluctuation of all paritipants (max, min, 75 percentile and 25 percentile of ground truth).

3.1.8 Conclusion

In the user-dependent case, the system is able to correctly identify up to 94% of individual actions. Given the fact that, in general, dietary monitoring is likely to require a statistical analysis of such actions, this can be considered sufficiently accurate for most purposes. Clearly, in user-independent cases — in which individual variations in how cutlery is used obviously exist — the accuracy lowered to approximately 77%. In terms of comparison of the textile pressure matrix with the FSR, it is clear that the TPM's ability to precisely locate dishes outweighs the disadvantage of noisier signal quality. This manifests itself in the nearly 10 percent higher accuracy in the user-dependent recognition. Understanding how the position information can be leveraged in the case of multiple users and more complex dish setups (such as bowls from which food is taken onto personal plates) can be an interesting subject of future research.

The results of this study clearly show that pressure-sensitive surfaces such as TPM table cloth analyzing cutlery-related actions are a valuable source of information for dietary monitoring. As outlined at the beginning, dietary monitoring is inherently a multimodal problem; obviously, as with other information sources, surface pressure sensing is insufficient to cover all daily dietary intake aspects, particularly with snacks and casual drinking. However, with its unique unobtrusiveness regarding the dining experience for users during meals, we believe that surface pressure sensing will become a major contributor for augmented and connected dietary monitoring systems that could include ambient, portable, and wearable devices.

In a grander scope, this study shows that TPM can be used as an unobtrusive activity detection method that is not coupled directly with the activity source, but through objects that the activity source is interacting with.

3.2 Smart-Mat

3.2.1 Introduction

This project is motivated by the observation that many strength related exercises involve contact between different body parts and the ground, resulting as a 'body-print'. For different exercises, a characteristic dynamically changing body-print can be identified. As a soft material based sensor, TPM can be embedded under the surface of exercise mats to measure such body-print changes. This also means the monitoring exercises can be performed without the need to wear any additional sensors. In contrast to fitness trackers and smartwatches that are detecting gym exercises, these IMU based sensors can only accurately detect motion on the attached body part. For example, a smartwatch may be able to detect upper body exercises, but not the swing of the arm which is not wearing the smartwatch, or leg exercises. [2]

We consider the core design space dimensions. Since the body-prints are direct results of the user's activity on the floor, the *propagation* dimension is *direct*. As the goal of exercise activities in this study is not creating those body-prints, the *intention* dimension is *unintended*. For an sports mat, normally the users have a fixed position relationship with the mat while doing exercise, and the mat has a dimension that approximately covers the area of the users while they are lying down. Therefore the *coverage* dimension should be *exact*.

This empirical study is conducted in two phases. In Phase 1, a pilot study with a prototype hardware explores the plausibility and the algorithms to detect various exercises. The hardware in Phase 1 made compromises in the coverage dimension due to the technology limitation at the time. An improved hardware that meets all design space requirements enables Phase 2 of this study. In Phase 2, the TPM enabled sports mat can distinguish even minor variations of the same exercise category.

3.2.2 Phase 1 Study: Prototype Hardware

In the first phase, the prototype uses an early version of the TPM system. The electronic hardware is Version F-2 from Table 17 from Chapter 7. The hardware follows the architecture based on FPGA and parallel ADCs, and achieves a scale of 80 × 80 matrix (6400 sensing points) at the sampling rate of 40 frames per second. The operation is controlled by an Altera EP4CE6 FPGA, another Altera EP3C24Q240 FPGA is used for powering the 80 active electrodes. Three of the four 24-bit ADCs operate in parallel to measure the resistance of the passive electrodes. Every ADC has a 32-to-1 analog multiplexer to route the analog signal into its input. The output of every ADC is sent to the computer via a dedicated virtual COM port (VCP) channel to the data receiving computer. A program implemented in C++ with the Qt environment receives, parses and logs the data for further analysis.

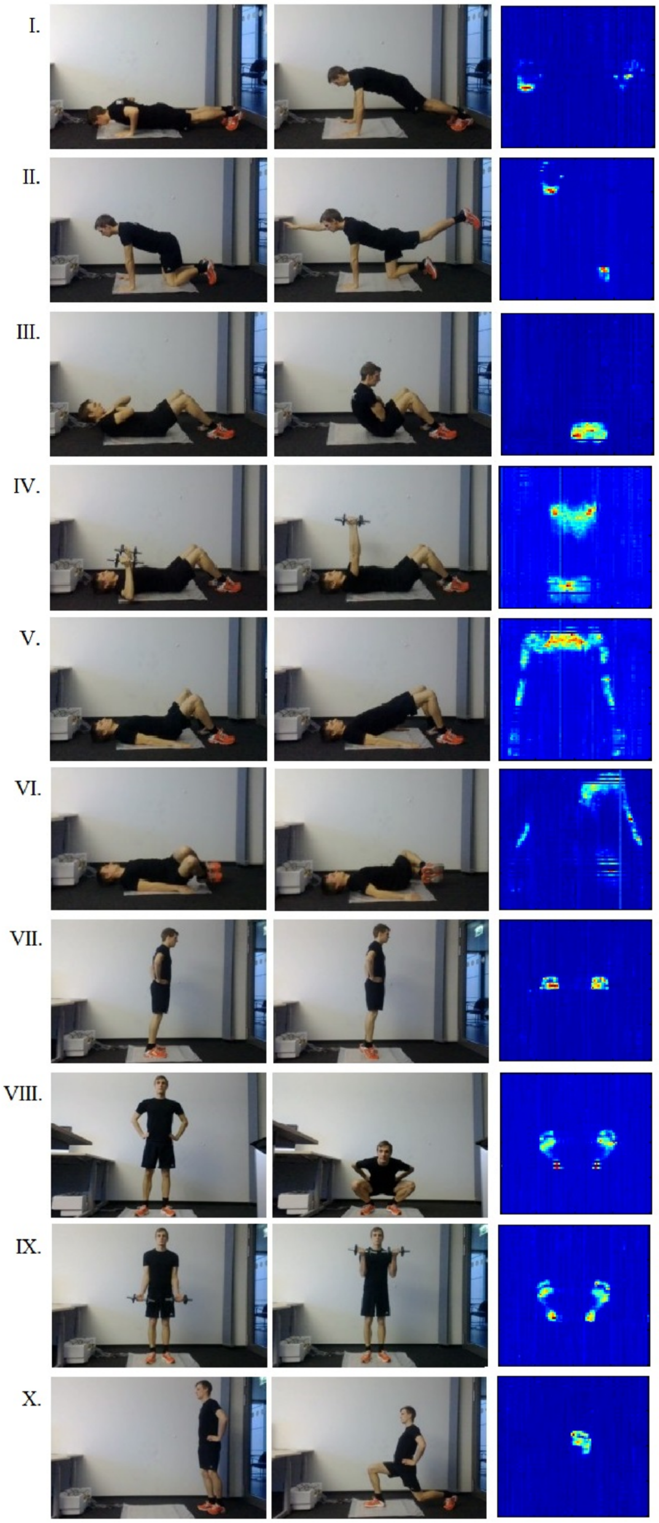


Fig. 27. Exercises performed in the experiment: I.push-up, II.quadruped, III.abdominal crunch, IV.chest press with dumbbell, V. bridge, VI. segmental rotation, VII. calf raise, VIII. squat, IX. biceps curl with dumbbell, X. lunge. The pictures on the right are typical frames corresponding to the exercise.

Details about the hardware are described together with the systems from other studies in Chapter 2. The electrode arrays are hand-tailored laminated aluminium stripes with a pitch of 1cm, since by the time the SimpleSkin universal metallic stripe fabric has not been developed. The middle layer is the CarboTex fabric produced by Sefar.

[2] This section is partially based on the publication [37]. Section 3.2.3 and Section 3.2.4 are brief summaries from Sundholm, Mathias's contribution. The rest of the chapter is my own contribution.

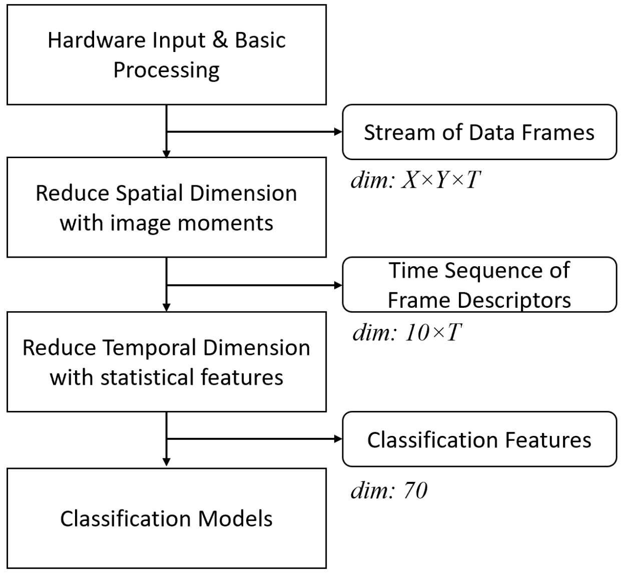

Hardware Input & Basic Processing
Stream of Data Frames
dim: X×Y×T
Reduce Spatial Dimension with image moments
Time Sequence of Frame Descriptors
dim: 10×T
Reduce Temporal Dimension with statistical features
Classification Features
dim: 70
Classification Models

Fig. 28. Smart-mat Phase 1 algorithm flow chart

3.2.3 Identify Exercise Categories

Ten exercise categories are defined as shown in Fig. 27, and performed by 7 participants. These exercises are compiled based on the training commonly used and recommended by fitness trainers. From Fig. 27, every category in general has its specific dynamic body-print which can be already visually distinguishable. Every participant performed 10 exercises for 2 sets × 10 repetitions.

In this study, *feature domain* is defined as temporal sequences of 10 image moments of every frame (3 central moments and Hu's seven moments [82]). It is essentially the *frame descriptor* concept which is described in Chapter 8. Then features for classification are calculated from a time window of the *feature domain* as mean, standard deviation, minimum, maximum, range of the window, and minimum, maximum of the approximate 1st derivative.

10-fold cross-validation is performed with a KNN classifier. For the same person, leave-one-set-out is performed. Every repetition is individually annotated, and taken as a single sample. The confusion matrix is shown in Fig. 29 (a) with 88.7% accuracy. For the person independent case, leave-one-participant-out is performed. Instead of using the annotated repetition, a sliding window is used with 100-sample window size and 10-sample window step within every set. Within each set, it is assumed that only one exercise exists. Thus majority voting decides the class of the set. The resulting accuracy is 86.4% as shown in Fig. 29 (b).

3.2.4 Counting Repetitions

Counting exercise is performed by dynamic time warping. For every feature domain, the template is calculated by the following method:

1) interpolate all the training exercise repetitions to the same length;
2) first a crude template is produced by calculating the average at every point in time of all the training exercises;
3) then the training repetitions are shifted, so that has the least error from the crude template, then padded with border values;
4) every training repetition is then given a weight, which is the inverse value of the error between the weight and the crude template;
5) the final template is then calculated as the weighted average of the training repetitions.

Then a sliding window is used. Every window τ is compared with the template as a 10-dimensional dynamic time warping process. The resulting measure $\mu(\tau)$ is defined as the warping path / warping distance. Thus a peak of the time sequence of $\mu(\tau)$ indicates currently there is a local best match of the template, and thus can be used to determine the central time of the repetition. A threshold and tolerance are combined to find the highest peak when there are several peaks inside a small period.

Every participant generates a group of training templates. A cross-validation of each participant tested with every other participants' templates is carried out. By comparing to the time ground truth, the average counting F1-score is 82.8%. The best template group is selected from the cross-validation process, and the resulting counting F1-score is over or close to 90% in most cases, only 2 exercises from certain participants are below 60% and 3 more below 70%.

3.2.5 Discussion of the First Phase

The result of using the 'smart-mat' for sports exercise recognition has motivated the further progress of the large scale hardware direction. Version F-2.1, F-4 from Table 17 in Chapter 7 are specifically designed for a better integrated smart-mat system. Not only the hardware is more mature and integrated, but the analog part that measures the passive electrodes has also been significantly improved to address multiple problems such as the 'row shadow' problem and broken line problems in the hardware of the paper [37]. The 'row shadow' is a phenomenon that one pressed sensing point will influence all the points connected to the same passive electrode, as explained in Chapter 2.3.1.

The introduction of the metallic stripe fabric by Sefar has also contributed to the next phase. It superior to the hand-tailored aluminum foils in terms of uniformity and durability. Aluminum foils are easy to break after repetitive deformation. Also, the solder joints between the foils and the copper wires connecting to the electronics are easy to break, leading to some of the broken line problems.

In [37] the data analysis is offline. As a further improvement, openCV in Qt has been used to produce an online application that performs feature calculation, classification and peak counting (without dynamic time warping). [3] The counting result is also directly sent to a Google Glass through the wireless network for real-time feedback.

In future work, apart from gathering more data, with more exercise variances with the improved hardware, it is interesting to evaluate if the workout quality can be distinguished or measured. In computer vision, some skeleton

[3] The result has been demonstrated live interactively to public visitors on Cebit 2015 in Hannover for 5 days, and Government Open Door 2016 in Berlin for 2 days, with exercise counting of push-ups and abdominal crunching.

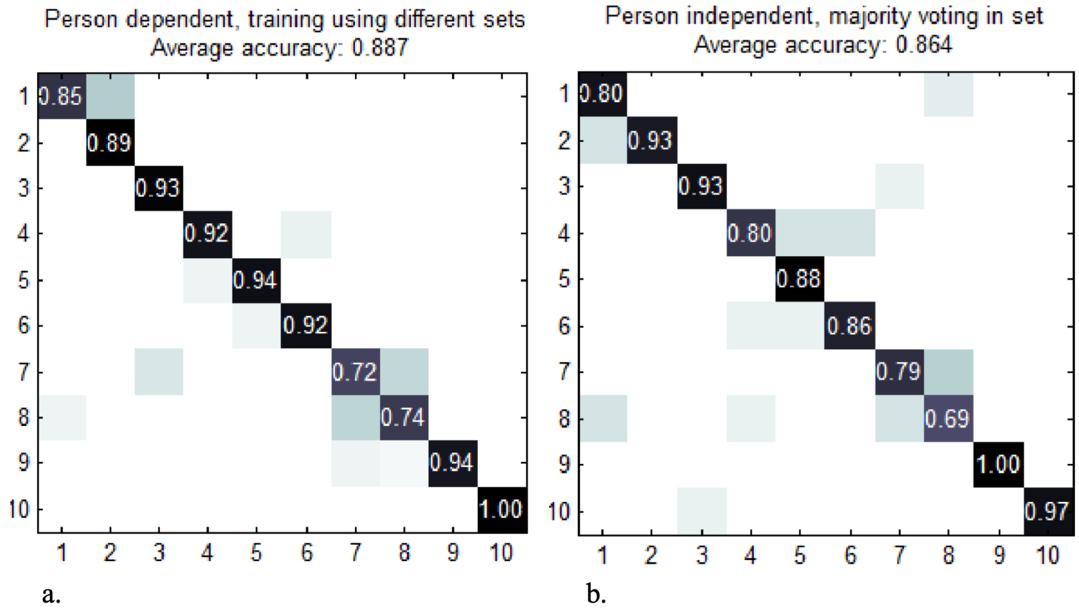


Fig. 29. Smart-mat Phase 1 classification confusion matrix: a. leave-one-recording-out person dependent result; b. leave-one-person-out with majority voting within set.

models can be constructed from the 2D or 3D image for activity tracking [83]. This inspires a similar algorithm proposal for matching a skeleton model based on the ground body-print pressure profile.

3.2.6 Phase 2 Study: Distinguish Minor Variances

The work of Phase 1 has proven that the distinct dynamic body prints can be used to distinguish different exercises. However, these exercises are drastically distinctive from each other. Therefore the exercises from Phase 1 is rephrased as 'exercise categories'. In Phase 2, we intend to distinguish further minor variations of exercises from the same category. Table 3 lists the categories and variations this study has defined. The exercise from Phase 1, standing dumbbell curls, is removed since it requires weights, and also the *propagation* from the design space point of view is *indirect* instead of *direct* as the other exercises. Each category has a shared reference pose. The variations are different types of exercises, different range/direction of motions, or different balances.

The dataset is consisted of 3 recordings from 12 participants each. 9 exercise categories and in total 47 exercise variations. Each variation is repeated for 10 times in every recording.

In Phase 2, the TPM enabled sports mat sensor has a larger area of 2-by-1 meters compared to the 0.8-by-0.8 meters. This allows the surface to cover the entire person's exercise range. While in Phase 1, when a person is lying down, the prototype could only cover the upper body; or when the person is in the bridge pose, it could only cover the upper shoulders. Thus the feet or legs cannot be covered by the Phase 1 prototype. Also, in the lunge exercise, one foot has to step out of the Phase 1 mat, making it impossible to acquire any information regarding both feet such as the distance or balance.

The feature extraction workflow described in Fig. 28 has been improved through other studies in this dissertation by the time of Phase 2, with details explained in Chapter 8. In particular, the pool of 10 frame descriptors is expanded to 17. Also, while converging temporal sequence of frame descriptors into time domain features, instead of simple statistical features, frequency features through Fast Fourier Transform and Wavelet Transform are added, resulting in 39 temporal features instead of 7 in Phase 1.

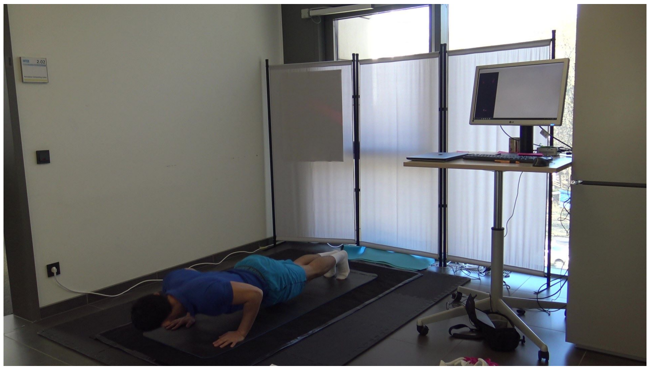


Fig. 30. Smart-mat study Phase 2 setup

3.2.7 Deep Features

The frame descriptors rely on image moments to describe the information about the shape. However, when it comes to minor variations when two classes share a common reference shape, these frame descriptors become less effective. One alternative is to use deep learning techniques. There are several pre-trained convolutional neural network models for image classification such as GoogleNet or Inception-V3. In general, these networks are trained by thousands of images to extract detailed information. Then the features

TABLE 2
Smart-Mat Phase 2 Study: Deep Features and Frame Descriptors Comparison

	Deep Features		Frame Descriptors	
	exclusive	cross-val	exclusive	cross-val
F-Tree	0.218	0.392	0.284	0.459
L-SVM	0.332	0.568	0.301	0.686
C-SVM	0.304	0.611	0.350	0.878
F-KNN	0.218	0.599	0.220	0.929
W-KNN	0.232	0.519	0.228	0.843
B-Tree	0.361	0.898	0.382	0.920

are fed to fully connected classification neural networks to classify the image. For our purpose, which is to extract TPM imagery information, the outputs of the last layer before the classification network can be used as features. Instead of using the pre-trained models' classification network, the said features are taken as *deep features* in the spatial domain, which is similar in concept with the frame descriptors. More details on the deep features are explained in Chapter 8.

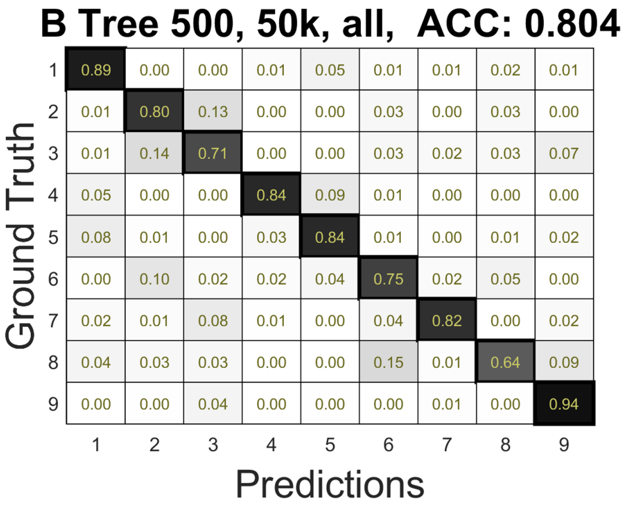
This study takes GoogleNet as the deep feature extractor model. It generates 1024 deep features, of which 362 have constant zero values. Then the same 39 temporal features are performed on the time sequences of the first 20 non-zero deep features.

3.2.8 Results

To compare the performance of the deep features and frame descriptors, the following procedure is used:

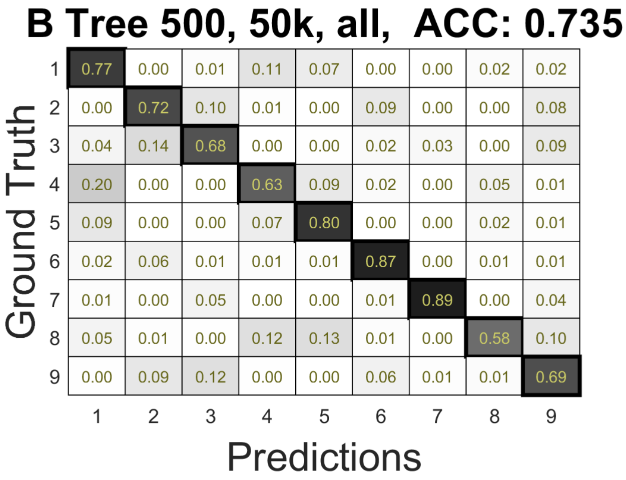
1) The dataset is divided into the training dataset and testing dataset with data from different participants in the two groups.
2) 5-fold cross-validation (cross-val) with the training dataset is performed on different classifiers to compare the performance of each, including: fine tree (F-Tree), support vector machine with linear kernel (L-SVM), support vector machine with cubic kernel (C-SVM), fine K nearest neighbour (F-KNN), weighted KNN (W-KNN) and bagged trees with 500 learners (B-Tree).
3) The trained classifier models are tested on the testing dataset, which is essentially person independent (exclusive) validation, meaning the models have never encountered the data from the testing person.

Table 2 shows the accuracy results of the said validation procedure. For reference, the chance level of 47 classes is 0.0213. With the lowest cross-validation result being 0.392, meaning the variations can be detected on some extent much better than the random chance level. B-Tree gives around 0.9 accuracies with both feature extraction methods. Also, for every classifier model, cross-validation normally out-performs the person independent exclusive validation. This is expected since in cross-validation, the data from the same person, or over-lapping sliding windows can exist both in the training and testing data pools. With all classifier models, in the cross-validation results, frame descriptors out-perform the 20 deep features by slight margins. However, since there are in total 662 valid deep features, the first 20 deep features chosen here are only the order of appearance in the network. There could be more effective features from the network, which requires further analysis in the future.



	1	2	3	4	5	6	7	8	9
1	0.89	0.00	0.00	0.01	0.05	0.01	0.01	0.02	0.01
2	0.01	0.80	0.13	0.00	0.00	0.03	0.00	0.03	0.00
3	0.01	0.14	0.71	0.00	0.00	0.03	0.02	0.03	0.07
4	0.05	0.00	0.00	0.84	0.09	0.01	0.00	0.00	0.00
5	0.08	0.01	0.00	0.03	0.84	0.01	0.00	0.01	0.02
6	0.00	0.10	0.02	0.02	0.04	0.75	0.02	0.05	0.00
7	0.02	0.01	0.08	0.01	0.00	0.04	0.82	0.00	0.02
8	0.04	0.03	0.03	0.00	0.00	0.15	0.01	0.64	0.09
9	0.00	0.00	0.04	0.00	0.00	0.00	0.01	0.00	0.94

Fig. 31. Smart-mat study phase 2 confusion matrix of 9 exercise categories with first 20 deep features × 39 temporal features



	1	2	3	4	5	6	7	8	9
1	0.77	0.00	0.01	0.11	0.07	0.00	0.00	0.02	0.02
2	0.00	0.72	0.10	0.01	0.00	0.09	0.00	0.00	0.08
3	0.04	0.14	0.68	0.00	0.00	0.02	0.03	0.00	0.09
4	0.20	0.00	0.00	0.63	0.09	0.02	0.00	0.05	0.01
5	0.09	0.00	0.00	0.07	0.80	0.00	0.00	0.02	0.01
6	0.02	0.06	0.01	0.01	0.01	0.87	0.00	0.01	0.01
7	0.01	0.00	0.05	0.00	0.00	0.01	0.89	0.00	0.04
8	0.05	0.01	0.00	0.12	0.13	0.01	0.00	0.58	0.10
9	0.00	0.09	0.12	0.00	0.00	0.06	0.01	0.01	0.69

Fig. 32. Smart-mat study Phase 2 confusion matrix of 9 exercise categories with 17 frame descriptors × 39 temporal features

All the evaluation of phase 2 follows the process of leave-3-people-out cross-validation. To compare with Phase 1, the best performing classifier model B-Tree is used to classify among the 9 categories. The confusion matrix of deep features is drawn in Fig. 31 and that of frame descriptors is in Fig. 32. In Phase 1, it is assumed that within every repetition, all the sliding windows belong to the same exercise. Therefore majority voting within every repetition is used, which essentially further improved the final accuracy. However, in Phase 2, majority voting within a repetition is not used since variations can happen during consecutive exercise actions.

Deep feature, in this case, performs better than frame descriptors by 7%. The miss-classifications gives further insights on how the two feature groups differ. In both feature groups, Category II (Push-ups) and Category III (Planking) are similarly miss-classified. Both categories share a similar pose that two hands or forearms are in the front while two feet are in the back. With frame descriptors, Category VIII (Bridge) is often confused by the classifier with Category IV (Only Back Visible) and Category V (Back and Arms Visible). This can also be caused by the fact that all three categories have similar overall body-print shapes. The same can be said with frame descriptors for Category IX (Side Plank) against Category II (Push-ups) and Category III (Planking). However, with deep features, these two cases are very well distinguished with almost no miss-classification. This could be the result that CNN models like GoogleNet distinguish components and details within images, and emphasize less on the overall shape of the whole picture. This assumption can also be supported by the result that in the deep features confusion matrix, some Category VIII (Bridge) data samples are classified as Category VI (Standing Up). In this case, the frame descriptors method successfully distinguishes the two categories, since standing and bridge cast distinct body-print on the mat. In both categories, two feet are present in the body-print. This could contribute to the miss-classification by the model trained by deep features.

The confusion matrices of the exclusive validation results with all 47 variation classes are shown in Fig. 33 for the deep features and Fig. 34 for the frame descriptors. From the confusion matrices, it appears that even the overall accuracy is less than 0.4, most of the miss-classifications are within categories. For example, for both methods, a prominent group of miss-classification is class 31 (shallow squat), 32(deep squat) and 33(toe touch), all under Category VI (standing up). All of the three variations are apparently two feet standing at the same position, with different up-and-down motions. From the design space point of view, these activities are indirectly propagated onto the TPM sensor. The same can be said with class 44 (Side Hip Thrust), 45 (Side Leg Raise), 46 (Arm Swing I) and 47 (Arm Swing II) under Category IX (Side Plank). Some variations have over 70% classification accuracy such as 28 (Leg Swing I) and 29 (Leg Swing II).

3.2.9 Conclusion

Overall, the Smart-Mat study has proven through both Phase 1 and Phase 2, that TPM sensors implemented inside an exercise mat can be used to distinguish various exercises through the dynamic change of body-prints. Directly propagated body-prints by the exercise, such as push-ups or crunches, can be most effectively distinguished. Further, indirectly propagated exercises can also be distinguished, such as through the pressure changes under the feet while doing upper body movements, including dumbbell curls, squats or touch toes. Phase 2 also established a more comprehensive dataset in this context with 47 exercise variations. Those variations can also be well distinguished 38.7% accuracy against the 2.13% chance level.

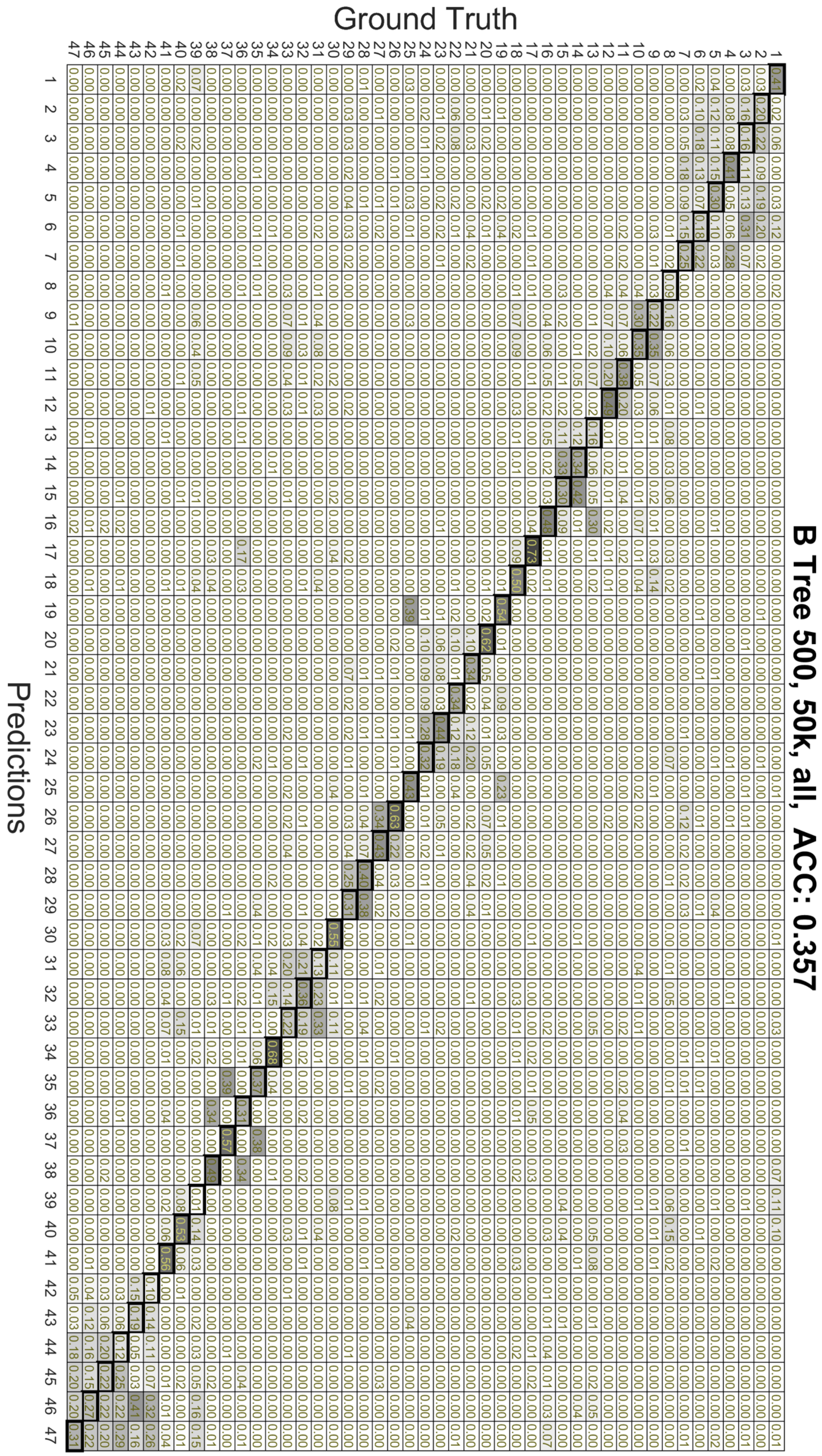


Fig. 33. Smart-mat study Phase 2 confusion matrix of 47 variation exercise classes with first 20 deep features × 39 temporal features

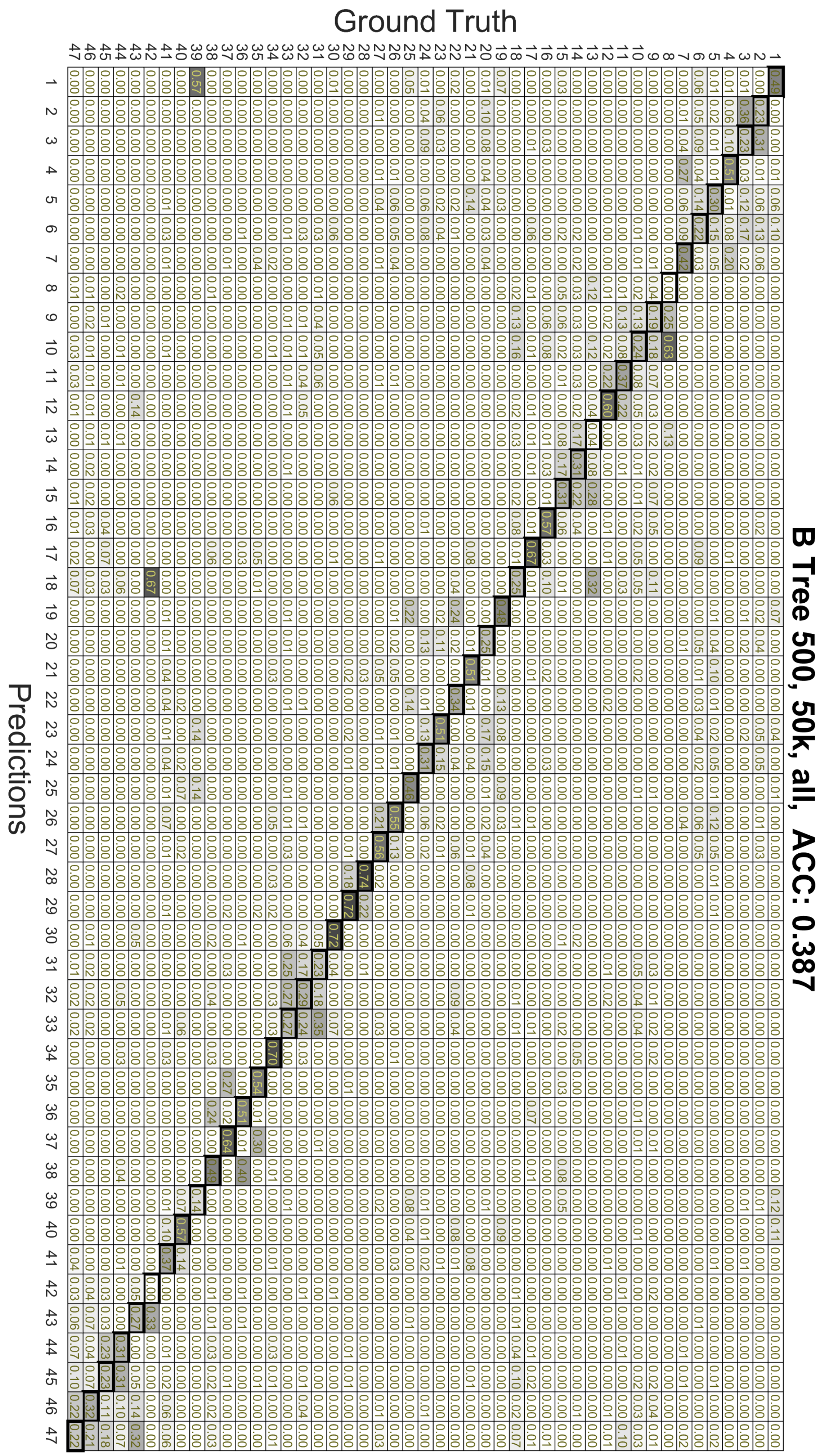


Fig. 34. Smart-mat study Phase 2 confusion matrix of 47 variation exercise classes with 17 frame descriptors × 39 temporal features

TABLE 3
Smart-Mat Phase 2 Study: Class Definition

Index	Exercise	Variation and Descriptions
Category I : Crunches		
1	Reference: Lie Down	knees up, feet down
2	Crunches I	lower range of motion, regulated by hands pointed to the sky with arms straight
3	Crunches II	medium range of motion, regulated by hands reaching to the knees with arms straight
4	Crunches III	higher range of motion, regulated by elbows reaching to the knees with arms straight
5	Side Crunches I	lower range of motion, regulated by hands pointed to the sky with arms straight
6	Side Crunches II	medium range of motion, regulated by both hands pointed to each knee alternatively
7	Side Crunches III	higher range of motion, regulated by elbow touching the opposite knee alternatively
Category II : Push-ups		
8	Reference: Hold Push-up	hands shoulder wide
9	Push-up I	hands shoulder wide and half range of motion, regulated by not fully extending elbows
10	Push-up II	hands shoulder wide and full range of motion, regulated by having elbows fully extended
11	Push-up III	hands wider than shoulder, full range of motion
12	Alternating Push-up	alternating single side push-up, hands wider than shoulder.
Category III : Planking		
13	Reference: Standard Plank	depending on the person, either feed or knees can be on the floor
14	Slack Plank	torso not engaged so the spine bends downwards naturally
15	High Hip Plank	raise hip higher than standard straight plank
16	Plank Dip	Standard Plank but move body forward and backwards around the elbow support
17	Plank Push-up	hands are chest wide, change between plank position and push-up position
18	Chest Wide Push-up	hands are chest wide (same as 17), only push-up without placing elbows on the mat
Category IV : Only Back Visible		
19	Reference: Back	lift legs and elbows, place hands on the back of the head
20	Leg-up Crunches	lift legs, crunching exercise till elbows touch the knees
21	Alternating Cycling	lift legs, alternately touch one elbow with the opposite knee, while the other opposite elbow-knee pairs are extended
22	Leg-only Cycling	lift elbows, empty cycle with only leg motions
23	Leg-lift I	raise legs from flat position to vertical position, with lower back pressing on the mat, this variation engages the abdominal muscles more effectively
24	Leg-lift II	same leg motion as 23, but have lower back suspended with an arched spine, and instead use hip as the anchoring point, this variation engages the abdominal muscles less effectively
Category V : Back and Arms Visible		
25	Reference: Hold Leg-up	with arms and back relaxed on the mat, while legs are raised up vertically
26	Leg-raise I	raise leg vertically and upwards from the relaxed horizontal position with lower back suspended
27	Leg-raise II	similar as 26, but always keep thighs upwards without lowering legs to the horizontal position, this variation engages the torso muscles more than Leg-raise I
28	Leg Swing I	swing legs from the left and horizontal, then upwards vertical, to the right horizontal positions, with knees bent
29	Leg Swing II	similar as 28, but knees are extended straight
Category VI : Standing Up		
30	Reference: Stand	feet shoulder width, feet angle is decided by squeezing glutes
31	Shallow Squat	empty squat, half range of motion, regulated by with relaxed arms, the hands reach the knees
32	Deep Squat	empty squat, full range of motion, regulated by with relaxed arms, the elbows reach the knees
33	Toe Touch	bend downwards and reach toes with fingers repetitively, heels may leave the mat
34	Tip Toe	raise the heels repetitively
Category VII : Arch Step		
35	Reference I	alternately step forwards and then back, with one foot staying behind
36	Reference II	alternately step backwards and then front, with one foot staying in front
37	Lunge forwards	step similar as 35, but with the knee behind bent downwards
38	Lunge backwards	step similar as 36, but with the knee behind bent downwards
Category VIII : Bridge		
39	Reference: Bridge	with feet or heels, upper back, shoulders, arms and head staying on the mat, suspend the hip and lower back so that the thighs and torso are on the same slope line
40	Hip Thrust	same pose as 38, but move hip up and down with the help of glutes
41	Single Hip Thrust	same motion as 39, but with only one leg supporting the body and the other leg extended
Category IX : Side Plank		
42	Reference: Side Plank	body face to one side, with one foot or knee and one elbow of the same side supporting the body, torso and legs are straight on the same line
43	Slack Side Plank	similar pose as 41, but with the torso relaxed
44	Side Hip Thrust	similar pose as 41, and move hip up and down
45	Side Leg Raise	similar pose as 41, and raise and lower the leg that is not supporting the body
46	Arm Swing I	similar pose as 41, and swing the free arm from below the body to the vertical upwards position, face the front of the body
47	Arm Swing II	similar motion as 45, but face the moving hand

3.3 ID Carpet

3.3.1 Introduction

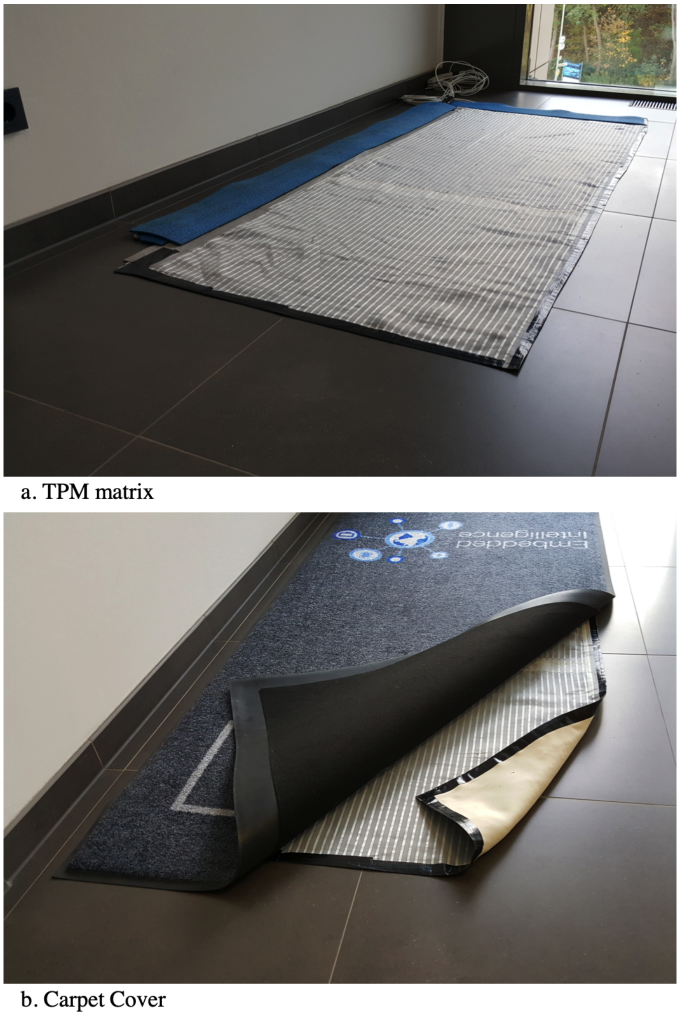
In most of the previously discussed algorithms for extracting information from TPM imagery, features calculated from time and space domain are all rather hand-picked. Although through the feature selection discussion in Chapter 8, specific applications can use only part of those pre-defined features, it is still less than a fully automatic process. Artificial neural networks are generally 'black boxes' that learn the relationship between the input and response by enhancing the neural connections in the network. This section continues on the discussion of utilizing deep features from the previous section.[4]

Convolutional neural networks (CNN) [84] can achieve super-human accuracy on image classification tasks [85] [86] [87] [88]. Recurrent neural networks (RNN) especiall with Long Short-Term Memory cells (LSTM) [89] have been used to classify sequences of images [90] and to recognize activities [91] [92]. Since TPM generates imagery data which is similar to pictures or videos, and computer vision techniques have already been used in extracting features, it is interesting to see how deep neural networks can recognize activities from the TPM imagery data.

A major limitation of using deep learning techniques is that the training of the networks requires a large amount of annotated data. In the computer vision discipline, images are easy to acquire, such as the ImageNet [93] and MS-COCO [94] datasets. However, for novel sensing techniques, a dataset on such scales is difficult, if not impossible, to acquire. This limitation can be overcome by performing transfer learning, i.e., using labelled data from one domain and transferring the learned knowledge or model to a target domain. A typical scenario for transfer learning involves using a convolutional neural network trained on a very large dataset, and then further fine-tune it on the target dataset which is relatively small in size. An already trained CNN is used for transfer learning by removing the last fully-connected layer and using the activation responses of the last hidden layer as the feature descriptors of the input dataset. The resulting feature descriptors (deep features) are then used to train a classification model.

3.3.2 Experiment and Baseline

The dataset used for evaluating transfer learning is acquired in a carpet application scenario. As people walk, we cast dynamic morphing sequences of our footprint as shown in Fig. 37. The study is carried out in [60]. The hardware is an improved version of the 'smart-mat' mentioned in the previous section. The electronics is Version F-2.1 from Table 17 in Chapter 2. The TPM matrix is made of metallic fabric stripes and CarboTex, and thus is fully fabric as shown in Fig. 35. There are 120 × 54 points with 1.5cm pitch, each sampled at 25 Hz with 24-bit ADCs.



a. TPM matrix

b. Carpet Cover

Fig. 35. ID carpet: Hardware

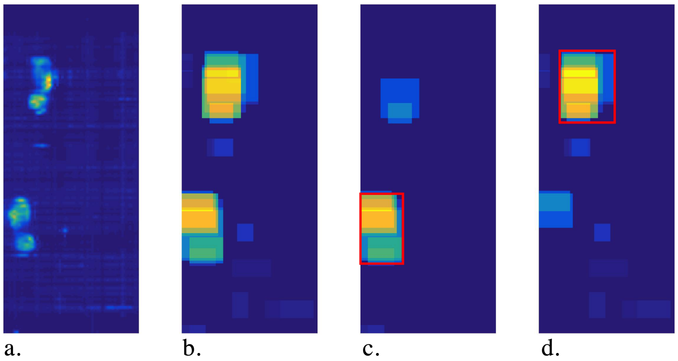


Fig. 36. Step segmentation: a. sum of pressure mapping over a walking event, b. all bounding boxes during the event, c. detection of the first step, d. detection of the second step

From the design space point of view, the important dimensions in this study are the *intention* and *coverage* dimensions. As a floor based biometric system, the *intention* dimension can be *unaware*. This has a strong indication in the privacy and ethical aspects, as the users can give away their identity simply by a single step without noticing a biometric monitoring device is present. The *coverage* is *excessive*, thus the system can be implemented in a corridor

[4] This section is based on the publications [60] and [58]. As the leading author of [60], I have only included my own work; as the supervisor of the first author of [58], Section 3.3.3 is a brief summary of the method and results from the work by Singh, Monit Shah and Pondenkandath, Vinaychandran. Since the published work has gone through extensive wording, some sentences and passages have been quoted verbatim from the said sources.

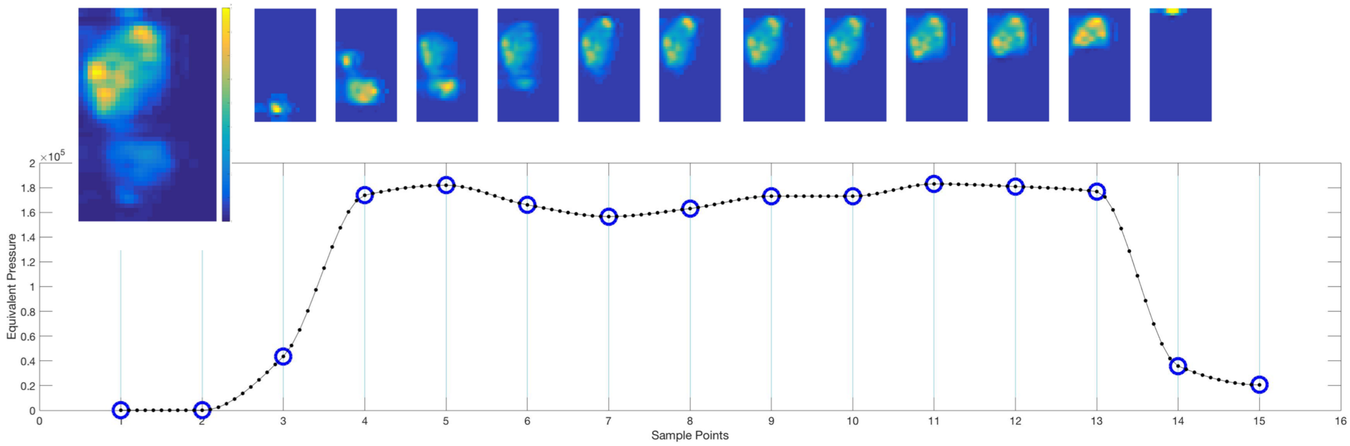
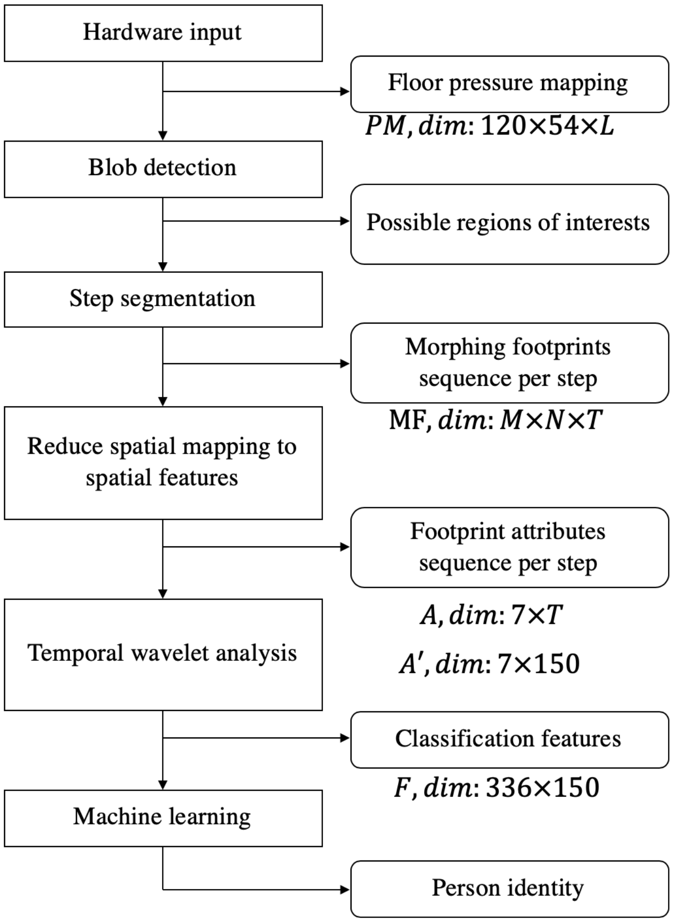


Fig. 37. The morphing footprints of one step (top) and the first attribute - average pixel value (bottom), The blue circles are original data A_1 and the black dots are A'_1



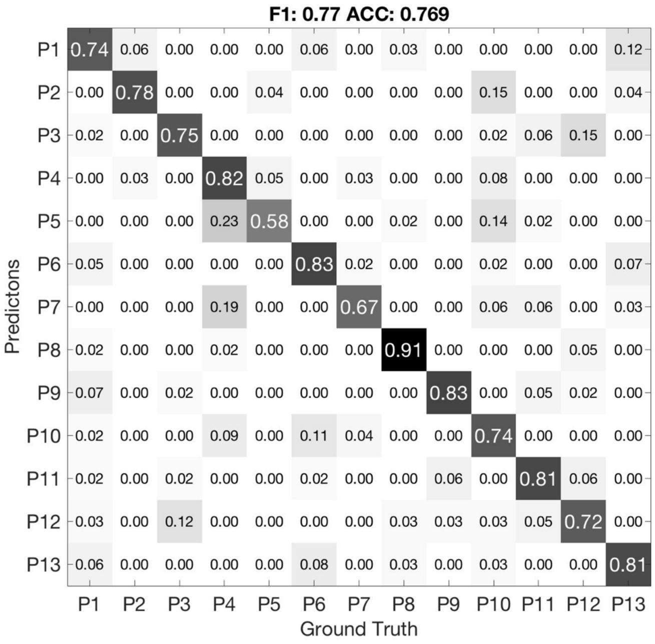
Hardware input
Floor pressure mapping
$PM, dim: 120\times54\times L$
Blob detection
Possible regions of interests
Step segmentation
Morphing footprints sequence per step
$MF, dim: M\times N\times T$
Reduce spatial mapping to spatial features
Footprint attributes sequence per step
$A, dim: 7\times T$
Temporal wavelet analysis
$A', dim: 7\times150$
Classification features
$F, dim: 336\times150$
Machine learning
Person identity

Fig. 38. Gait analysis workflow

or walk path.

13 people participated in the experiment; each walks with their normal shoes on the smart carpet of at least 12 rounds. Overall, 529 steps were recorded. The ground truth includes the starting and ending time for every step, and their identity.

To offer a baseline for further deep learning evaluation, a workflow with standard feature-classification process is established as shown in Fig. 38. First, the footprint steps need to be segmented by the algorithm. Take Fig. 36(a), which is an accumulated frame during a time window, for example, segmenting the steps follows this procedure:


F1: 0.77 ACC: 0.769

Predictons \ Ground Truth	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13
P1	0.74	0.06	0.00	0.00	0.00	0.06	0.00	0.03	0.00	0.00	0.00	0.00	0.12
P2	0.00	0.78	0.00	0.00	0.04	0.00	0.00	0.00	0.00	0.15	0.00	0.00	0.04
P3	0.02	0.00	0.75	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.06	0.15	0.00
P4	0.00	0.03	0.00	0.82	0.05	0.00	0.03	0.00	0.00	0.08	0.00	0.00	0.00
P5	0.00	0.00	0.00	0.23	0.58	0.00	0.00	0.02	0.00	0.14	0.02	0.00	0.00
P6	0.05	0.00	0.00	0.00	0.00	0.83	0.02	0.00	0.00	0.02	0.00	0.00	0.07
P7	0.00	0.00	0.00	0.19	0.00	0.00	0.67	0.00	0.00	0.06	0.06	0.00	0.03
P8	0.02	0.00	0.00	0.02	0.00	0.00	0.00	0.91	0.00	0.00	0.00	0.05	0.00
P9	0.07	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.83	0.00	0.05	0.02	0.00
P10	0.02	0.00	0.00	0.09	0.00	0.11	0.04	0.00	0.00	0.74	0.00	0.00	0.00
P11	0.02	0.00	0.02	0.00	0.00	0.02	0.00	0.00	0.06	0.00	0.81	0.06	0.00
P12	0.03	0.00	0.12	0.00	0.00	0.00	0.00	0.03	0.03	0.03	0.05	0.72	0.00
P13	0.06	0.00	0.00	0.00	0.00	0.08	0.00	0.03	0.00	0.03	0.00	0.00	0.81

Fig. 39. Gait analysis confusion matrix

1) Dynamic thresholding is performed to remove the background. Every frame is sorted into a 10-bin histogram, the threshold is decided as the center value of the next bin of the highest count bin. This is because, for a carpet, most of the pixels are the background.
2) Then bounding boxes are placed on the frame by blob detection. Fig. 36(b) shows all of such bounding boxes added up together. Each box is filled with the average pixel value within its region.
3) If the box's average pixel value is higher than a second dynamic threshold, it is decided as the start

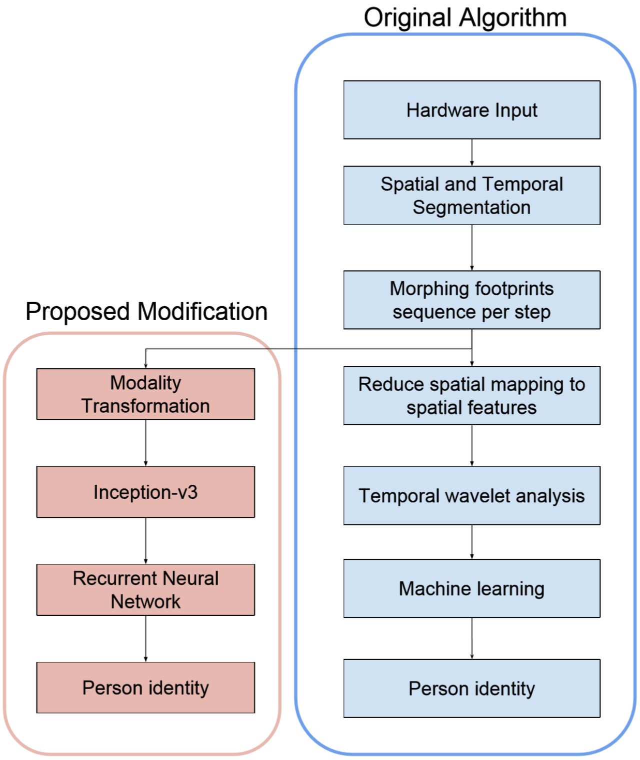


Fig. 40. Gait analysis workflow with neural networks

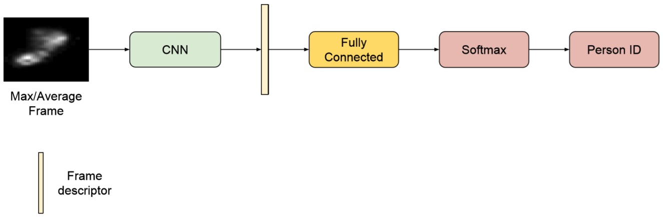


Fig. 41. Pipeline diagram of using key frames with CNN.

of a step (a spawning point). This second threshold is calculated again with a similar histogram selection method, but of all the average pixel values of the boxes of all the frames within a single walking event.

4) After the spawning point, the step is tracked by looking for the box whose center is closest to the step box in the previous frame, and no more than 30 pixels away. When no such box is present, this step is decided as finished.

5) The next step detection does not happen after the finishing of the previous step, but for every frame, the algorithm looks for new spawning points that are further away from the detected steps by 30 pixels. This is because the next step usually happens before the previous step finishes. This also makes the algorithm natively support multiple concurrent steps.

The step segmentation algorithm for the start and end time results in 97% precision and 91% recall considering an error margin of 2 frames.

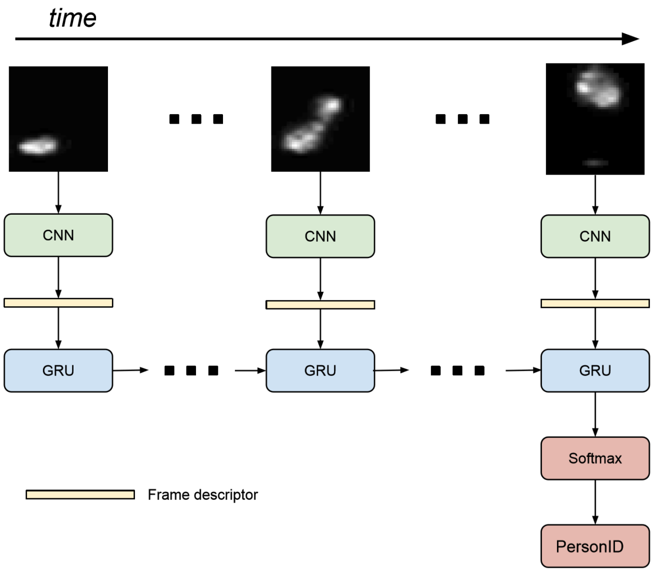


Fig. 42. Pipeline diagram of using sequences of frames with RNN.

Every detected footprint is cropped with the bounding box that covers all its bounding boxes, then seven frame descriptors are calculated:

- average pixel value;
- centroid coordinate;
- maximum pressure point's value;
- maximum pressure point's coordinate;
- pressed area.

Every step has then 7-by-T (T is the number of frames) of temporal sequences of frame descriptors as a set of A_i. It is interpolated on the time domain to 150 samples as A'_i as shown in Fig. 37 with the dash line.

The first set of features is statistical representations of A'_i: $mean(A'_i)$,$std(A'_i)$,$var(A'_i)$,$max(A'_i) - min(A'_i)$. Then wavelet transform as mentioned in Chapter 8 is performed, with 10 filterbank iterations and 'db8' as the mother wavelet. Then features are calculated by the same method from Chapter 8. For every filterbank's wavelet transform coefficients, mean, variance, standard deviation, skewness and kurtosis are calculated. In total, every step has 336 features.

A support vector machine classifier with the quadratic kernel is used for the 10-fold cross-validation. The resulting confusion matrix is shown in Fig 39, with an accuracy of 76.9%.

3.3.3 Deep Neural Network

To use deep learning methods with transfer learning, the general approach is to replace the features calculated from pre-processed morphing footprint imagery, as the workflow in Fig. 40. First, the frames are converted into gray-scale images with a linear colormap, with brighter pixels representing higher sensor value. Since CNN can be used to recognize individual images and RNN can be used to recognize an image sequence, both approaches are evaluated. With CNN, two *key frames* - maximum frame and average frame (KF_5 and KF_1 from Chapter 8) are used from each step. Key frames are a concept proposed in this work, to use static spacial matrices to represent the signal changes from a temporal window of spacial frames, by performing operations per pixel along the time axis. With RNN, the entire sequence of the step is fed into the neural network.

The transfer learning approach follows the idea of transferring from the image classification task, i.e., using a pre-trained model from ImageNet or Coco-DB. Either the classification layer is removed or used as a feature extractor and a new classification layer is added. Thus the CNN is used as a fixed feature extractor. The CNN is provided with the transformed image input, which is resized to fit the CNN input size. The activations for the entire network are computed by forward propagating the input through the network. The pre-trained CNN in this project is the Inception-v3 model from [95], which is a CNN variant that focuses on improving computational efficiency along with performance. The Inception-v3 architecture consists of 17 layers: 3 convolutional layers followed by a pooling layer, 3 convolutional layers, 10 Inception blocks and a final fully connected layer, and requires the image input to be of 229 × 229 pixels. The activations from the fully connected layer is used as a 2048 dimensional output for every image input, which is then used as a feature vector for further classification with a new fully connected layer. The pipeline is shown in Fig. 41.

For the RNN method using the entire sequence of frames, the pipeline is shown in Fig. 42. All of the frames associated with a step are processed through the Inception-v3 model to extract a single descriptor for each frame. These descriptors are then fed one after another into a layer of Gated Recurrent Units (GRU) [96], which generates a classification result upon completing the sequence.

The cross-validation results are summarized in Table 4. From it, it can be seen that using the average frame with a single CNN and the complete sequence with RNN result in higher accuracy than the baseline using conventional feature extraction processes. Especially, with RNN, taking the information of the time sequence, the result is $> 10\%$ higher than the conventional approach. Note that in the conventional approach, wavelet analysis also takes information about the pattern on the time domain, but without the time order.

TABLE 4
Accuracy comparison of different approaches

Feature Type	Image Set Type	Accuracy	to baseline
Conventional	All sequences in a step	76.9%	0%
Deep CNN	Max Frame	71.99%	-4.91%
Deep CNN	Average Frame	78.41%	1.51%
Deep RNN	All sequences in a step	87.66%	10.76%

3.3.4 Significance and Outlook

The major significant finding of this project is that, dynamic pressure profiles from individual footsteps can be measured by a TPM enabled carpet and be used for person tracking and identification. The approach in this work only requires a single footstep, instead of looking at the relationships between adjacent footsteps.

Another significance of this study is on the cross knowledge domain transfer learning. Normally, transfer learning of neural networks is from one image dataset to another image dataset. In this project, a neural network trained by images such as cat, car, people, etc., is able to detect pressure mapping sensor data, and outperform conventional methods with statistical and frequency features. And more interestingly, this dataset in particular, every sample is just a morphing footprint, which is even difficult for actual humans to distinguish.

This project is still on a small scale. The core concept should be elaborated in the future work. First, more datasets from other applications with the TPM sensor should be evaluated. And in this project, the baseline features are only part of the entire feature base introduced in Chapter 8. It is interesting to see when the conventional feature approaches are exhausted, if the deep learning approaches are still superior. Also, it will be interesting to test on other pre-trained models. In this project, the CNN is essentially used as an automated feature calculation model; feature selection algorithms such as Neighbourhood Component Analysis (NCA) from Chapter 8 can be used here to investigate which features from the CNN model have more contribution. Also, it would be interesting to compare the result of using standard linear classifiers such as KNN or SVM with those CNN calculated features.

And ultimately, the vision is to develop a 'TPM AI' that is a unified neural network model, which can recognize activities not only per application, but also distinguish what the application is. Just like human experts in TPM sensor data, when the experts see the pressure imagery data, they may already be able to tell the application scenario.

Chapter 4 Object Subspace Empirical Studies

There has been a wide range of studies involving smart objects in ubiquitous computing, especially with the rise of the Internet of Things [97] [98]. Smart objects are typically daily objects augmented with sensors and connectivity to give insights on people's activity or behaviour. Actuators can also be implemented inside smart objects to offer interactive experience or information feedback.

Smart objects can help offer information on several levels, including:

- Control input: the simplest being the already omnipresent binary sound or motion detectors for door and lighting controls. With more potent sensors that can offer information on multiple dimensions, such as X, Y, Z axis, smart objects can serve as more fine-grained control input such as navigating a cursor or tuning intensities. In this regard, TPM sensors can offer both two-dimensional localization information (X and Y axis) and intensity information at every location (Z axis).
- Utility/object usage: the usage information of certain objects, including the times of use, consumption quantity, or the usage context, is widely studied in areas like workplace management and elderly assistant. TPM can sense not only if an item is in use, but also how much specific parts of the item has been used through touch.
- Emotional states: human factors such as phycological and emotional states are an important aspect in the field of human-computer interaction, computing systems, education, elderly care, etc. [99] [100] [101] Emotional states are well known to be related to bodily contacts [102], which can be measured by touch-sensitive sensors such as FSR or TPM sensors.

From the design space aspect, in the object subspace, the *Propagation* dimension is mostly *direct*, since typically users interact with smart objects through contact. The Intention dimension can be any of *intended*, *unintended* and *unaware*. In interactive smart objects, the intention is obviously intended. And in smart object studies that aim at using the objects as a media to acquire information on the users' behavior and psychological aspects, and knowing the smart objects may influence the result, the intention is thus either unintended or unaware. The TPM can either be partially or exactly covering the augmented objects or the target users. Virtual surface can be used to accommodate irregular shapes of coverage, such as different parts of a toy or a robot. The *Sampling* rate of smart object TPM sensors depends on the target activity. For example, usual interactive touch can be measured with around 10 to 30 Hz, but fast-moving gestures, or sports activities require higher speed sampling rates. Since smart objects are supposed to be moved around, the *Mobility* dimension is usually *mobile*. While in some studies, where the data throughput (decided by the number of sensing points and sampling rate) exceeds the mobile bottleneck, *mobility* can be compromised to *movable*, especially in the early prototype stage.

This chapter covers three empirical studies in the object subspace.

- Seat Cover: (object, exact match) while the algorithm to detect posture from the back of a seat is simple yet efficient, this application demonstrates the entire software stack including all four layers from Fig. 90 in Chapter 6, to give user real-time feedback.
- Robot Skin: (object, canvas) in this application the wavelet analysis features are examined in depth, as the application goal is to distinguish emotional related touch gestures.
- Student Workshop: a modular framework of the TPM system including both hardware and software resources are given to groups of master students in an organized workshop to implement their own ideas. Through the workshop, not only various projects were rapidly developed, the TPM framework implementation is also proven to be helpful in the computer science education process.

4.1 Seat cover

This chapter presents a smart object sensing system in the form of a wireless chair cover that unobtrusively monitors people's posture on ergonomic design chairs, by covering the back piece of the chair and measuring the pressure profile of the user's back (back body-print). The sensor system has 1024 sensitive points, covering 48-by-48 cm^2 area. With a simple and efficient classification algorithm, the classification accuracy is around 80% among 10 postures, including lordotic and kyphotic lumbar spine on different extends, and lean to the sides on different extends. The web browser based user interface offers timely and reprogrammable intervention from the user's posture history. [5]

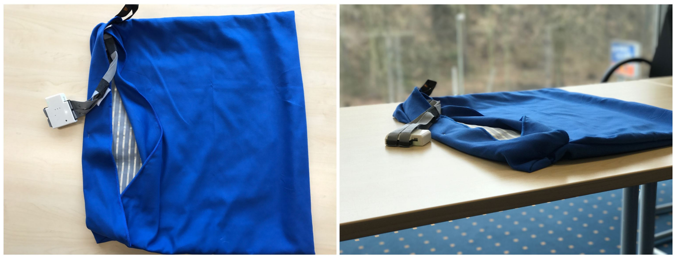


Fig. 43. Internal of the smart seat cover

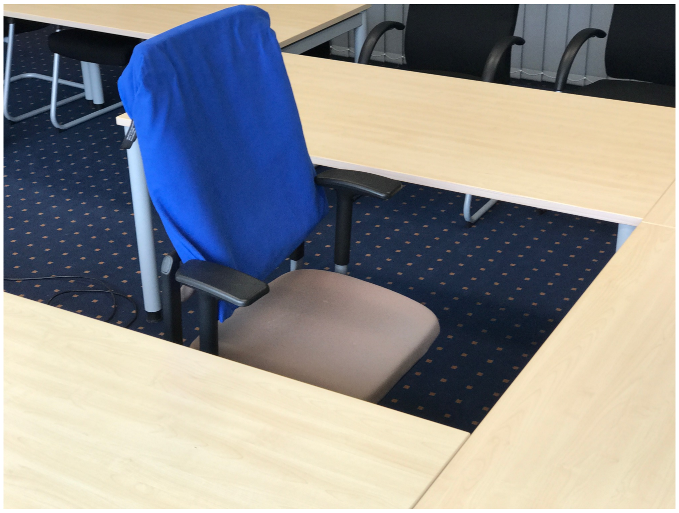


Fig. 44. A chair equipped with the smart seat cover

4.1.1 Empirical Study Background

In the office, transportation, home, etc., scenarios, ergonomic products that are designed to offer users better comfort, regulate their postures and help them relieve stress for seating and relaxing are ubiquitous. However, the manufacturers and ergonomic designers have very little control or feedback on the proper usage of the post-sale ergonomic products; while the users, on the other hand, would mostly use the products in an intuitive way.

For example, many office chairs offer adjustable features to accommodate a wide range of sitting preferences. As studies have shown in [103], sometimes the users may use ergonomically designed chairs in a comfortable way which they were used to (before adopting the new chair), yet is not the way the designers and ergonomic engineers intended the products to be used, nor how physiotherapy studies have recommended. Moreover, as the user's attention may be occupied by their activities such as working, interacting with other people or objects, or relaxing, they could stay in certain postures for extended periods, which is also shown by physiotherapy studies to promote increased stress on the spinal region. [104] By making 'bad postures' comfortable, this could have none or negative influences on the users' health and quality of life.

In modern society, people with sedentary occupations spend a major part of their time sitting on chairs. [105] [106] A wide range of clinical ergonomic and orthopaedic studies have suggested that inappropriate postures can have negative effects on the cervical, thoracic, lumbar part of the spine region, shoulder and pelvis, including fatigue, stress or pain. [107]

Traditionally, children are taught to sit uptight for a better back; however, studies have shown different postures have different effects on the spine and muscles, without an agreement of which posture is the best for sitting. For example, in [108], two basic postures ('erected posture' and 'flexed posture') are studied, concluding that sitting with the flexed posture mechanically flattens the lumbar spine and is thus more beneficial when sitting and lifting heavy weights compared to an erected posture which imposes a lumbar lordosis. In [109], O'Sullivan, et al. investigated the trunk muscle activations in different postures similar in the last study. The findings suggest that in 'erected posture' the lumbar multifidus and internal oblique muscles are significantly more activated; while in the 'thoracic upright posture', the neck muscles are significantly activated, this posture thus causes more shoulder region stress as supported by the study in [110]. And in the 'flexed and slump posture', most of the trunk muscles investigated remain relaxed. Thus, it appears flexed and slump postures, or 'lumbar kyphosis postures' is more beneficial for the spine and disc structures and causes less muscle stress. However, it may cause greater stress to articular and ligamentous structures [111] [112], as well as stress on the anterior annulus and an increase in the hydrostatic pressure in the nucleus pulposus at low load levels [108].

Based on various evidence, it appears that various postures have their own advantages and disadvantages. In [104], Vergara and Page suggest that low mobility are the principal causes of the increase in sitting discomfort. This study agrees with the muscle activation studies and considers that static muscular effort is a major contributor to short term lumbar and dorsal pain. It is supported in more recent studies such as [113], which shows less sitting fatigue in dynamic sitting compared to static sitting.

This calls for a method that automatically detects the user's real-time posture, compare to the recommendations from clinical ergonomic expert knowledge, and distinct chair manufactures, and offer timely feedback. Studies have shown that pressure sensors placed between the user and

[5] This section is based on the publication [36]. As the leading author of the publication, I have only included my own work; unless other authors' contribution is an integral part of explaining my work, in which case it will be explicitly stated. Since the published work has gone through extensive wording, some sentences and passages have been quoted verbatim from the said source.

TABLE 5
Seat Cover Study: Class Definition

Class	Definition
C1	**Sit straight up**, with the spine up tight (the back piece is locked straight, all the others the chair is unlocked)
C2	**Sit with flexed spine**, look forward (as if looking at a computer screen)
C3	**Sit with flexed spine**, look deep downwards (as if writing on the desk, or looking at a smartphone on the legs)
C4	**Lean back**, the back fully in contact with the chair
C5	**Lie on chair**, slide down from the lean back posture, with the lower back suspended from the chair
C6	**Reach to left**, with body facing the side (as if talking to people, or operating the telephone, etc. on the side)
C7	**Lean left**, with the upper body's weight focused on the armrest, face front (as if looking at a computer screen)
C8	**Slight lean left**, the person's weight is still on the back piece, without elbow support, but the spine is slightly bent to the side
C9	**Reach to right**, with body facing the side (as if talking to people, or operating the telephone, etc. on the side)
C10	**Lean right**, with the upper body's weight focused on the armrest, face front (as if looking at a computer screen)
C11	**Slight lean right**, the person's weight is still on the back piece, without elbow support, but the spine is slightly bent to the side
C12	**Not a posture**, no user is seated on the chair, instead, some bags may be put on the chair or jackets hang on the back piece.

the chair can be a useful source of information. For example, in [114], four pressure sensors are placed at the back of the chair, and in [115], an 8-by-8 pressure sensor matrix is placed on the seating surface. However, most of the systems used in such studies are not suitable for public consumers in their everyday lives.

4.1.2 Hardware Requirement

Since a swivel office chair can be dragged around during work, the sensing system should be tether-free. Thus the hardware version M-1.1 from Table 17 in Chapter 2 is used to power a 32-by-32 TPM matrix with a pitch of 1.5 cm. Every pixel is scanned with the integrated 12-bit, multi-channel ADC from the micro-controller dsPIC33FJ256GP710A at 20 frames per second. The data is sent to a computer through the Bluetooth module RN42 on the electronics board. The TPM sensor is covered inside a microfiber shroud, resulting in a thin and soft fabric piece as in Fig. 43.

An office chair, the 'Please' model from Steelcase is used, which is ergonomically designed to offer many adjustment options, including: height and horizontal retraction of the seating piece for different leg lengths; back piece height and maximum tilt; the back piece is also separated into two parts, with spring support so the curvature can fit the user's back. The TPM sensor is hung on the back piece to measure the seater's back pressure profile. It is also possible to place the sensor at the seating piece. However, this study opted for the back piece because (1) the pressure under the user's body is much greater than behind the back, which may cause deep wrinkles to the fabric and (2) from the user experience point of view, people are generally more comfortable being monitored at the back than under the buttocks.

4.1.3 Experiment

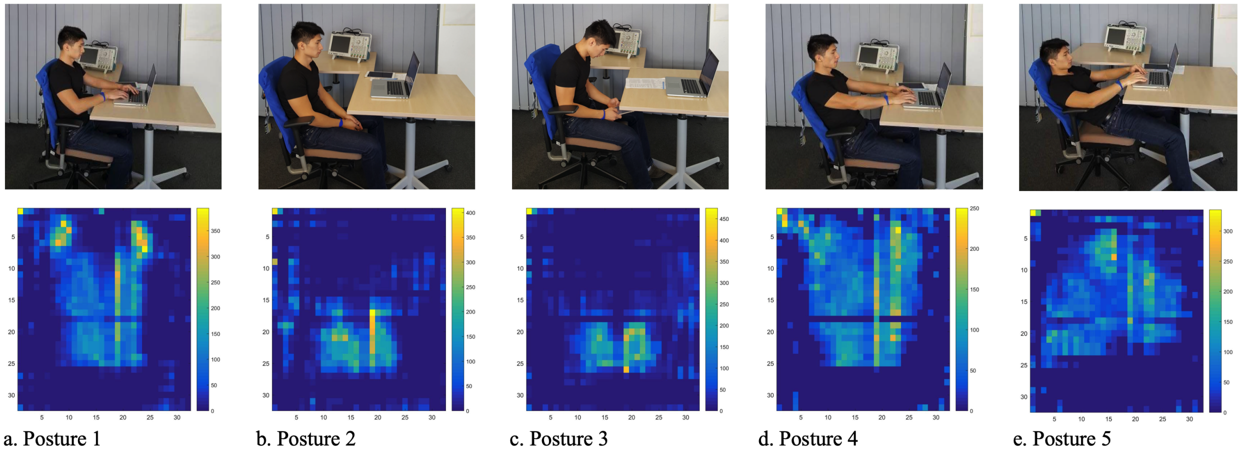
First, to validate the methodology and to establish a training dataset, a controlled experiment is set up, where participants are asked to sit at the chair with defined postures listed in Table 5, with visual demonstrations in Fig. 45 and Fig. 46. The postures are decided based on observations of daily office activities, the descriptions in Table 2 instruct how the postures are performed, and the definitions emphasize variations of the spine movement.

Overall 16 people (5 females and 11 males) participated in the experiment. Their age ranges are 23-30 (males) and 22-27 (females), and their height ranges are 170-193cm (males) and 144-171cm (females). Every participant is asked to progress through the postures for two iterations. In every iteration, each posture is performed for three instances, every instance lasts for at least thirty seconds. Between every instance, regardless of the posture class, the participants are asked to step away from the chair and sit down again, so that they do not stay at the same spot every instance.

4.1.4 Posture Detection Algorithm

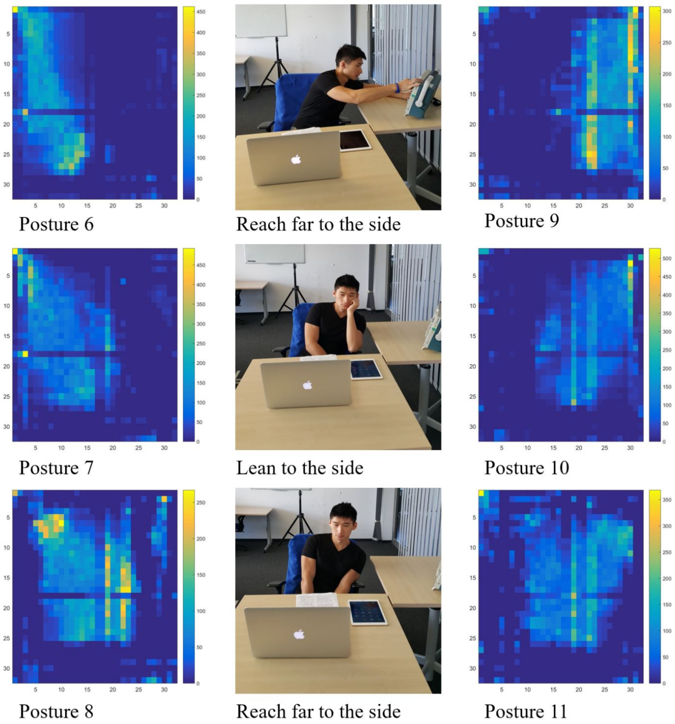
This application shows a more straightforward activity recognition task than the other examples in this work. Essentially, the postures are transient states, and thus there is only the spatial domain for every data sample. In Fig. 45 and Fig. 46, example pressure mapping data are shown side by side with the corresponding postures. For every instance of the participant's sitting, in the time domain 10 frame samples are evenly selected, and manually annotated according to Table 5.

First, for a single participant the data are randomly separated (evenly each posture) into training groups and testing groups, the algorithm is given the ground truth of the training data and will predict the class of the testing data. For every testing frame, the normalized cross-correlation between this frame and every training data frame (templates) is calculated one by one. Every normalized cross-correlation returns a value between -1.0 and 1.0, corresponding to the correlation coefficient between the testing frame and the template. For N templates, every testing frame will have an array of N correlation values corresponding to every template, named as the correlation score vector. K majority voting (in this case, K=20) is used to decide the prediction class. If an alias occurs (two different classes occupy the same number in the K largest values), for each class in the K largest values, the sum of the correlation score values is calculated, and the prediction is determined by the bigger sum. The process from taking a new testing frame to conclude the final posture class takes 0.4-0.5 second in the test with a Dell XPS 9550 (i7 CPU) computer with the algorithm implemented in Python.



a. Posture 1 b. Posture 2 c. Posture 3 d. Posture 4 e. Posture 5

Fig. 45. The examples of Postures 1 to 5 and the corresponding pressure mapping data.



Posture 6 Reach far to the side Posture 9

Posture 7 Lean to the side Posture 10

Posture 8 Reach far to the side Posture 11

Fig. 46. The examples of Postures 6 to 11 and the corresponding pressure mapping data.

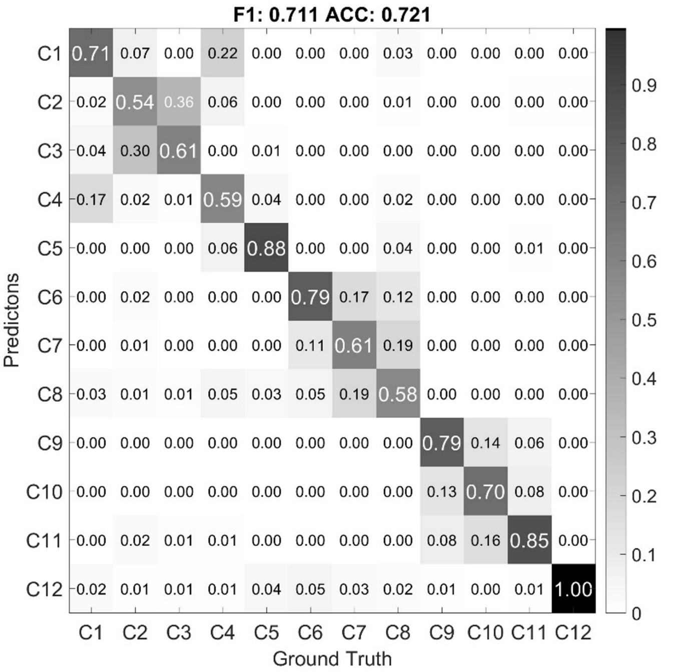


Fig. 47. Confusion matrix of the classification result for 12 classes in the Seat Cover study.

4.1.5 Cross-validation

Leave-one-recording-out cross-validation is performed. The predicted results are compared with the ground truth of the testing data, and visualized in Fig. 47 as a confusion matrix. From the confusion matrix, it can be seen that a vast majority of the data samples are correctly classified. For 12 classes, the random chance level is 8.3%. Therefore, an average accuracy of 72.1% can be considered high above randomness.

C1 and C4 are confused by an average of 20%, from Table 5 and Fig. 45, these two classes are very similar to each other. In the actual experiment, the participants do not subjectively distinguish these two postures in general, except for that the chair back is locked at the vertical position in C1 and unlocked in C4. More miss-classifications happen between C2 and C3. These two postures share the feature of a kyphotic curvature of the spine. It is worth mentioning that C8 and C11, slightly lean to the side, are very well separated from the frontal neutral postures (C1 to C5). Moreover, all the side postures are very well separated from each side. Most miss-classifications only happen within the same side (C6 to C8 and C9 to C11). Therefore, it makes sense to group C1 and C4, C2 and C3 and redraw the confusion matrix as in Fig. 48. From it, a much clearer separation between different classes is presented, and the accuracy progresses to near 80%.

The cross-correlation-majority voting algorithm then is taken as the classification engine, and all the data used in the cross-validation is taken as the training database. The cross-validation result measures how trustworthy the clas-

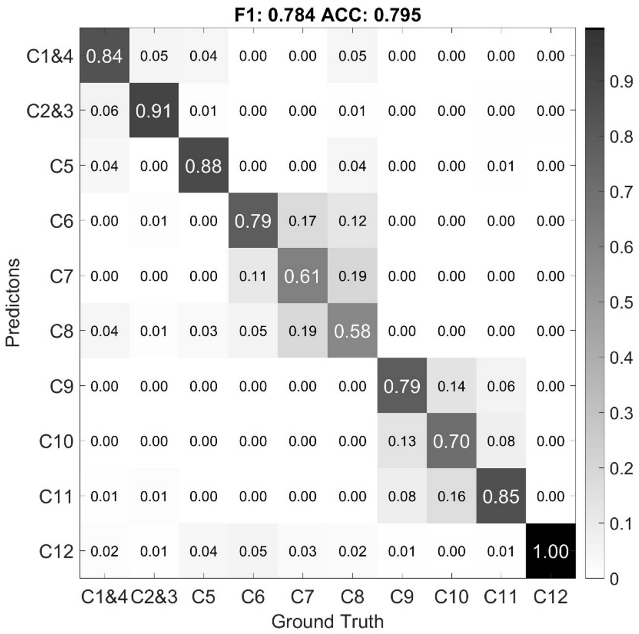


Fig. 48. Confusion matrix of the classification result for 10 classes in the Seat Cover study.

sification engine with the database is in the unsupervised scenario in the following sections.

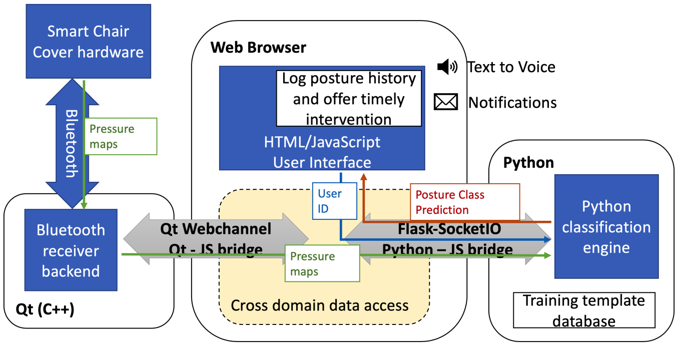


Fig. 49. The structure of the software for real-time feedback and notification.

4.1.6 Real-time Intervention System

The classification engine is implemented in Python, since it is a popular open source platform. The full software stack from Fig. 90 in Chapter 6 is demonstrated in this application. To have a more flexible and fluent user interface implementation, web designing is used. The central part of the software system is implemented in HTML and JavaScript that runs in a browser as illustrated in Fig. 49 which is an implementation of Fig. 95.

The basic structure includes critical data links across different language platforms for bi-directional real-time data transmission (webchannel between Qt and JavaScript, Flask Socket-IO between Python and JavaScript). First, the raw sensor pressure mapping data from the smart blanket is received by the Bluetooth receiver executable program, and forwarded to the JavaScript program. The JavaScript program can visualize the pressure mapping data, but more importantly, it relays the data to the Python classification engine. If the system is shared by different users, the user can also select his or her ID from the Web browser, and the ID is also sent to the Python engine. The classification program then predicts the current posture using the classification engine, and sends the prediction result back to the JavaScript program. This process can operate on a 1-second cycle, giving a fine time grain posture information.

The JavaScript program then keeps the history of the user's posture of the day, and offers timely interventions via means of sound and desktop notifications. For the scope of this work, the intervention is decided on two criteria:

1) If the user has been in a bad posture (out of C5, C6, C7, C10 and C11) for more than 1 minute.
2) If the user has been in the same posture for the majority (over 80%) of the past 10 min.

This decentralized design makes it easier for future modifications. The classification engine can be changed into other methods that are implemented in Python without modifying the other components of the software; the intervention decision can be easily modified within the JavaScript program for other orthopedic or clinical ergonomic opinions.

4.1.7 Conclusion

Through this empirical study, TPM is an ideal option for smart furnitures as force sensitive furniture covers. Apart from this posture monitoring application, demonstrators with car-seat to detect driver posture and body-shape identification have been developed; a couch cover with a force-touch wireless trackpad has also been made.

Another major contribution of this application is to showcase the entire heterogeneous software stack running in real-time, which includes the low-level, hardware interface, middle-level data processing and classification, and the upper-level user interface and feedback.

4.2 Robot Skin/Clothing for Sensing Emotional Touches

Previously, we have used the TPM sensor to detect control gestures [116]. This empirical study evaluates the possibility of using TPM sensors as robotic skin or clothing, to detect emotional-related gestures for human-robot interactions. It is a cross-over between wearable and object-based applications. [6]

The sensor covers a $20 - by - 20cm^2$ area with 400 sensitive points and samples at 50 Hz per point. Seven gestures are defined inspired by the social and emotional interactions of typical people to people or pet scenarios. Two groups of mutually blinded experiments are conducted, involving 29 participants in total. The data processing algorithm first reduces the spatial complexity to frame descriptors, and temporal features are calculated through basic statistical representations and wavelet analysis. Various classifiers are evaluated and the feature calculation algorithms are analyzed in detail to determine each stage and segments' contribution. The best performing feature-classifier combination can recognize the gestures with 93.3% accuracy from strangers.

4.2.1 Study Background

Tactile sensors are an essential sensory in robotics, since they contribute largely as the synthetic counterpart of biological skins on humans and other animals. They are crucial in providing control feedback for safely and securely grasping and manipulating objects [117] [118] [119].

However, a majority portion of biological skins are not sensitive enough, or evolved for precise force sensing and localization for assisting controls [120]. In nature, bodily contact is an important aspect of emotional communication between humans and animals [102]. Studies have shown that body movements are specific for certain emotions [121]. As pointed out in [119], most tactile sensors are based on piezoresistive materials, which have the problem of hysteresis and poor linearity. However, for sensing emotional touches, precision tactile sensing is not necessary.

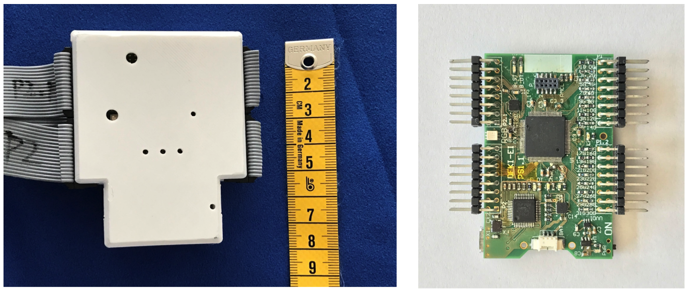
In recent years, the focus on robotics research has evolved from precise and delicate movements to perform various tasks, to a more in-depth communication between human and robotic interactions. In [122], a humanoid robot WE-4RII can effectively express seven emotion patterns with body language. Touch is fundamental in human-human interaction and as robots enter human domains, touch becomes increasingly important also in human-robot interaction (HRI). In recent years, several approaches have been developed to whole-body tactile sensing for robots, e.g., for the iCub [123], [124] or the HPR-2 [125] robots. These systems are cell-based, where each cell comprises a small circuit board holding necessary sensors and electronics and, while presenting excellent sensing capacity, they constitute a relatively hard surface with limited flexibility.

In a recent study comprising 64 participants communicating emotions to an Aldebaran Nao robot using touch, people interacted for longer time when the robot was dressed in a textile suite [126], compared to a standard Nao with a hard plastic surface. These results indicate that the surface material of the robot may be significant for extending and directing tactile HRI. This application is thus inspired by these results, investigating the use of touch-sensitive smart textiles TPM as a potential alternative to cell based-robot skins.

One informative aspect of tactile HRI is the type of touch [126]. In [127], individually designed 56 capacitive sensors are installed in a toy bear to detect affection-related touch. The data processing algorithm relies on signal features such as amplitude and base frequency from all the sensors. In [128], a touch sensing furry toy is developed with a combination of conductive fur touch sensors and piezoresistive touch-localization sensors. Using statistical features of the signal and random forests classifier, the prototype recognizes 8 gestures with a 86% accuracy. This application investigates the capacity of distinguishing between seven different types of touch, listed in Table 6, based on sensor data gathered from the smart fabric.

TABLE 6
Robot Skin Study: Prediction Class of Gestures Definition

Index	Gesture	Details
P1	grab	whole hand grabbing the dummy's arm
P2	poke	with one finger, quick and forceful action
P3	press	with multiple finger tips, slow action
P4	push	whole hand including palm, slow action
P5	scratch	with multiple finger tips, quick repeating actions
P6	pinch	on a small area, forceful action
P7	stroke	with multiple fingers, gentle repeating actions



a. Electronics module with enclosure

b. Main PCB

Fig. 50. Robot skin study: electronic hardware

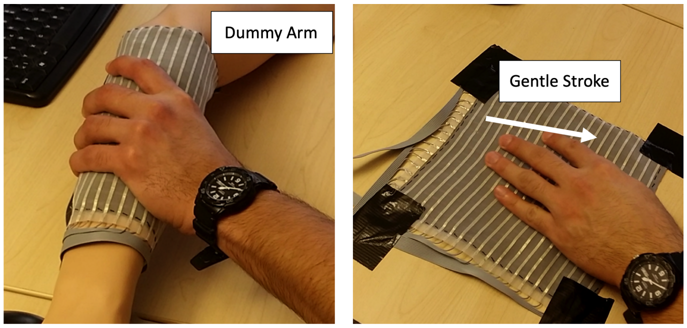
4.2.2 Hardware Requirement

In this study, the hardware Version M-1.1 from Table 17 in Chapter 2 is used (Fig. 50). A 20 × 20 square TPM sensor is used, with 1cm pitch, 12-bit ADC resolution and 50 frames per second sampling rate.

4.2.3 Experiment

Seven gestures are defined as listed in Table 6. Only in P1 - grab, the sensing skin is wrapped around a dummy arm (Fig. 51 a), and for the rest of the gestures, the sensor is

[6] This section is based on the publication [62]. As the leading author of the publication, I have only included my own work; unless other authors' contribution is an integral part of explaining my work, in which case it will be explicitly stated. Since the published work has gone through extensive wording, some sentences and passages have been quoted verbatim from the said source.



a. P1 Grab b. P7 Stroke

Fig. 51. Robot skin study: experiment setup and gesture example a. P1 Grab and b. P7 Stroke

fixed on a flat surface (Fig. 51 b). This setup should be seen as a pre-stage to mounting the fabric as the skin for a robot, constituting a robot-agnostic baseline without the full complexity of uneven surfaces and the robot's own motions. The focus of the evaluation is to investigate if the sensor can be used to physically distinguish these expressive gestures.

4.2.4 Dataset

Two groups of data are recorded using the same sensor setup. The two groups are conducted by different persons, and they do not have knowledge of how the other group recorded the experiment. The two groups are therefore mutually blinded, including the experiment conductors. It ends up that in Group A, the conductors instructed the participants to perform *P5 Scratch* and *P7 Stroke* as a quick burst of three repetitions in a single action; while in Group B, the participants only perform the gestures continuously for every action.

- Group A: 24 people, 2 recordings per person. Every gesture is repeated 16 times. During the recording, the participants are asked to use both their right and left hands to perform the gestures equally in multiple repetitions. The participant pool consists of 12 males and 12 females. The hand size is assumed to be a contributing influence factor in this experiment. The hand sizes of the males range from 17.5 to 20 cm, and 17 - 18.7 cm for the female participants. There is one left-handed participant in each gender.
- Group B: 5 people, single recording per person. Every gesture is repeated 16 times. The participants use only one hand of their choice to perform all the gestures. There are 2 female and 3 male participants. Their hand size ranges from 16.5 cm to 21.5 cm.

Group A is used for the majority of data analysis and algorithm evaluation. Group B serves as a reference to see how the algorithm is influenced by a completely new setup with different environment and directing persons.

The participants are given the literal descriptions of the seven gestures only as in Table 6, without visual guidance. They are instructed that they can relate each gesture to suitable emotions, however, the experiment instruction does not prescribe specific emotion-gesture bindings. The data is manually annotated to separate every gesture action by the experiment conductor. Overall, 5376 gesture actions are recorded in Group A and 560 in Group B.

4.2.5 Evaluation

Every *frame* is up-sampled from 20-by-20 to 40-by-40 to increase the spatial resolution with bicubic interpolation. To extract information, first, the 2D spatial data is reduced into limited information as *frame descriptors*. The following descriptors are calculated from every frame:

- D_1 mean value of all pixels' value
- D_2 maximum value of all pixels' value
- D_3 standard deviation of all pixels' value
- D_4 distance from the center of gravity to the frame center
- D_5 distance from the maximum point to the frame center
- D_6 the number of pixels that has higher value than a threshold
  $(threshold = mean + standard\ deviation)$

The frame descriptors, therefore, reduce the 2-dimensional information to 6 temporal vectors. For example, if a gesture lasts 3 seconds, a stream of 150 frames (each 20-by-20) are generated by the tactile sensor, and six arrays, each 150 in length, are abstracted as the temporal sequences of frame descriptors. D_1 and D_4 describes the intensity and location of the center of the pressure; D_2 and D_5 offers information of the highest pressure point; D_3 describes how scattered the pressure is on the surface; D_6 describes the surface area of the contact.

The experiment data is manually segmented by the experiment conductor roughly before and after the contact. To make sure the data samples cover exactly the contact time, a cut-off threshold is defined:

$$Threshold_{cutoff} = min(D_1) + (max(D_1) - min(D_1)) \times 10\%$$

The samples before the first t_1 when $D_1(t_1) > Threshold_{cutoff}$, and after the last t_2 when $D_1(t_2) > Threshold_{cutoff}$ are removed.

Fig. 52 visualizes the temporal sequences of *frame descriptors* from different classes. One action of each gesture from every person is randomly selected. For comparison purposes, the sequences are scaled to the same 400-sample window using linear interpolation; the original data sequences have different lengths. The next step is to extract features to distinguish between different classes. For example, in subplot $D_6 - P2$, *P2 Poke* has significantly smaller D_6 than the other gestures; *P5 Scratch* and *P7 Stroke* have distinct higher frequency movements than the other gestures in all the frame descriptors; *P1 Grab* and *P4 Push* has higher average value in D_1 and D_2 than the other gestures.

Two types of features are used from the temporal sequences $D_x(t)$ of size T: basic statistic representations, and wavelet analysis.

The 5 basic statistic representations are:

- the average value in the window

$$mean(D_x) = \sum_{t=1}^{T} D_x(t)$$

- the standard deviation in the window, defined as

$$std(D_x) = \sqrt{\frac{1}{T-1} \sum_{t=1}^{T} |D_x(t) - mean(D_x)|^2}$$

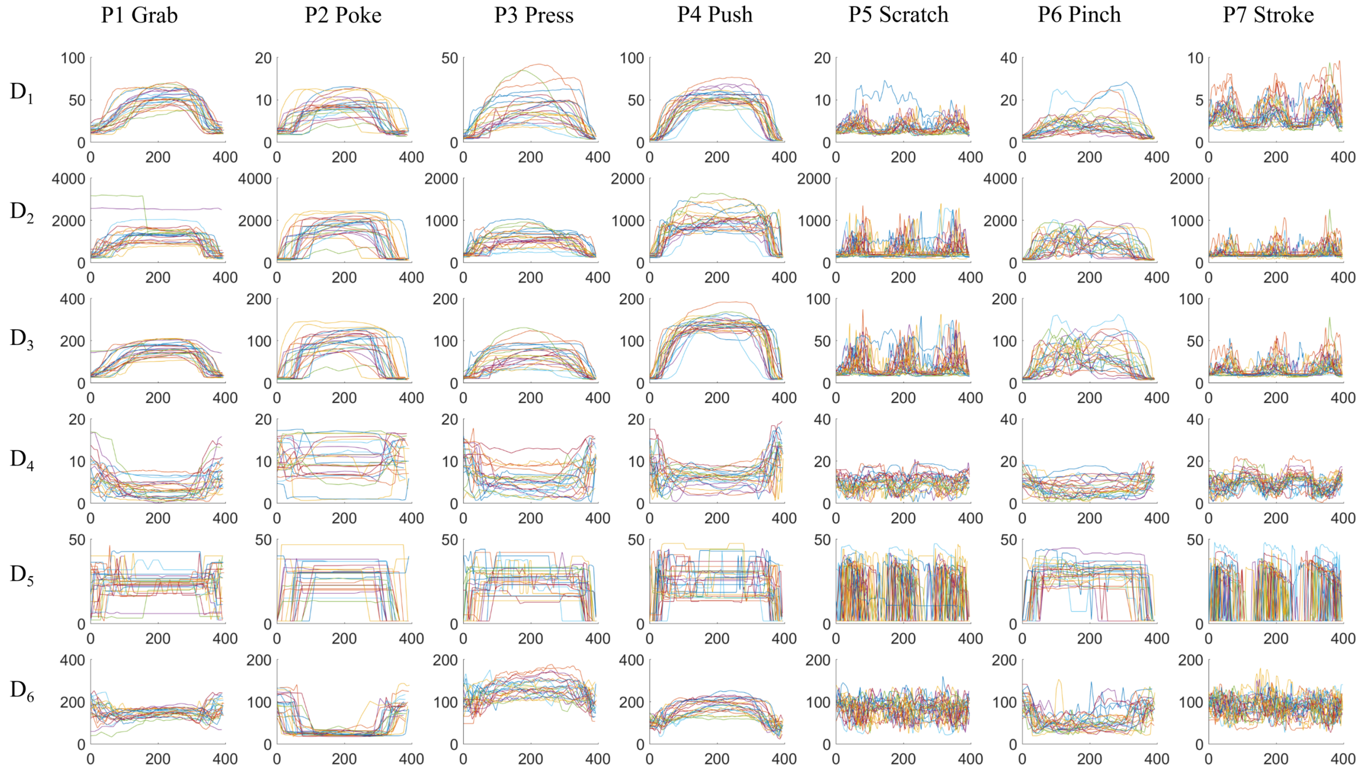


Fig. 52. Comparison of the temporal sequence of frame descriptors from each class. Each curve is from a different person within Group A. The x-axis are sample in the temporal domain, the y-axis are scales of the values from different frame descriptors. The original data has different temporal length; in this plot they are scaled to the same 400-sample window using linear interpolation for visual comparison.

- the absolute range of the sequence: $max(D_x) - min(D_x)$
- the kurtosis of the sequence, which measures how outlier-prone the data is. defined as:

$$kur(D_x) = \frac{\frac{1}{T}\sum_{t=1}^{T}(D_x(t) - mean(D_x))^4}{(std(D_x))^4}$$

- the skewness of the sequence, which is the asymmetry measurement of the data around the mean value. It is defined as:

$$skw(D_x) = \frac{\frac{1}{T}\sum_{t=1}^{T}(D_x(t) - mean(D_x))^3}{(std(D_x))^3}$$

These features would describe the distribution of the temporal sequence, and are commonly used in statistic analysis.

The temporal features are not to be confused with the frame descriptors. For example, $D_3(t)$ is the standard deviation of all the pixels from a frame at sample t at a particular point in time; D_3 is the sequence of the standard deviation of each frame within the window; $std(D_3)$ is the standard deviation of all the 2D standard deviation frame descriptors within the window. Frame descriptors reduce the spatial domain data into limited measures, and the temporal features further reduce the temporal domain information. For one window of gesture, 6 sequences of frame descriptors are calculated, which results in overall 30 basic features.

Wavelet transform converts frequency-related information from the data into features. It offers frequency and temporal localization of the target signal.

The $D_x(t)$ signal of length T is padded with its boundary values with a padding size of $T/2$: before D_x, $D_x(1)$ are inserted $T/2$ times repeatedly, and at the end of D_x, $D_x(T)$ are inserted $T/2$ times. Then the padded signal $D'_x(t), t \in [1, T]$ is multiplied with a symmetric Hamming window $w(t), t \in [1, T]$:

$$D''_x(t) = D'_x(t) \times w(t),$$

$$w(t) = 0.54 - 0.46cos\left(\frac{2\pi t}{T-1}\right), t \in [1, T]$$

Fig. 53 visualizes the boundary padding and hamming window process. Padding and window functions are typical techniques in signal processing to remove the influence of the sampling window.

Fast wavelet transform implemented by the LTFAT toolbox [129] is used, which follows Mallat's basic filterbank algorithm for discrete wavelet transform [130]. Fig. 54 offers an illustration and comparison of two different source signals going through 5-level and 10-level filterbanks. Essentially, each filterbank iteration calculates a vector of coefficients as results. The calculation uses a mother wavelet, which is scaled and shifted to provide frequency variance and temporal localization. In this study, the Daubechies 8 (db8) wavelet is used as the motherwavelet [131]. Other standard mother wavelets can also be used in this process; however, once chosen, the mother wavelet should not be

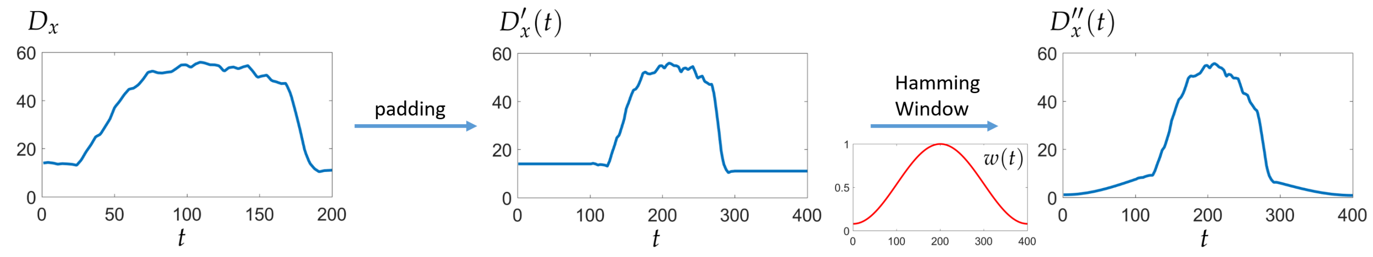


Fig. 53. Visualization of the boundary padding and hamming window pre-processing before wavelet transform

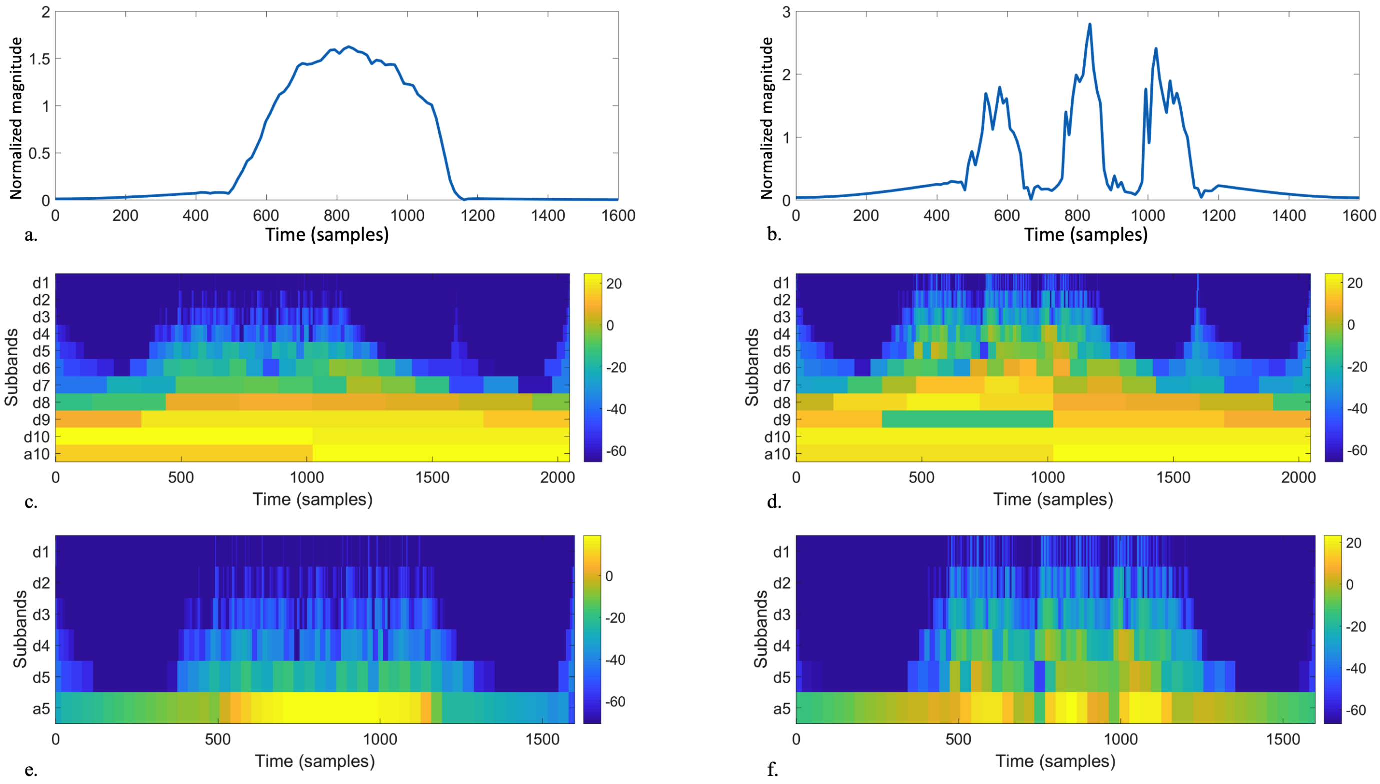


Fig. 54. Visual illustration of deriving wavelet features. (a) and (b) are the source signal; (c) - (f) are scaleograms of the wavelet transform coefficients; ;(c) and (e) are the results from (a); (d) and (f) are the results from (b); (c) and (d) are the results of 10 filterbank iterations; (e) and (f) are 5 filterbank iterations.

changed because the wavelet transform will have different references. Higher iteration targets the higher frequency and finer temporal localization, which results in a longer vector of coefficients. For example, assume the sample frequency is f, with $T = 1600$ samples, a five-level filterbank discrete wavelet transform as in Figure 54 (e) and (f) results in wavelet coefficients as shown in Table 7. With the highest level defined as J, the coefficients are

$$C(D_x) = \begin{pmatrix} C_{dj}(D_x), j \in [1, J] \\ C_{aj}, j = J \end{pmatrix}$$

These coefficients are unique to the specific signal, as they can be used to reconstruct the signal by the inverse wavelet transform. Each subband contains temporal localization of the corresponding frequency. Therefore their distribution information can be used as unique features.

For the last three subbands (in the example in Table 7, level 3-5, subband d3, d4, d5, a5), we calculate the $mean(C_n(D_x)), n \in \{aJ, dJ, d(J-1), d(J-2), d(J-3)\}$ as the first four wavelet features. For the lower levels $C_dj(D_x), j \in [1, J-4]$ which have significantly finer temporal granularity and bigger number of coefficients, the following features are calculated to describe the distribution information:

- $mean(C_n(D_x))$
- $max(C_n(D_x)) - min(C_n(D_x))$
- $std(C_n(D_x))$
- $kur(C_n(D_x))$
- $skw(C_n(D_x))$

Therefore, for every sequence of frame descriptors D_x for $J = 5$, 14 features are calculated from the wavelet transform; for $J = 10$, 39 features are calculated.

4.2.6 Evaluation

The following classifiers from the Matlab Classification Learner app are used for comparison:

TABLE 7
Robot Skin Study: frequency range and coefficients of the discrete wavelet transform.

Subband	frequency in fn	frequency in Hz	coefficients $C(D_x)$
d1	fn/2 - fn	25 - 50	$C_{d1}(D_x, n), n \in [1, 800]$
d2	fn/4 - fn/2	12.5 - 25	$C_{d2}(D_x, n), n \in [1, 400]$
d3	fn/8 - fn/2	6.25 - 12.5	$C_{d3}(D_x, n), n \in [1, 200]$
d4	fn/16 - fn/8	3.125 - 6.25	$C_{d4}(D_x, n), n \in [1, 100]$
d5	fn/32 - fn/16	1.5625 - 3.125	$C_{d5}(D_x, n), n \in [1, 50]$
a5	0 - fn/32	0 - 1.5625	$C_{a5}(D_x, n), n \in [1, 50]$

- Medium Tree (maximum 20 splits decision tree)
- Linear Discriminant Analysis (LDA)
- Support Vector Machine (SVM) with linear kernel
- SVM with quadratic kernel
- K-nearest neighbors (KNN) with $K = 10$
- distance weighted KNN with $K = 10$
- Bagged Trees (random forest bag, with decision tree learners)

For cross-validation, three settings are considered:

- Random cross-validation (K-Fold): the training data and testing data are from the same data set with k-fold cross-validation.
- Leave-one-recording-out: as the data from the same experiment session may exhibit greater similarity, we use separate different sessions from the same person into training and testing data of the classifier.
- Person independent exclusive: the training data and testing data are from two groups of persons; the two groups are mutually exclusive. So that the classifier has no previous knowledge of the person being tested.

TABLE 8
Robot Skin Study: Classification accuracy comparison of different classifiers using basic features.

Classifier	Accuracy
Medium Tree	80.50%
LDA	79.10%
SVM(Linear)	90.80%
SVM(Quadratic)	91.90%
KNN(K=10)	86.20%
Weighted KNN(K=10)	87.00%
Bagged Trees	89.60%

The basic features are considered with all the frame descriptors ($D_x, x \in [1, 6]$). The results are listed in Table 8. The average accuracy of all classifiers is 86.44%, which is well above the chance level of seven classes 14.29%. The best performing classifier is SVM with a quadratic kernel with an accuracy of 91.90%.

Different from the basic features, the amount of wavelet features depends on the filterbank iteration (J) of the discrete wavelet transform. Therefore, the wavelet features are evaluated by trying different filterbank iterations, as shown in Table 9. The obvious trend is that as J increases, the accuracies of all classifiers are increasing. On average, while $J = 6$ and $J = 4$, the results are inferior to the basic features, while $J = 8$ and $J = 10$ yields slightly better results than the basic features most of the classifiers (except for LDA and KNN). Even though there are instances that a higher J yields slightly lower accuracy (for example, Quadratic SVM with $J = 10$ and $J = 8$, it is within the random error range, because every result is from a unique randomly separated 5-fold cross-validation.

Next, both the basic features and wavelet features are combined. From Table 10, all of the results are better than either basic features as in Table 8 and wavelet features from Table 9. For example, with the LDA classifier, basic features and wavelet features ($J = 4$) yield the accuracy of 79.10% and 77.20%; while when both feature sets are combined, the accuracy is improved by 4.50% to 83.60%.

Assuming application is not limited by the computational power, after comparing the contribution of the basic features and wavelet features, basic features and wavelet features ($J = 10$) combined are chosen for the following evaluation.

The six frame descriptors offer information from different angles: D_1 and D_4 describes the average center pressure point by the value and location; D_2 and D_5 are the maximum pressure point; D_3 is the variation of the pressure profile; D_6 measures the pressed area. Here how each frame descriptor contributes to the classification result is discussed. Table 11 shows the results of cross-validations with separate frame descriptors. The results are based on all basic features and wavelet features with $J = 10$. Different descriptors contribute differently in combination with different classifiers. For example, $[D_1, D_4]$ give less accuracy than D_6 with the KNN classifiers, but more with Quadratic SVM. Overall, the combination of all frame descriptors $[D_{1-6}]$ offer superior result than any of the individual descriptors. This means that all of the descriptors make positive contributions to the classification result.

4.2.7 Result and Discussion

Base on the previous analysis, all the frame descriptors, with both basic features and wavelet features ($J = 10$) are taken into consideration, because all analysis findings indicate that all of the factors contribute positively to the classification result. The support vector machine classifier with a quadratic kernel is chosen as it offers the best accuracy.

The first result in Figure 55a is from the randomly separated 5-fold cross-validation from all participants in Group A, it is essentially the confusion matrix of the corresponding result from Table 9. The values in the matrix are ratios of the current prediction in the overall ground truth of its class; on the diagonal, the values are the true positive ratio of each class. The F1 score is calculated as the harmonic mean of the average precision and recall of all the classes; the ACC score is the accuracy, which is the average true positive rate.

TABLE 9
Robot Skin Study: Classification accuracy comparison of different classifiers using wavelet features with varied levels of filterbanks.

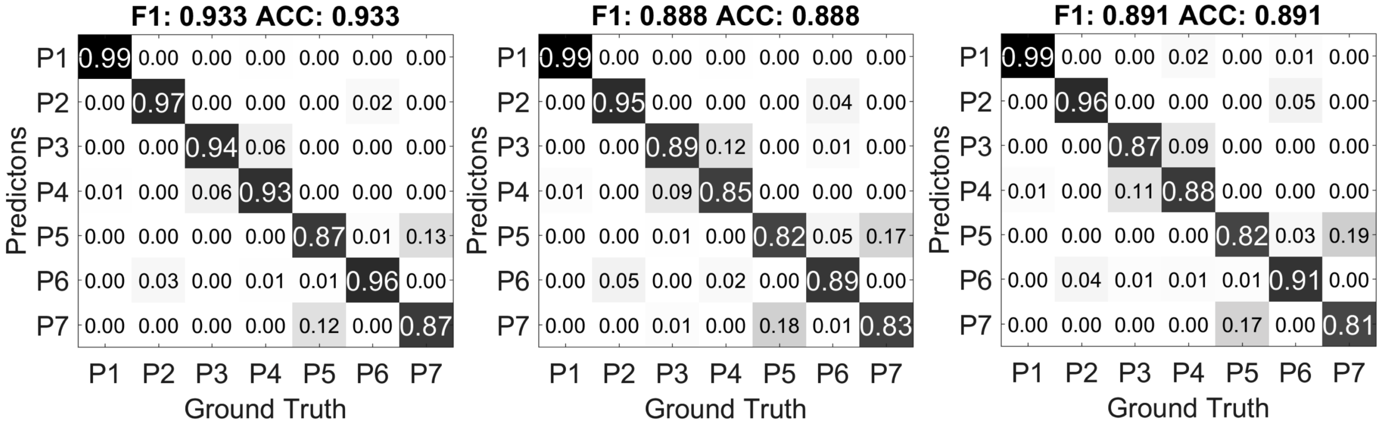
Classifier	ACC(J=10)	ACC(J=8)	ACC(J=6)	ACC(J=4)
Medium Tree	82.10%	79.70%	78.10%	77.70%
LDA	75.00%	77.90%	74.10%	77.20%
SVM(Linear)	91.70%	91.10%	88.50%	87.10%
SVM(Quadratic)	92.30%	92.40%	89.70%	87.80%
KNN(K=10)	85.40%	84.30%	77.50%	77.00%
Weighted KNN(K=10)	86.10%	84.60%	77.80%	77.60%
Bagged Trees	91.60%	91.00%	88.00%	86.60%

TABLE 10
Robot Skin Study: Classification accuracy comparison of using different classifiers with both basic and wavelet features (for wavelet features, with varying J).

Classifier	ACC(J=10)	ACC(J=8)	ACC(J=6)	ACC(J=4)
Medium Tree	83.80%	83.70%	83.20%	81.20%
LDA	80.20%	82.30%	81.20%	83.60%
SVM(Linear)	92.80%	92.70%	92.10%	91.70%
SVM(Quadratic)	93.30%	93.60%	92.80%	92.20%
KNN(K=10)	87.10%	86.90%	84.00%	86.00%
Weighted KNN(K=10)	87.70%	87.20%	83.90%	86.00%
Bagged Trees	92.40%	92.30%	91.50%	90.80%

TABLE 11
Robot Skin Study: Comparison of the contribution of different frame descriptors, the features used are both basic and wavelet ($J = 10$) features.

Classifier	ACC(D_1, D_4)	ACC(D_2, D_5)	ACC(D_3)	ACC(D_6)	ACC(D_{1-6})
Medium Tree	70.90%	70.50%	71.40%	74.30%	83.80%
LDA	68.80%	67.70%	64.50%	73.00%	80.20%
SVM(Linear)	85.80%	83.40%	85.20%	85.00%	92.80%
SVM(Quadratic)	88.00%	84.60%	86.10%	85.30%	93.30%
KNN(K=10)	74.90%	73.70%	73.10%	78.80%	87.10%
Weighted KNN(K=10)	75.50%	74.70%	74.10%	79.10%	87.70%
Bagged Trees	84.90%	82.50%	83.10%	85.00%	92.40%



F1: 0.933 ACC: 0.933

Predictons \ Ground Truth	P1	P2	P3	P4	P5	P6	P7
P1	0.99	0.00	0.00	0.00	0.00	0.00	0.00
P2	0.00	0.97	0.00	0.00	0.00	0.02	0.00
P3	0.00	0.00	0.94	0.06	0.00	0.00	0.00
P4	0.01	0.00	0.06	0.93	0.00	0.00	0.00
P5	0.00	0.00	0.00	0.00	0.87	0.01	0.13
P6	0.00	0.03	0.00	0.01	0.01	0.96	0.00
P7	0.00	0.00	0.00	0.00	0.12	0.00	0.87

a. Group A Cross-validation

F1: 0.888 ACC: 0.888

Predictons \ Ground Truth	P1	P2	P3	P4	P5	P6	P7
P1	0.99	0.00	0.00	0.00	0.00	0.00	0.00
P2	0.00	0.95	0.00	0.00	0.00	0.04	0.00
P3	0.00	0.00	0.89	0.12	0.00	0.01	0.00
P4	0.01	0.00	0.09	0.85	0.00	0.00	0.00
P5	0.00	0.00	0.01	0.00	0.82	0.05	0.17
P6	0.00	0.05	0.00	0.02	0.00	0.89	0.00
P7	0.00	0.00	0.01	0.00	0.18	0.01	0.83

b. Leave-1-Recording-Out

F1: 0.891 ACC: 0.891

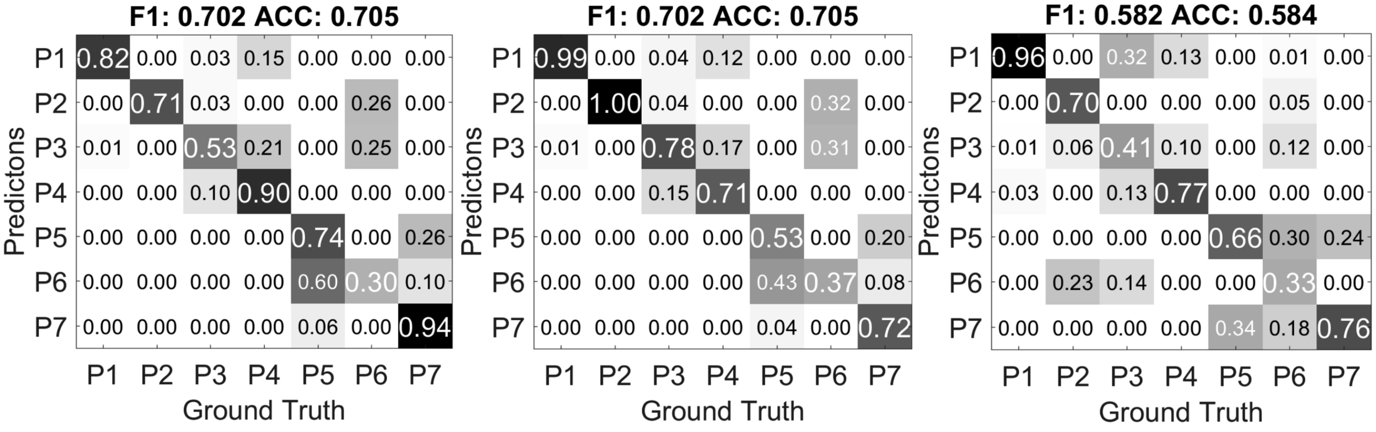
Predictons \ Ground Truth	P1	P2	P3	P4	P5	P6	P7
P1	0.99	0.00	0.00	0.02	0.00	0.01	0.00
P2	0.00	0.96	0.00	0.00	0.00	0.05	0.00
P3	0.00	0.00	0.87	0.09	0.00	0.00	0.00
P4	0.01	0.00	0.11	0.88	0.00	0.00	0.00
P5	0.00	0.00	0.00	0.00	0.82	0.03	0.19
P6	0.00	0.04	0.01	0.01	0.01	0.91	0.00
P7	0.00	0.00	0.00	0.00	0.17	0.00	0.81

c. Person Independent Exclusive

Fig. 55. Robot skin study: confusion matrix of the cross-validation result on Group A dataset.

Data recorded in the same session may possess greater similarity than another session from the same person. In Group A, every participant attended two recording sessions on different days. These sessions are separated into two sets, each set contains one session from all participants. One set is used as training, and the other as testing; the process is then reversed as the training and testing sets are exchanged. The confusion matrix in Figure 55(b) is the average of both results.

Next, the study evaluates how well the classifier can predict on a stranger’s data. The 24 participants are randomly separated into 4 parties, each 6 people. Then every party is used as the testing data while the other three parties are the training data. This process is repeated 4 times so that every party is used as testing data once. The result in Figure 55(c) is the average confusion matrix of the four repetitions.



a. Person Independent Exclusive Group B

b. Training: Group A Testing: Group B

c. Training: Group B Testing: Group A

Fig. 56. a. Confusion matrix of the person independent exclusive case of Group B, b. and c. Confusion matrix of the validations using Group A and Group B as training and testing.

All three results show very well separation among all of the classes. Major miss-classification happens between *P3 Press* and *P4 Push*, *P5 scratch* and *P7 stroke*. *Press* and *push* are similar actions, except *push* has greater contact area and generally greater force; *scratch* and *stroke* are both repeating actions, while *scratch* may have smaller area of contact. Overall, the average 88.8% and 89.1% accuracy in leave-1-recording-out and person independent exclusive cases are also well above the random chance level of 14.3%.

For the mutually blinded experiment Group B, the confusion matrix of person independent exclusive validation is shown in Figure 56a. Except for the miss classifications observed in Group A, more data from *P6 Pinch* are classified as *P3 Press*.

Then a classifier is trained with the feature data from Group A, and tested with the data from Group B. Figure 56b shows the confusion matrix result. Compared to the confusion matrix from Figure 55c in the person independent exclusive case, *CM A-B* has near 19% drop of accuracy and F1-score. This means the mutually blinded experiment setting does decrease the recognition rate of the gesture recognition approach. Notably, 12% of *P4 Push* gestures are classified as *P1 Grab*, but most of the *P1 Grab* gestures are correctly classified. The mutual miss classifications between the pairs of *P3 Press* and *P4 Push*, *P5 scratch* and *P7 stroke*, which are observed in the cross-validation of Group A, are further increased. Most interestingly, in the Group A only cross-validation, *P6 Pinch* is clearly distinguishable from the other classes, while in *CM A-B*, it is largely miss classified into *P2 Poke* and *P3 Press*. Also gestures from *P5 Scratch* is miss-classified as *P6 Pinch*. The mutually blind experiment setting could mainly cause this.

Figure 56c shows the result of using Group B as training, and Group A as testing (*CM B-A*). As the accuracy decreases, the classifier for *CM B-A* has only 560 samples as training data, while in *CM B-A*, the classifier is trained with 5376 samples. And also Group B contains only the data from one hand of each person, while in Group A, both hands are used for recording the data. The major miss-classification caused by *P6 Pinch* also exists in *CM B-A*, further suggesting the mutually blind setting is the underlying cause.

Overall, the comparison of *CM A-B*, *CM B-A*, and the cross-validations within Group A and Group B concludes that (1) a completely blinded setting regarding experiment and instruction can make a difference in recognition results; (2) more training data can improve the accuracy for person independent cases.

4.2.8 Conclusion

This application developed a textile robot skin prototype from tactile pressure mapping sensors and algorithms to investigate Human-Robot interfaces through various kinds of touch, which is still uncharted territory. The textile touch sensing skin is soft and the feel is close to clothing materials. In a small region, it can detect different modes of touch gestures with the same skin patch through our evaluation.

The data processing and feature mining algorithms also serve as a detailed analysis with example data on the wavelet analysis features first introduced in Chapter 8. The contribution of each frame descriptor and each feature set is evaluated, with different classifiers. The overall result is that all of the frame descriptors with all of the feature sets provide the optimal classification result of 93% with a support vector machine classifier with a quadratic kernel. The contribution breakdown also helps further optimizing computational complexity. For example, with only the basic features on all the frame descriptors, the accuracy drops less than 2% from the optimal accuracy; with only D_1 and D_4 frame descriptors, the accuracy only drops 5%.

The increased miss-classification in the exclusive person independent settings (Fig. 55c and Figure 56a) and the mutual blind experiment settings (Figure 56) evaluation reveals interesting aspects when it comes to strangers, which is similar to what may happen between human-human interactions: for example, someone's normal pet on the shoulder may feel too heavy for some certain people. This opens the possibility to progressively improve tactile communication through learning or even differentiate between users of the robot through social-purposed touch sensing.

4.3 Student Developer Workshop Projects

A couple of researchers with the knowledge and access to TPM technology only has limited time to explore empirical applications. Therefore a student workshop is held, with existing hardware version M-1.1 from Table 17 in Chapter 2, and a collection of software functions as a sandbox implemented by Ankur Mawandia in Matlab. [7] [8]

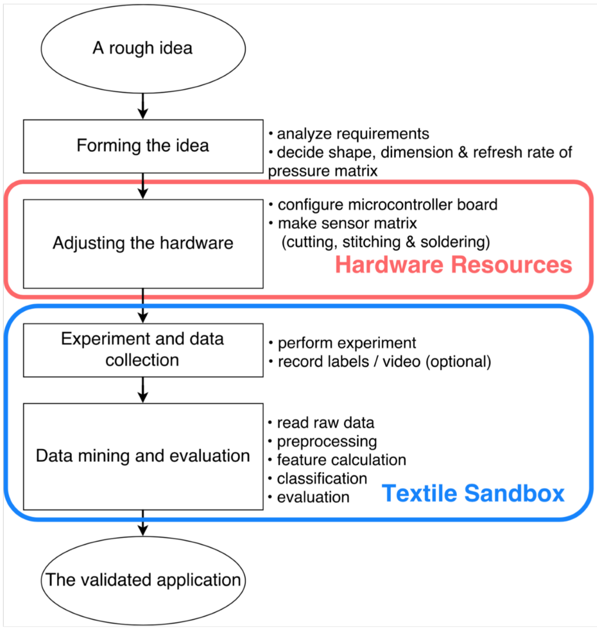

A rough idea
Forming the idea
• analyze requirements
• decide shape, dimension & refresh rate of pressure matrix
Adjusting the hardware
• configure microcontroller board
• make sensor matrix
(cutting, stitching & soldering)
Hardware Resources
Experiment and data collection
• perform experiment
• record labels / video (optional)
Data mining and evaluation
• read raw data
• preprocessing
• feature calculation
• classification
• evaluation
Textile Sandbox
The validated application

Fig. 57. Typical procedures in new application exploration with textile pressure sensing matrix: the “Textile Sandbox” serves as the general framework for data recording and mining.

4.3.1 *the Matlab Textile-Sandbox*

The typical workflow of exploring an application with the Textile-Sandbox is shown in Fig. 57. The design guidelines of the sandbox are:

- Per step support: Every step in the workflow should be provided with software tools.
- Configurable: Application-specific parameters such as matrix size, refresh rate, etc. can be configured by an editable .csv file.
- Fast kickstart: Together with the framework, a compact example of an annotated dataset and configurations is provided to execute the data processing chain by calling a single function and a few more mouse clicks.

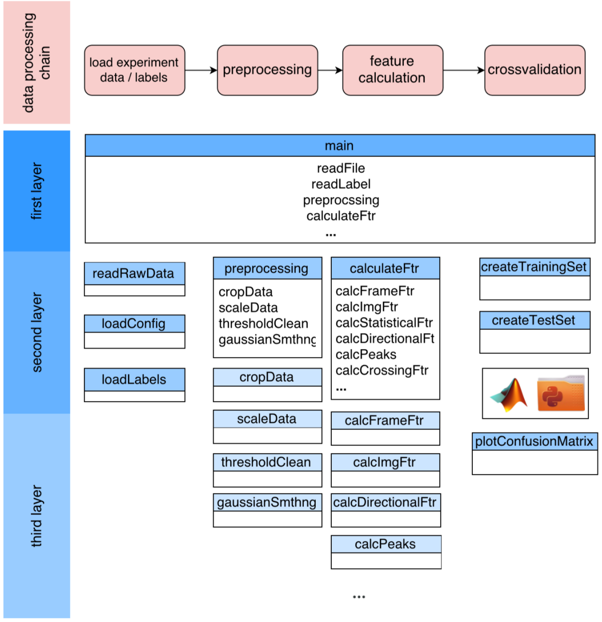


Fig. 58. Data processing flow and corresponding design in Textile-Sandbox.

- Modular design: The Sandbox is divided into three layers of modules as in Fig. 58. Users can sequentially execute each module and validate its output. The intermediate outcome of former steps is automatically saved, so that the user can resume the data processing at any step in the progress.
- Documented: An online ‘help’ documentation is created about resource downloading, experiment design tutorials and how to use the labelling tools and data mining tool. It is indexed and can be searched upon.

Two annotation tools are included in the Sandbox. The first one is a light-weight online tool for the experiment participants to record annotation while performing the experiment; the second is an offline tool that plots the recorded data to manually input annotation. The second tool is meant for experiments where the participants perform some process which may include combinations of different activities, and is difficult to control while conducting the experiment.

The data mining tool has similar functionality as the Tier 1-3 in Chapter 8 but with more simplified functions to make it easier for beginner developers to grasp the concept of processing TPM imagery. The first part of the data mining tool is a set of preprocessing methods such as DC removal, upscaling, filtering to enhance the data quality within each frame. The second part is the feature calculation. 21 basic features are provided but this part also supports custom features. The 21 basic features are:

Statistical features from the time series of frame descriptors (10 features): two descriptors are calculated from each frame: the sum of all pixels and the number of pixels with higher values than a calculated threshold. From the time series formed by these two descriptors within the event, the maximum, minimum, mean, number of peaks and number of mean-crossing are calculated.

[7] This section is based on the publication [132] in which I am the leading author. The Matlab ‘sandbox’ from Section 4.3.1 is implemented by co-author Ankur Mawandia and the student workshop is lead by co-author Prof. Jingyuan Cheng. Since the published work has gone through extensive wording, some sentences and passages have been quoted verbatim from the said source.

[8] The developer user study is carried out by Mawandia, Ankur and supervised by Prof. Jingyuan Cheng, in the Technical University of Braunschweig. Mawandia implemented a software sandbox in MATLAB based on previous publications from me and my co-authors. And student participants use the sandbox to evaluate TPM sensing in their own proposed applications. I also supported with advising on hardware and software details during the study and the paper writing after the study.

TABLE 12
Time distribution in application exploration with Textile-Sandbox

Project	Matrix	Classes	Participant×Repetitions	Evaluation[1]	Accuracy
SpyOnMe	32 × 8 2cm pitch	6 [2]	3 × 5	Bagged Trees 8-fold	91%
Win Your Heart	16 × 16 2cm pitch	7 [3]	1 × 10	Random Forest 10-fold	84%
Pressure Password	16 × 16 1cm pitch	5 [4]	2 × 10	SVM 5-fold	76%
Smart Pillow	16 × 16 2cm - 4cm pitches [5]	5 [6]	3 × 10	Bagged Trees 10-fold	87%

[1] all projects used the 21 basic features provided by the Textile-Sandbox.
[2] typing on keyboard, writing with a pen, sketching with a pen, internet surfing, playing computer games, idle, and absence.
[3] scratching, hugging, holding the toy's upper part, holding the lower part, beating, pinching and touching with the face.
[4] five distinct passwords.
[5] the asymmetric pitch is used to accompany the length of the pillow.
[6] 4 sleep positions (supine position , prone position , lying on the left side , lying on the right side) and 1 kneeing posture .

Pressure center shift (4 features): Three frames from each activity event, the first, the last, and the frame with the highest pixel-sum, are selected. The center of weight [x,y] of these frames are calculated. The difference of [x,y] between the first and the last frame, and between first and the highest pixel-sum frame, are considered as another 4 features.

Image descriptors the average frame (7 features): Each pixel in the average frame is the average of the pixel location within the event period. The 7 Zernike image moments [133] of this frame are calculated as features.

The third part performs cross-validation using the features from the second part, with the Matlab Classification Learner.

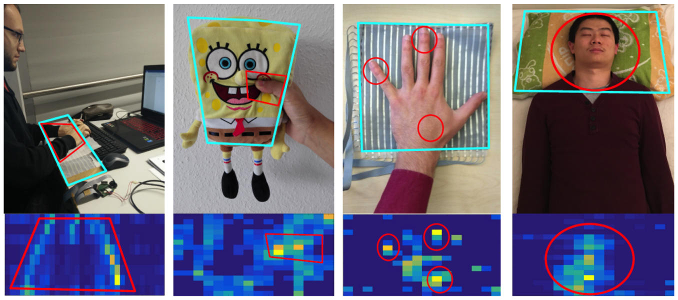

a. Spy On Me b. Win Your Heart c. Pressure Password d. Smart Pillow

Fig. 59. Applications explored using Textile-Sandbox, along with the representing pressure distribution of one selected activity, (a)SpyOnMe (two arms lying on the table while tying on keyboard), (b)Win Your Heart (grabbing with thumb), (c)Pressure Password (pressing with the middle and the little fingers), (d)Smart Pillow (supine position).

4.3.2 Student Workshop Results

A workshop for master students major in Computer Science or System Techniques is held, where the students develop their own application from scratch. From 10 participants, only 3 had background knowledge on ubiquitous computing through some earlier lectures. Four groups were formed voluntarily. All the groups managed to individually propose and explore one application within only 40 hours (The applications are shown in Fig. 59, detailed time distribution is listed in Table 13). All applications are based on general ready-to-use electronics, version M-2.1.

The details of every application are written in [132]. Table 4.3.1 briefly summarizes the evaluation process and results of every application. Briefly, the motivation for every application are:

SpyOnMe: activity monitoring at workplace. Detection of typical activities like typing on a keyboard, browsing the web or away from work can give insights into the person's role, his/her methods and performance at workplaces. SpyOnMe is proposed, as an non-intrusive method to monitor workspace activities based on pressure between forearms and the desk.

TABLE 13
Time distribution in application exploration with Textile-Sandbox

Task	Time Spent
Introduction lecture*	6 hr
Software practice with existing dataset	4 hr
Propose an application	3 hr
Making matrix	4 hr
Data recording and evaluation with Textile-Sandbox	20 hr
Presentation on the explored applications	0.5 hr
Sum	37.5 hr

Win Your Heart: a toy for children behavior analysis. This application aims to enable non-obtrusive monitoring of children's behaviour and their mental status by identifying their interactions with a pressure sensitive stuffed toy.

Pressure Password: motionless unlocking. Pattern or numerical locks are fairly common on mobile devices and keyless entry systems. However, pattern-based locks have been shown to be highly insecure as intruders can observe movements and easily crack the pattern [134]. Numerical entry systems such as in ATM machines have been shown vulnerable to thermal cameras [135]. The developers explored a motion-less password system based on the born shape of palm and length of fingers, and the combination of multiple fingers at different intensities of pressure. All the combinations look the same, making the password hard to copy by observing.

Smart Pillow: sleep position detection. A pillow covered with pressure sensing matrix can help monitor the sleep posture and enhance sleep quality. Five postures are defined, including 4 sleep positions (*supine position*, *prone position*, *lying on the left side*, *lying on the right side*) and 1 kneeing posture.

This pilot developer user study shows that making a well-documented framework can unleash the potential of the TPM sensing technology in pervasive and wearable sensing. This would also be a continuous future evaluation with the more matured, open-sourced framework from the work of this dissertation.

Chapter 5 Wearable Subspace Empirical Studies

Wearable technology has seen a successful transition from academic research to consumer electronics and is still continuously growing, incorporating new technological innovations. For example, smartwatches has a global market of 10 million USD as of 2017, and is projected to have a growth of 23% in the following five years. [136] They offer convenient information access, activity tracking, and computing capabilities.

In general, there are two categories of wearable computing devices: solid and soft. In solid wearable devices, all the functional components are rigid, such as smartwatches or smart glasses. Their entire assembly can have moving or soft parts such as watch bands or hinges, but those parts do not serve any purpose in the computing system of the wearable devices. Soft wearable devices have flexible essential and functional components, such as optical fiber, rubber or textile sensors, etc. With the development of flexible displays and flexible electronics, the future of wearable technologies seems to move towards the soft side of the spectrum.

In wearable sensing, TPM can be used as touch sensitive buttons as control inputs to build smart textile human-computer interface systems. TPM can also be used for wearable activity recognition. One general approach is to relate the activities with the external force applied onto certain surfaces of garments, such as an in-sole that measures the pressure distribution from the foot, or a soccer shoe that detects ball kicks from the impact on the upper surface. Another approach is to relate the activities with the motion of the garments' surface, which is the *self* propagation in the *Propagation* design space dimension. Essentially, as people move their bodies, they can feel the clothing they are wearing as a force. As a result of Newton's third law, the clothing also receives the counter force, which can be measured by the TPM sensors. When applied mechanically tightly coupled with the skin, such approaches can also be categorized as planar pressure mechanomyography.

With fabrics such as the CARBOTEX and SimpleSkin produced by Sefar, TPM has several traits that can is very suitable for integrating into soft wearable devices. The fabrics are comfortable to wear, and can be tailored to fit irregular surfaces, together with wireless electronic hardware versions M-1 and M-2, wearable garments with pressure sensors can be worn and tested in real-life sports activities without obstructing the users' freedom of motion.

In the wearable design space, as explained above, the *Propagation* dimension includes both *direct* and *self* methods. Users normally are aware of the smart garments, thus the *Intention* is either *intended* or *unintended*. Unintended systems are mostly aimed at unobtrusive activity tracking, while intended systems are typically control interfaces. Smart garments are normally *partial* in their *Coverage*, with a common problem of sensor shifting. In this regard, matrix sensors such as TPM can handle slight position changes as long as the active locations stay in the sensors' region of interest. Wearable garments are apparently *mobile* in the *Mobility* dimension, which may limit the size of coverage. Normal wearable activity recognition requires *medium* sampling rate, while as shown in later in the soccer shoe study, some sport activities require *high speed* in the *Sampling* dimension.

This chapter then discusses 2 empirical studies on using TPM in the wearable space for sports activity recognition.

- Textile Pressure Mechanomyography (MMG): this study demonstrates that TPM sensors can measure muscle activities by the pressure between the garment and the body (mechanomyography). A tight-fitting leg band is placed on the thigh to recognize various leg exercises. The approach is also demonstrated on other parts of the body and compared with off-the-shelf electromyography (EMG) devices.
- Soccer Shoe: in this study, TPM sensors are placed on top of a soccer shoe to capture the moments of the ball impact. This study showcases what only a wireless sensor system with high-speed sampling rates can achieve - not only detecting the ball kick events, but also analyzing the force pattern during the kick which is less than 20ms.

5.1 Textile Pressure Mechanomyography (MMG)

This section discusses a wearable motion monitoring solution with textile pressure mapping sensors. [9] The core concept is that motion and activities from the users will cause planar pressure variations of worn textiles garments. The sensing fabric can be supported with elastic bands made out of a combination of rubber and textile, which is common in sports accessories as shown in Fig. 60. The fabric can be fit into normal clothing without the support of elastic bands. Slim-fitting or tight-fitting clothing can be used in a similar way as with elastic bands to monitor muscle activities. Regular-fitting or loose-fitting clothing can be used to detect contact-triggered activities such as posture monitoring from the back, or touch input for human-computer interaction as in Fig. 61.

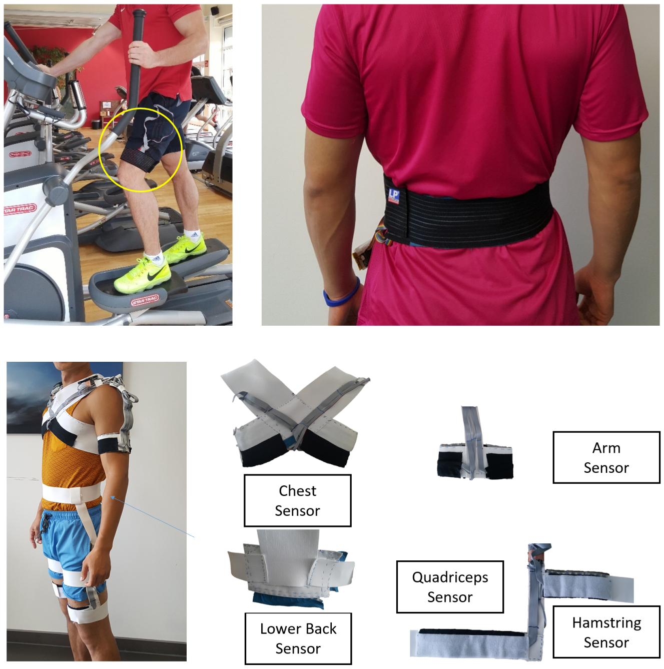


Fig. 60. TPM sensors can be integrated into stretch bands that targets different body part movements.

5.1.1 Abstract

In the recent years, fitness and sport have become an important topic in wearable and pervasive computing, as many consumer products include hardware with IMU-based fitness tracking, heart-rate reading, and also software kits to enable people to relate their activity and vital data with their activity level and training progresses. A vast number of consumers use running trackers with pedometers (IMU) and GPS-location tracking. Some body-builder also use smart-phone based training log to keep track of their training. However, most of the current data are from motion sensors and user input, and the information that can be interpolated from them are basically repetitions of a specific motion.

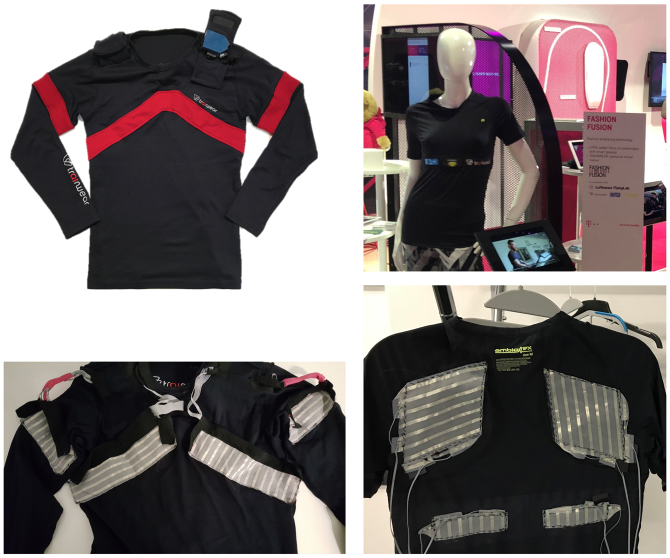


Fig. 61. TPM sensors can also be buried under a slim fit clothing without stretch band support for body motion monitoring.

Yet staying in the motion and repetition does not cover the entirety of training goals. The purpose of modern fitness and sport can be (1)well-being and health, (2)weight loss, (3) sports performance, (4)body form building. And the roots are primarily aimed at balancing calorie intake and output, improving cardiac performance and muscle hypertrophy. Sport science researchers have shown that the same motion of moving the weight can result in different muscle stimulation in terms of sections and intensities, depending on various factors such as weight and pose. [137], [138], [139], [140], [141], [142]

Therefore, the focus of exercise tracking should see a shifting from the motions alone to the muscles. Muscle activity study is also an important topic in life and sport science. Understanding how muscles move during various activities and exercises can help us compare the effect of different exercises, choose more efficient and secure methods for exercising the muscles and inducing muscle hypertrophy, which asserts its role in fields such as bodybuilding, professional sport, injury rehabilitation, elderly caring, etc.

Today muscle monitoring is largely restricted to constrained lab experiments. Long-term monitoring under real-life conditions is difficult due to the lack of unobtrusive mobile systems. Most existing approaches such as electromyography (EMG), magnetic resonance imaging (MRI), ultrasound imaging (USI), etc require either bulky hardware or specialized attachment of electrodes to the skin, making widespread deployment unfeasible. While more straightforward approaches such as force sensitive resistors attached to elastic bands [143] or inertial measurement units that measure skin surface motion have been studied, they provide only a very limited amount of information about the muscle.

[9] This section is based on the publications [61] and [56]. As the leading author of the publications, I have only included my own work; unless other authors' contribution is an integral part of explaining my work, in which case it will be explicitly stated. Since the published work has gone through extensive wording, some sentences and passages have been quoted verbatim from the said sources.

5.1.2 Hardware

An evaluation experiment is set to determine that various activities can be distinguished from the same sensor at one particular body part. The experiment is set in a leg-band configuration to attach the sensors on the surface of the tight. The hardware used during the experiment is F-3.1 as from Table 17 in Chapter 2. It has a 24-bit ADC scanning at $50fps$. It is connected to an Android smartphone via USB-OTG implemented with a FTDI interface chip.

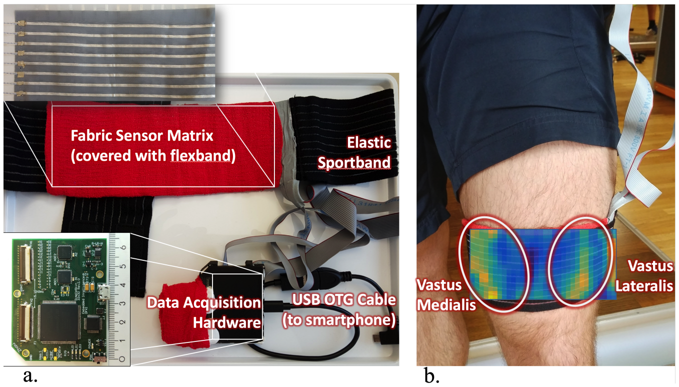


Fig. 62. Leg band study: experiment hardware set up.

An 8×16 sensor patch is stitched onto an off-the-shelf elastic sportband, which is commonly used for supporting joints and muscles. As no electrode-skin contact is required, we cover the sensors with a cohesive textile band to shield it from sweat during an intensive workout. The sensor is wrapped around the leg to cover the lower part of vastus lateralis and vastus medialis from the quadriceps muscle group as in Fig. 62(2).

After the evaluation experiment, the data is truncated from 24 bit to 12 bit, the classification results do not change noticeably. Therefore the newer hardware version M-1.1 or M-2 can also be used for wearable garments. The newer versions have lower resolution ADCs but better power efficiency and wireless transmission capability, thus the design can meet the mobility design dimension requirement. The major advantages are that those are built on low power consumption microchips instead of FPGAs, and they have wireless Bluetooth connectivity. With a $800mAh$ Li-Po battery, version M-1.1 or M-2 can operate continuously with the Bluetooth transmission for over 8 hours; while in the version F-3.1, the FPGA alone consumes $> 150mA$ depending on the implemented logic.

5.1.3 Experiment

To introduce controlled movement, and for safety concerns, exercises with free weights such as barbell squats are avoided; instead, gym machines that are designed to limit the freedom of movement are chosen and hence, in most cases, the movement of the weight is initiated by the targeted muscles. Each session includes the following procedures:

1) Warm up with the Cross Trainer's '7-minute warm-up' program (Fig. 63 A);
2) For each machine, perform 3 sets of each 12 repetitions, followed by one set, in which the participant performs until either 15 repetitions or failure. The machines include:
   a) Leg Press Machine (Fig. 63 B)
   b) Seated Leg Curl Machine (Fig. 63 C)
   c) Leg Extension Machine (Fig. 63 D)
3) Cool down with the Cross Trainer's '7-minute warm-up' program, same as step 1.

The participants are free to take pauses, drink, walk around or any normal activities inside the gym. The system will record all of those non-workout movements as the 'NULL' class. Every participant recorded four sessions, numbered $Day1 - 4$. Between every workout day, the participants take several days off to rest the leg muscles. For $Day1$ and $Day3$, the weight of each machine is constant; while for $Day2$ and $Day4$, the weight is increased after each set. The actual constant weight, starting weight and increment is participant dependent because they have different muscle strength, and their performance actually improves from $Day1$ to $Day4$. In Step 2, the order of the three machines is shuffled within the four days, so that the participants begin and end with a different machine each day to eliminate the possible bias that he/she might be already tired upon arriving at the last machine.

The exact wrapping tension and position of the sensor is not precisely defined; instead, the only standards are (1) the shapes of the muscles are visible on the app while the users stand and tighten their legs; (2) the users can fully curl their legs easily (stand and squat) with the band. The user can adjust the band if it slides during training.

5.1.4 Dataset

Overall 6 participants recorded 24 complete sessions, containing 288 sets of the three leg machines and 48 sets of the cross trainer. To ensure safety while performing those heavyweight exercises, the participants are regular members at the gym, and they are aware of how to perform all of the exercises safely. One participant is a professional gym trainer, also a university student in sport science. Three participants are university students who do regular sports or gym exercises, but are not professional athletes. Two participants are full-time researchers in the institute and do occasional sports. One of the three students is female, and the other participants are male. Their ages range: $21 - 27$, heights range: $171 - 187cm$, and weights range: $70kg - 85kg$.

The ground truth is annotated manually, detailed to the second of the start and end of each set or other movements. The class definitions are listed in Table 14. The single repetition is not annotated, because (1) counting can be easily implemented by the movement of the weights (e.g., some smart training machines are commercially available); (2) exercises with weights are slower, less consistent compared to free-hand exercises (e.g., push-ups), therefore, usually the exact separation between adjacent reps is not clear simply by looking at the person's movement. Instead, the repetition number and weight of each set are recorded in the form of a normal workout diary by the experiment supervisor.

After each set, the participant is asked to rate his/her effort of the past set from difficulty scores of easy (under weight), normal (the person feels fair effort from the muscles

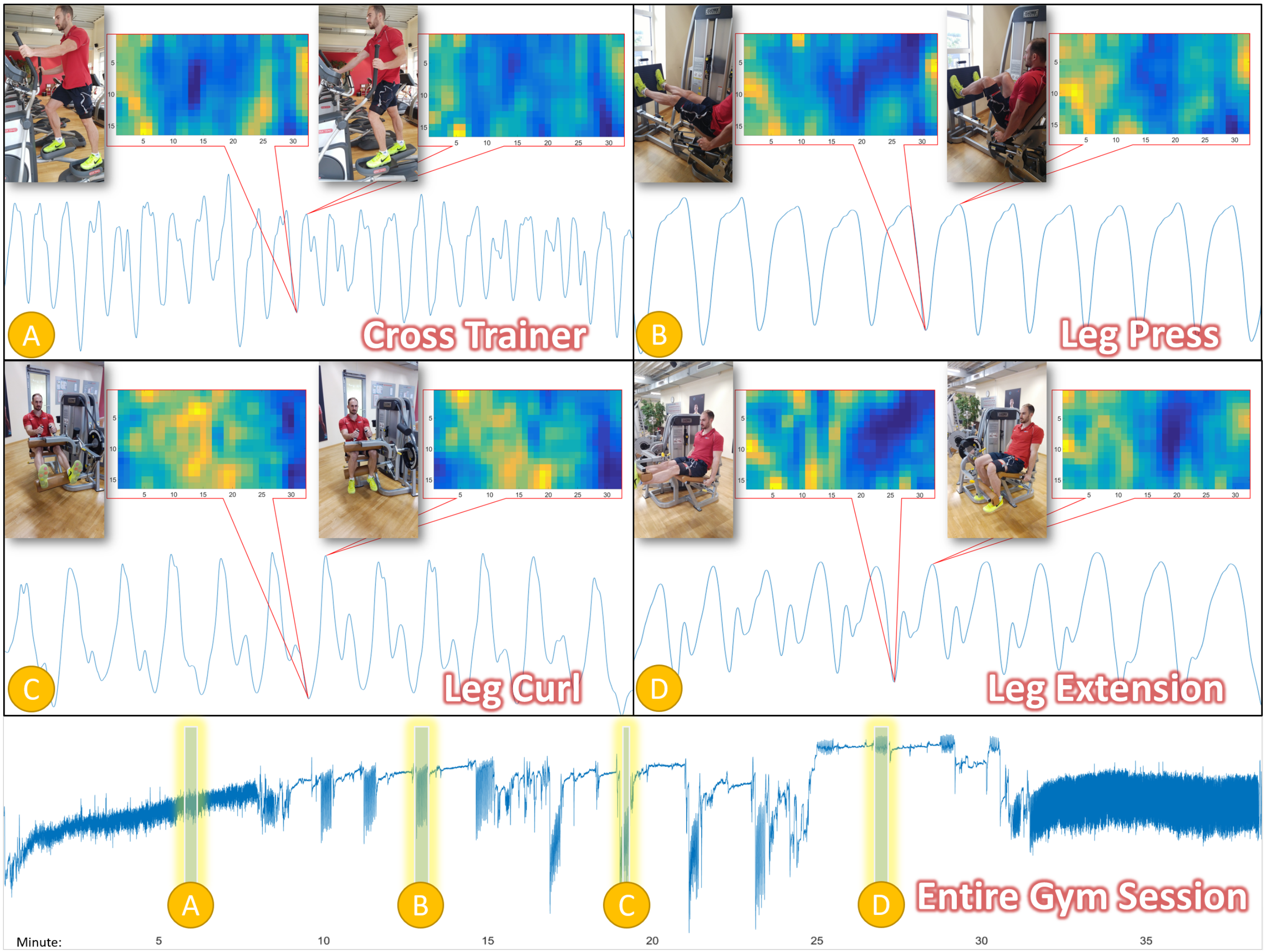


Fig. 63. Leg band study: signal example of different activities

moving the weight), hard (the person needs to continue focus and self-motivate moving the weight, movements may be inconsistent), limit (the person's movements are close to failure in moving the weight). Then the experiment supervisor, with a third person who looks at the recorded video, gives a rating of the effort based on the participant's rating, movement consistency, and facial expressions. It is worth mentioning that, the effort is a subjective score; and the effort does not necessarily increase as the set proceeds or weight increases. During the experiment, sometimes if the participant takes a long pause, he/she might regain more strength, and the next set can be easier than the previous one.

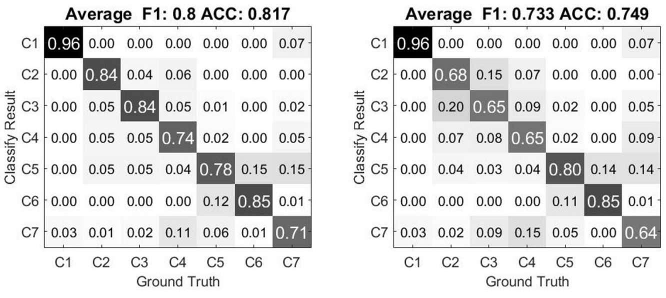
5.1.5 Signal Processing and Feature Extraction

The signal from the sensor is essentially a time sequence (stream) of 2D pressure distributions (frames). The 2D frame is first spatially up-sampled from 8×16 to 16×32. For the training and initial recognition, the data is pre-segmented according to the annotation. Then a sliding window (window size $8s$, step size $2s$) approach is used to step through the stream of data; each window is denoted as i. Features are extracted from two aspects:

TABLE 14
Leg Band Study: Class Definition

Class	Definition
Workout Activities	
Class 1	Cross Trainer
Class 2	Leg Press
Class 3	Seated Leg Curl
Class 4	Leg Extension
Non-workout Activities	
Class 5	Mounting/Dismounting/Adjusting machines
Class 6	Pause (on or off machines)
Class 7	walking

- *Temporal:* For each frame, the system computes three central moments (sum w_i and centroid x_i, y_i) and the pixel value of the pixel (σ_i) which has the maximum standard deviation during the current window.
- *Spatial:* the inter-frames are calculated, which are the difference per pixel between every pair of adjacent frames. They represent the change of the pressure mapping. We then sum those inter frames within the window, resulting in a single key frame AD_i,



(1) Person dependent case (2) Person independent case

Fig. 64. Leg band study: Leave-1-day-out average result

which represents the overall 2D pressure change in the window.

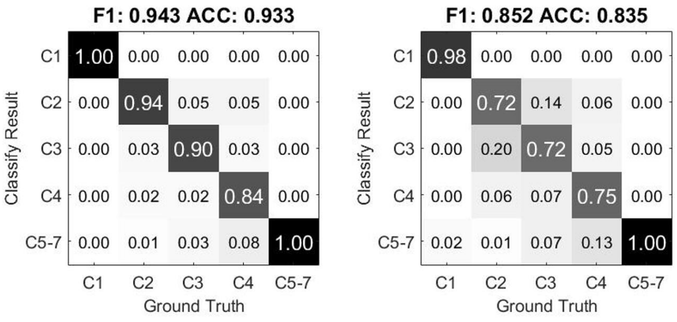
Several practical reasons could cause the sensor to have different offset values, such as the difference in wrapping tensions and positions, sliding and adjusting of the sensor band. To make the system more robust against such variations, the algorithm avoids involving absolute values in extracting features. For each temporal sequence $s \in \{w_i, x_i, y_i, \sigma_i\}$, s is normalized so the average value is 0, and the standard deviation(SD_s) is 1. The signal is filtered to remove DC and high frequency noise with zero-phase digital filtering. Then the following features are computed, denoted as $F_t(s)$:

1) magnitude range ($max(s) - min(s)$);
2) average absolute of the 1st-order derivatives ($avg(|s'|)$);
3) standard deviation of the 1st-order derivatives ($SD(|s'|)$);
4) range of the 1st-order derivatives ($max(s') - min(s')$);
5) central frequency of the FFT spectrum (without $0Hz$);
6) divide the spectrum into 5 even portions of frequency band, and then use the mean magnitude of each as a feature;
7) sort the data into a histogram of five bins, and use the count number of each bin as five features;
8) using half of SD_s as minimum peak height, find the local maxima and minima. The number count of maxima, minima, and maxima-to-minima ratio are used as three features.

Again, to make the feature extracted from the key frame AD_i translation invariant, instead of directly using AD_i, first the maximum pixel is located within a field that is 4 pixels retracted from the four borders. This pixel represents the most movement during the window. A 9×9 region is selected, centered at this pixel as the region of interest ROI_i. Thenthe first three cental moments and Hu's 7 moments [82] of ROI_i are computed, denote those 10 features as $F_s(ROI_i)$.

Overall 82 features are calculated from every window:

$$F(i) = \{F_t(w_i), F_t(x_i), F_t(y_i), F_t(\sigma_i); F_s(ROI_i)\}$$



(1) Person dependent case (2) Person independent case

Fig. 65. Leg band study: event-based Leave-1-day-out average result

5.1.6 Window Based and Event Based Classification

First, the classification results on the basis of individual windows are considered. The confidence-based AdaBoost algorithm ConfAdaBoost.M1 [144] with decision trees as the base classifier is used. The dataset is balanced by randomly picking an even number of samples from every class. Since the amount of data from every class is very unbalanced, 9 classifiers are trained with such random pick process; the final result is the majority of the 9 classifiers' outputs. 10-fold cross-validation of the complete dataset from each person is performed. This yields over 95% average accuracy; however, this is over-optimistic: the data from the same repetition set have larger similarity compared with other sets. Therefore, for everyone's 4 days, leave-1-day-out cross-validation is performed.

Fig. 64 (1) shows the average result of all participants as confusion matrix; the detailed precision, recall, $F1$ and ACC of every day from every person are listed in Fig. 66. Among the results, the gym trainer - $ID5$, has the highest scores, with minimum daily scores deviation. This is speculated to be a result of the participant doing the same machine exercise in a very consistent manner across different sessions, as his workout experience and body build allow him to control the weights with ease. We then performed leave-one-participant-out validation to examine the system's robustness against encountering unregistered users; the confusion matrix is shown in Fig. 64 (2).

Next, the classification is elevated to the event level. For every segmented set of $C1 - C4$, the majority of all windows' output is used as its final class. The remaining non-workout classes are combined into a single class $C5-7$. As shown in Fig. 65, there exists accuracy improvement in every class, and the resulting overall F1 scores increase to 0.943 and 0.852, as this process essentially smooths the window based result.

In this application, it is important to first separate the machine workout activities ($C1 - C4$) from non-workout activities ($C5 - C7$), and then further distinguish which machine the user is using. From the confusion matrix, it is obvious the most misclassifications happen within the workout classes, and non-workout classes; while very little confusion happens between the two categories. Among the four machines, the cross trainer is the most distinct from others; while it is misclassified with walking in some occasions, it is reasonable since the workout is a stepping action. In

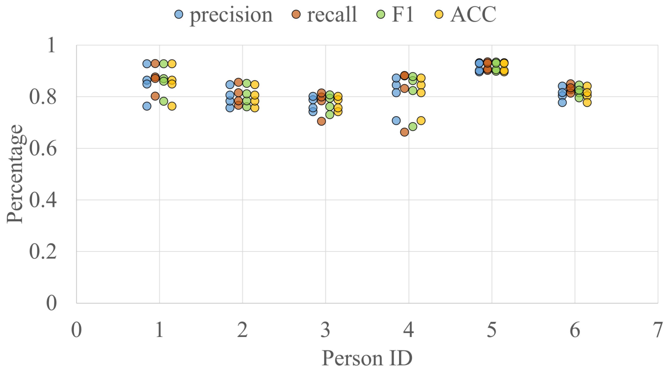


Fig. 66. Leg band study: distribution of leave-1-day-out individual day results

the person independent case, the misclassifications among $C2, C3, C4$ are increased but not significantly.

The accuracy degradation from the person dependent to the person independent case can also be a result of the participants' physiology variety in terms of muscle size, density, etc. As a baseline, it is speculated that the system can be more robust against switching users with more training data that cover more body types, and that in practical use case such a personal device can be easily trained with individual users for optimum accuracy.

5.1.7 Activity Detection (Spotting)

As a system that accompanies the user out of the lab environment, it should be able to automatically spot the user's activities in a continuous data stream without annotation. The challenge is to be able to separate the relevant activities from a $NULL$ class containing anything that the user can possibly do in between.

For activity spotting, the same sliding window i step and size are used to scan through the data without annotation, and extract the same set of features $F(i)$. The person dependent use case is assumed for optimal accuracy. For each day's data, a ConfAdaBoost.M1 classifier is trained with the annotated data from the other three days of the user (leave-1-day-out); during the training $C1-C4$, $C5-C7$ are combined into two classes $C1-4$, $C5-7$ to distinguish between workout activities and non-workout activities. The classifier is tested with $F(i)$ from every window of this day, and the output is named as the binary spotting result, in which $'1'$ indicates the workout activities. Singular events are removed, then adjacent $'1'$s are grouped into 'instances', each with the start and end points, then instances with less than 10 seconds gap are merged.

Then, two different sets of classifiers are trained: one trained with data from $\{C1, C2, C3, C4\}$, and tested on the windows marked as workout instances; another with $\{C5, C6, C7\}$ and tested on the remaining windows. The results within each workout instance are majority-voted. We compare this final spotting result with the ground truth on a 2 seconds (window step) temporal precision. The resulting F1-score is on average 0.779; however, only 31 out of 336 workout sets are not correctly spotted and classified (error $< 10\%$). The major misses are among $C5, C6, C7$, which also do not have a clear separation during the annotation.

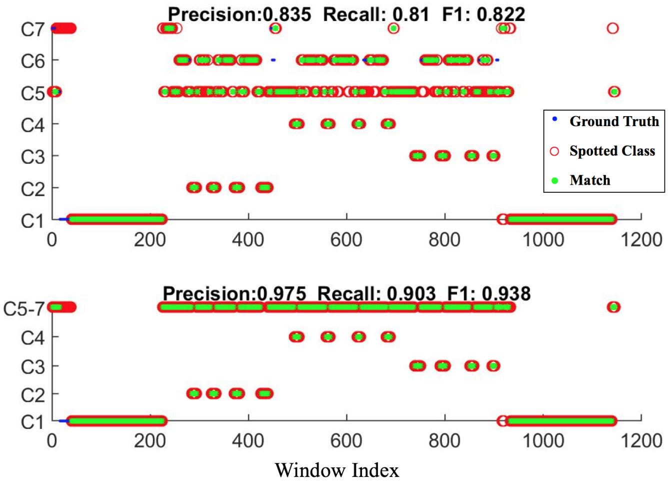


Fig. 67. Leg band study: spotting example for the session in Fig. 63, before and after combining $C5, C6, C7$

For example: when the user is approaching a machine, he/she might as well start adjusting the machine during the last few steps; when a user is relaxing, he/she might adjust the machine to release his body, or stand and slightly step around; some adjusting machine movements might last much shorter than the window size (pulling the weight pin). After all, the muscles are not actively asserting force on a workout level. Therefore, we combine $C5-C7$ in the spotting result, reducing the classes from 7 to 5, and the resulting average F1-score is 0.856.

Fig. 67 shows the spotting result of the experiment session in Fig. 63; Fig. 68 shows the distribution of the F1-scores of every session from every participants. The spotting result of each person accords with the classification result in Fig. 66.

5.1.8 Counting and Workout Quality

To automatically assess not only the type of training exercise, but also the amount is relevant. The next evaluation task is to count the number of repetitions within each spotted instance. For every instance, the sum $w(t)$ of each frame is scanned with a sliding window to locate the local maxima with the minimum peak height criteria of $median(w) + SD(w)/2$, and local minima smaller than $median(w) - SD(w)/2$ (w is the samples of the current window instead of the whole instance). Then define the smallest minima between two adjacent maxima as the start of a partition, and end of the previous partition. While this already defines a naive counting algorithm, it is sensitive to minor variations in the timing and execution of the individual repetition. Therefore, dynamic time warping (DTW) is used to inspect whether the same pattern has been repeated, or it is an overcounting of a locally abnormal peak. Because the repetitions here are not annotated, rather than choosing a standard template, the algorithm derives a template for each set, by a recursive DTW process:

1) the partitions are separated into pairs of two;
2) the time series of each pair are warped to have the same length in the time domain to have the optimum distance using dynamic programming [145];

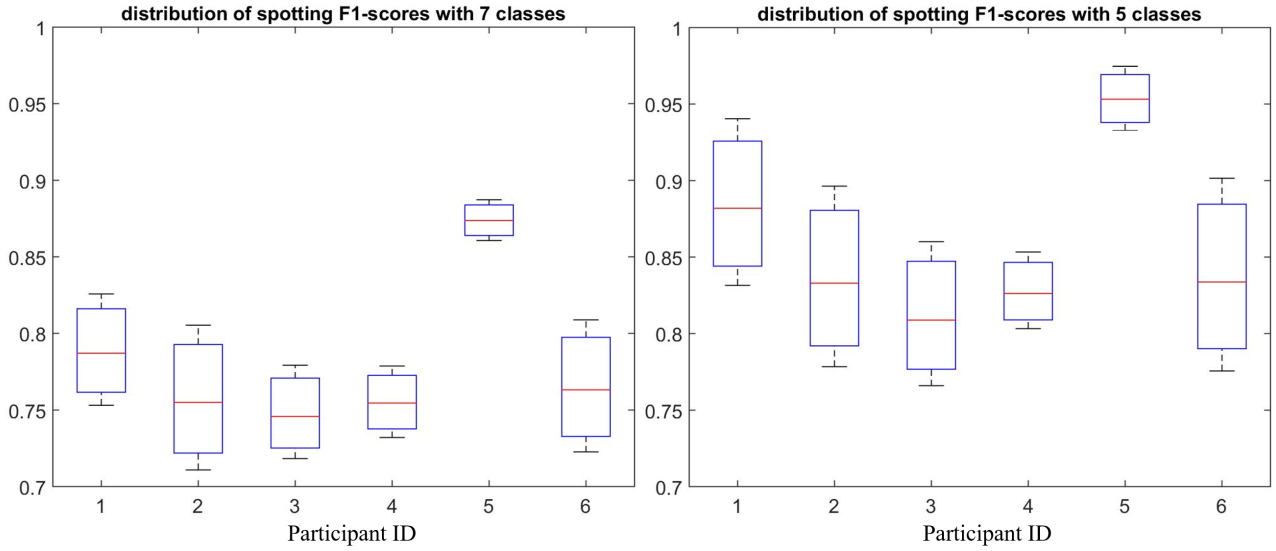


Fig. 68. Leg band study: distribution of spotting F1-scores of all participants

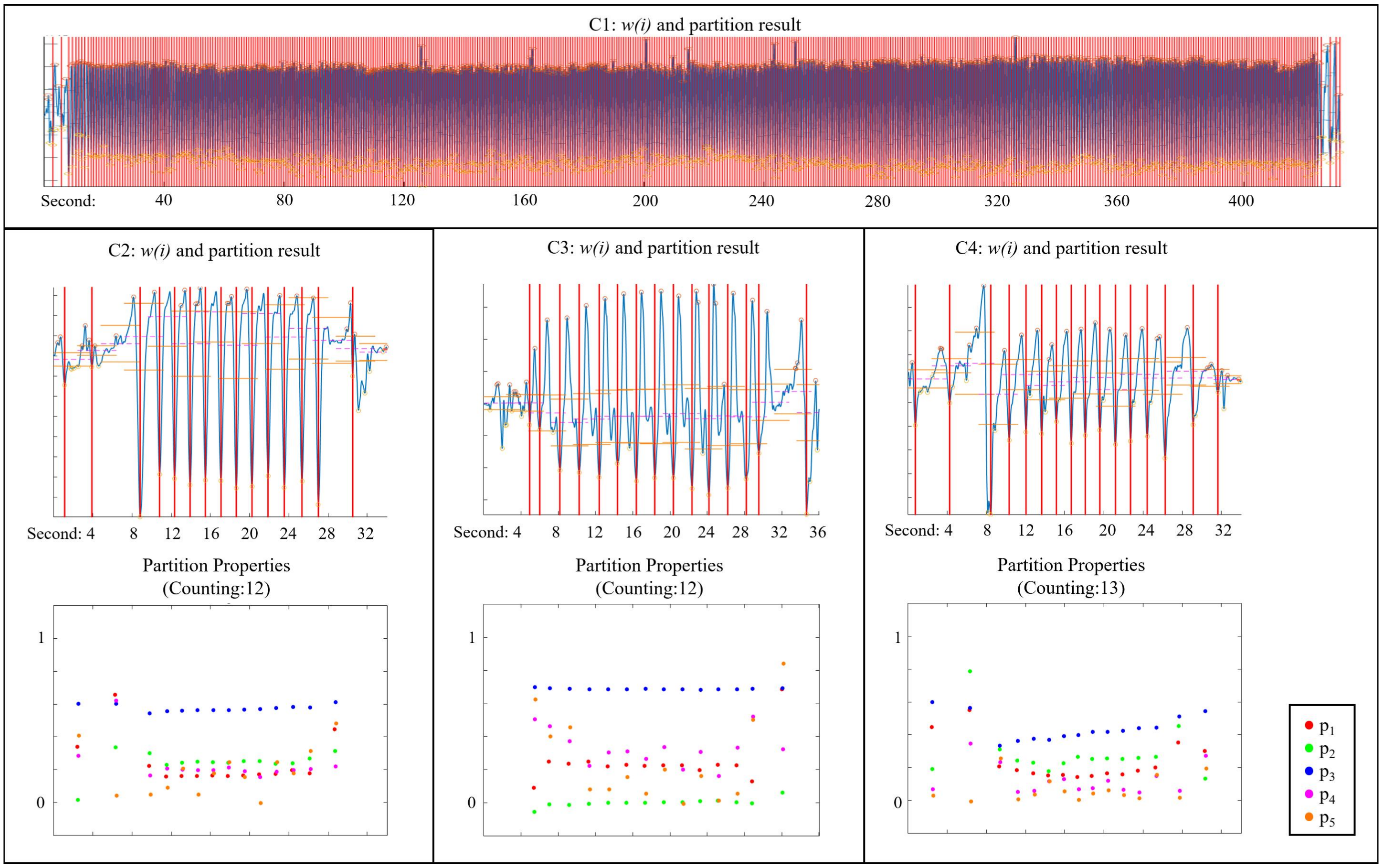


Fig. 69. Examples of partition and counting result. Red vertical lines mark the partition separations, orange and dash purple horizontal lines mark the standard deviation and median of each window.

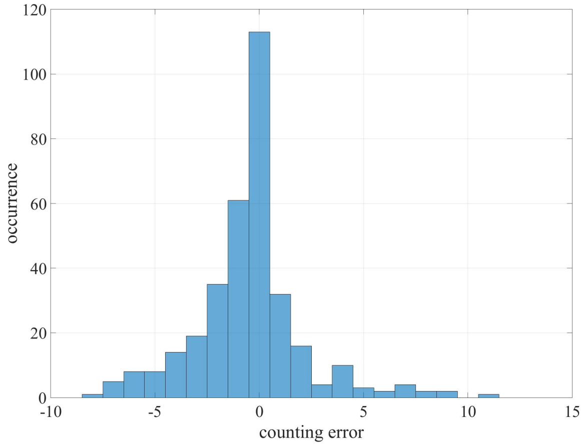


Fig. 70. Leg band study: histogram of counting errors

3) a new time series is calculated as the average of the two warped series;
4) by generating one new time series from each pair, the collection of new time series are grouped into pairs of two and go to step 2 until only one time series is left.

The remaining single time series is then taken as the template of the current set.

Then, the algorithm looks for the partitions shorter than half of the average partition length in the current set, which is an overcounting candidate. To decide if it is actually an overcounting, three time series are derived for each candidate: its own partition, the previous partition plus itself, itself plus the following partition. Then each of the three series is matched with the set's template with DTW, and calculates the mean value of the difference between the two warped series as the DTW warping error. The series with the smallest warping error is the one that best matches the template. Hence, if either the previous plus itself, or itself plus the next has the smallest error among the three, the corresponding two partitions are merged. DTW is used for correcting the counting results instead of the major counting criteria. To further save computational time, for cases that over counting candidates do not exist in a set, or for the *cross trainer* with hundreds of repetitions, the DTW correction process is not performed.

While counting of movement repetitions can be easily done through wearable IMUs or machines with position sensors on the weights/cable, the actual meaningful information is how much effort the user is asserting with the targeted muscles, and the consistency of the force patterns during the workout. For each partition's time series w, the following properties are calculated to represent the consistency variation:

1) p_1 length of the partition in seconds
2) p_2 range of absolute value $(max(w) - min(w))$
3) p_3 average of absolute value $(avg(w))$
4) p_4 DTW distance normalized by warping path
5) p_5 DTW warping error as previously explained.

The template of the DTW is calculated as the same recursive process, but after removing overcounting. Fig. 67 shows an example of the partitioning process from each machine workout class, and the according p_{1-5}. In the end, p_{1-5} outliers (within a dataset, points whose distance to the median is greater than the standard deviation) at the beginning and end of the set are removed, and the number of the remaining partitions is the final counting result. The counting results are compared with the ground truth, and shown as the histogram of errors in Fig. 70, from which, zero error takes the majority. In Fig. 67, outliers are kept to show how different the p values are for signals out of the set. p_{1-5} describe the speed, force intensity, and pattern variations, thus they can be used as a new measure for evaluating the workout consistency in future studies.

5.1.9 Observation of Warm-up Process

Muscle warm-up is an important process for improving sports performance and safety [146], [147], [148]. In this subsection, we demonstrate the capability of the system for evaluating the warm-up phase. While the skin temperature of the used muscles indicates the warm-up status; muscle warm up is, by nature, the process of increased blood flow, oxygen, metabolism, and adrenaline in the muscle, which also results in muscle volume and strength improvement. With a fixed elastic band around the muscle, this improvement causes an increment in the average pressure which can be measured by our sensor.

In Fig. 63, it is visible that the average $w(t)$ during the warm-up is increasing, while during the cool down phase, the increment is not as obvious; this difference exists in every experiment session. While the magnitude range for each step is smaller during warm up than during cool down in the particular session of Fig. 63, this does not apply to all workout sessions (in some sessions the difference is not distinguishable or even reversed). To quantify this and review all experiment sessions statistically, we fit each cross-trainer session's $x = w(t)$ with the following two functions:

$$y = k \times x + b$$

to approximate the linear increment;

$$y = m \times (1 - e^{-x/a})$$

to approximate the start to saturation trend. Fig. 71 shows the resulting k, b, a and m for every session(values are normalized for each coefficient to compare only the difference), from which, it can be concluded that during the warm-up, the average pressure increases more, with a lower starting value; and the average pressure saturates faster after the main workout routines, with a smaller start-to-saturation range.

5.1.10 Individual Set Difficulty

As each workout set has been rated an effort score, the relevance between the people's subjective perception of the exercise effort and the consistency of the force measured by the sensor is evaluated. To avoid building upon errors from counting, p_{1-5} in Section 5.1.8 is not used; instead, for every set, the algorithm uses the same peak detection scheme on the average pressure data with a sliding window, and also calculates the standard deviation of the signal within the window SD_i. Then the following pairs of features that describes the signal consistency are derived:

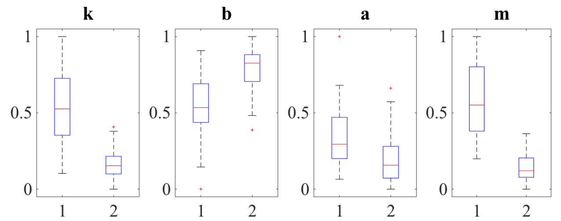


Fig. 71. Distribution of muscle warm up parameters. x-axis: 1=warm up sessions, 2=cool down sessions. Higher k means more increment; higher b means more starting tension; higher a means slower to saturation; higher m means bigger start-to-saturation range

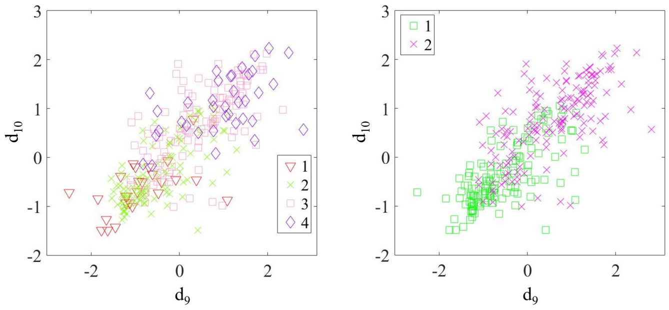


Fig. 72. Distribution of d_9 and d_{10}. (before and after grouping easy-normal and hard-limit; left: 1-easy, 2-normal, 3-hard, 4-limit; right: 1-easy & normal, 2-hard & limit)

1) avg (d_1) and SD (d_2) of local maxima heights;
2) avg (d_3) and SD (d_4) of local minima heights;
3) avg (d_5) and SD (d_6) of local maxima distances;
4) avg (d_7) and SD (d_8) of local minima distances;
5) avg (d_9) and SD (d_{10}) of SD_i;

The last pair of features are shown in Fig. 72, from which the trend of more significant deviation related to higher difficulty score is observable. To determine how well those features are separated among the four rating classes (easy: 1; normal: 2, hard: 3, limit: 4), 10-fold cross-validation with the ConfAdaBoost.M1 algorithm is performed. The resulting average accuracy is 42.50%, which is not high, yet still above random (with four balanced classes, the accuracy of random selection is 25%). *Easy* and *normal* are grouped as class 1; *hard* and *limit* as class 2, resulting in a cross-validation accuracy of 72.99%. As the ground truth is subjective, the TPM sensor offers a new aspect for evaluating the quality of the work out by measuring the consistency within each repetition set.

5.1.11 Alternative Sensor Placements

After evaluating the sensing modality through leg exercises, the hardware is further improved to enable a wireless setup between the sensor and the paired Android devices with version M-1.1 from Table 17 in Chapter 2. Therefore, the elastic band can be easily worn onto other parts of the user to explore the possibility of targeting different muscles. In Figure 73,74 and 75, the sensor is attached on the chest, back and arm.

In Figure 73, the user is performing flat bench presses with dumbbells. In Figure 74, the user is doing dead-lifts with a barbell: with a straightened back posture all the time, from an initial pose of deep squat, the user pulls up a weighted barbell from the floor till the straight standing pose, then reverses the action. In Figure 75, the user is doing cable curl-bar triceps extension: that is, while standing in front of a cable station, with the cable's terminal pulley fixed at top, he pulls down the attached bar to the lowest possible position, then releases the bar slowly to the level of his chest.

We calculate the 10 image moments (3 central moments and Hu's 7 moments) of every frame and visualize one set of repetitions in each figure. From the visualized data, the repetitions can be easily distinguished from the rest of the data by the significant magnitude, frequency, repeating pattern, etc. And in the bench press case, breathing is also visible during the relax period.

5.1.12 Comparing with EMG

From the ease of wearing, the TPM approach has a major advantage that the sensor does not require direct skin contact with the sensing elements. Yet it remains necessary to compare this approach with the existing electromyography (EMG) approaches. An easily available wearable EMG device with easy data access, the MYO armband (Thalmic Labs Inc.) is chosen. It is primarily designed for arm and hand gesture control by monitoring the muscles on the lower arm. It has 8-channel, 8-bit EMG measurement evenly distributed on the circular housing. The electrodes of MYO are solid metal pads, and are fixed on the skin by the compression of the rubber from the housing. In Figure 75, the MYO is worn on the upper arm of the same user while performing the exercise, and the data of the 8 channels is visualized together with the data (10 image moments) from our approach.

First of all, the MYO EMG signal reaches saturation (at ± 128) during most of the exercises; yet this could be a problem that the MYO is only 8-bit and is tuned to monitor less extreme muscle stimuli. Then, in the EMG data, it is easy to distinguish when the muscle is exerting force (actually pulling the weight); but it appears more data processing effort is needed to extract the actual pattern of the repetition. In comparison, the peaks and the repeating pattern is already visible from our approach simply after calculating the image moments.

5.1.13 Planar Pressure Mechanomyography (MMG)

While surface electromyography (EMG), which is measuring the electrical current changes from the body, is currently a popular sensing modality when it comes to measuring muscle movement, another method in physiological applications is surface mechanomyography (MMG), which is measuring mechanical changes introduced by muscle movements. [149] Unlike EMG, MMG is not a single method, but a branch of sensing methods in which the sensors' reading can be influenced by the attached muscle movement. The activation of small muscle motor units in combination results in macroscopic changes such as muscle thickness, stiffness and surface displacement.

In [150], Orizio, et al. demonstrated that the ripples of the surface force measured by a load cell, surface displacement measured by laser and muscle surface vibration measured by an accelerometer can be used for muscle activation measurement. A piezoelectric crystal contact sensor is used to measure the lateral oscillation on the muscle surface in [151]. Optical fibers can be wrapped around the body part

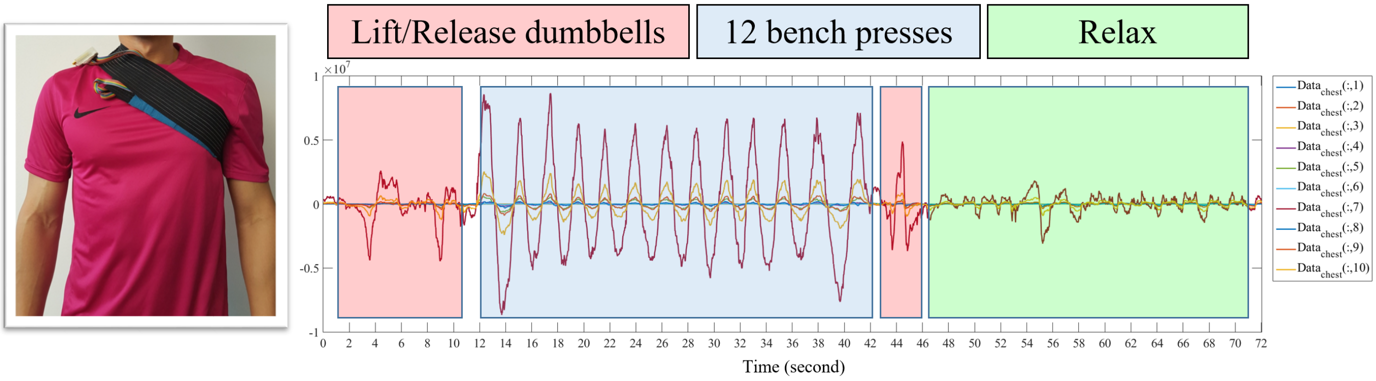


Fig. 73. Left: the pressure sensor band is attached on the chest, targeting pectoralis major muscle; right: the data (10 image moments of every frame) during a set of dumbbell bench presses, the y axis is the value of the calculated moments

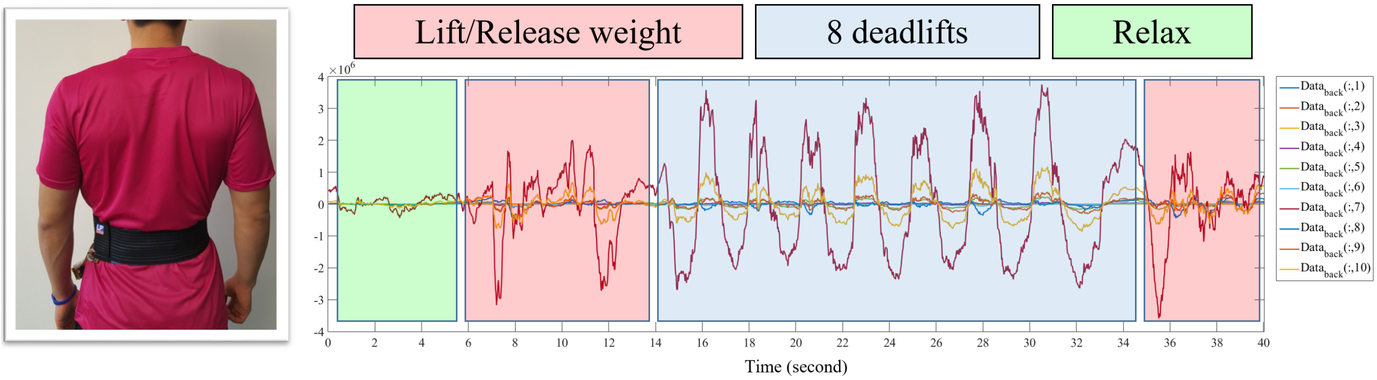
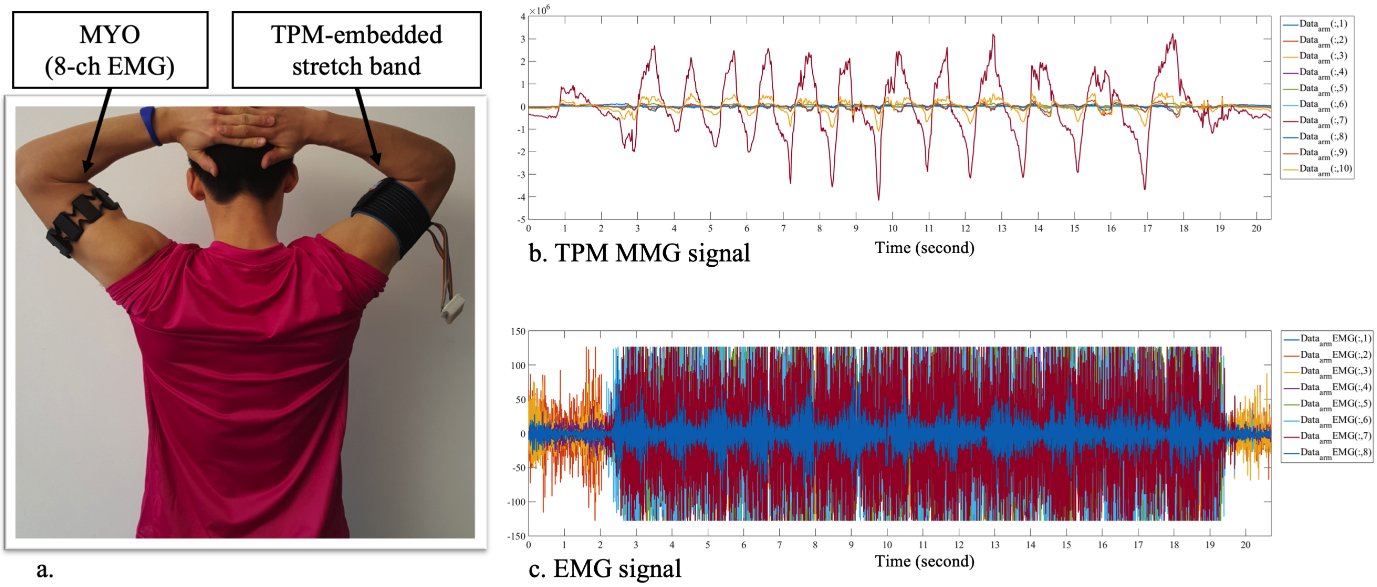


Fig. 74. Left: the pressure sensor band is attached on the lower back, targeting the lower part of spinal erectors muscles; right: the data (10 image moments of every frame) during a set of barbell dead-lifts, the y axis is the value of the calculated moments



a.

b. TPM MMG signal

c. EMG signal

Fig. 75. (a) a MYO armband is fitted around the biceps and triceps muscles and our pressure sensor covers the triceps, (b) the data from our sensor during a set of cable triceps extension, the y axis is the value of the calculated moments (c) the data from MYO during the same set of exercise, the y axis is the value of the raw EMG reading.

and bending on the micro structure results in changes of the light passed through the fiber. In [152], three optical fiber transducers are used around the arm to detect 6 distinct gestures.

In [153], a single point planar pressure MMG sensor is made from a force-sensitive resistor, with rigid enclosures on both sides to introduce a mass with inertia, and the muscle surface displacement will cause a dynamically changing force between the body surface and the mass. Even though this method is based on planar pressure sensors, the physical principle is more similar to an accelerometer.

The wearable garments in this work are essentially performing MMG based on planar surface pressure. With a stretchable garment that is closed around the muscle, the change of muscle volume during muscle motor units' movement essentially changes the tension between the body and the fabric surfaces. Overall the advantage of MMG over EMG is that there needs to be only mechanical coupling between the sensor and the body, while EMG requires stable electrical coupling, namely the conducting electrodes need to be firmly attached to the skin, in some cases liquids or gels need to be applied for better signal quality. Especially TPM sensor matrix, with its bigger area of coverage, is more robust against sensor sliding during movements. EMG is also sensitive to electro-magnetic interference (EMI), for example, the smartphone antenna can produce artefact in the EMG signal which may be mis-intepreted as physical motions [154].

5.1.14 Conclusion

In conclusion, this application has demonstrated a novel wearable system based on TPM MMG to measure the muscle movement during various sports activities. The approach is validated in a real world leg workout experiment with a first version prototype, which can be used for not only sport activity recognition, but also quality evaluation. With an improved tether-free prototype, other placements including chest, back, and arm also show similar potential from the quality of data during the relative workouts.

The major advantages of the approach are:

1) the high amount of sensing points make the device more robust against fixing variations;
2) its air permeability, flexibility and possibility to be isolated with sweat absorbing textiles make it very suitable for sports activities without introducing uncomfortableness and restriction;
3) the information can be extracted about the muscle activity goes beyond mere movement as shown by the warm-up and difficulty level analysis.

It remains interesting to investigate further leg exercises with free weights, such as squats, lunges, and the variations of pose and stance with our approach. With the wireless hardware making it possible to fit many other parts of the body without impairing the movement freedom, it is also interesting to perform detailed evaluations of other muscle workouts and, apart from the weighted resistance training, outdoor training, and other sports activities.

The approach has also led to the development of the Trainwear project in a fashion challenge hosted by Deutsche Telekom in Berlin, as well as a Percom demo [155] (best demo award), and presented in the Mobile World Congress 2018 in Barcelona.

5.2 Smart Soccer Shoe

In this section, a novel smart soccer shoe is implemented with TPM technology that measures the pressure force distribution on the non-planar shoe surface. [10] The smart shoe can be used for testing shot impact pressure and distribution which is interesting in the analysis of ball sports, especially soccer. The system is evaluated with a robotic leg in the Adidas robotic facility, with 17 different types of shots. The smart soccer shoe system endured extreme levels of impact with over $100km/hr$ ball speed. The final classification accuracy/F-score was almost 100 %. The evaluation demonstrated that such a system is capable of distinguishing various shot angles and intensities, which can be further translated into ball trajectories.

5.2.1 Study Background

As the world's most popular sport [157], much research effort has been put into the study of the science behind soccer [158]. From the classic projectile mathematics models [159], [160] to the more recent aerodynamics of the ball's flight path [161], [162], [163], [164] which concern the influence of a wider range of factors such as spin, surface roughness and the seam geometry of patches.These studies cover trajectory data measured by high-speed camera systems, both from real players and launching machines. Wearable systems to study kinematics and detect shot/pass actions based on inertial measurement units (IMU) are also being investigated [165], [166]. However, the actual physical contact which occurs between the shoe surface and the ball when kicking has not been sufficiently studied. This is mainly due to the constraints that the instrumentation should not change the physical properties of the contact surface, which is soft and irregular in shape; and that the impact time can be less than 20 milliseconds, making it difficult for tether-free implementations.

In [166], Akins has extensively validated IMU sensors and magnetic field angular rate and gravity sensors embedded into soccer shoes and shin guards. Compared to high-speed cameras and vision markers, the data showed a consistent correlation under all the testing activities. Data from IMU-instrumented soccer shoes and shin guards are also proven to be able to detect several soccer related actions in the work of [165]. In [167], in-shoe IMUs are compared with wearable electromyography on the leg muscles to evaluate kinematics and muscle activation.

Several insole foot planar pressure mapping systems can be found in [168], [169]. They are mainly used for gait analysis during walking, running or jumping; however, the placement would not give more information about shoe-ball contact, only ground-shoe contact. However, in [170], a 4-by-4 pressure mapping sensor is installed over the shoelaces to visualize the kicking force and center of pressure. This framework-focused pilot research did not include sufficient data in terms of shot variations, and furthermore, the coverage of the shoe surface is limited to only the shoelace area.

5.2.2 Hardware Requirement

Instead of introducing add-ons to the shoe, the sensing element should be integrated into the shoe surface material in an unobtrusive fashion that can be manufactured together with the shoes. In this application, an Adidas Messi 15.3 indoor soccer shoe is instrumented with TPM sensors.

The sensors are separately tailored into 3 patches. In order to form a grid of the two sets of parallel stripes, instead of 90° cross, each of them has a 40° angle. These are shown in Fig. 76, indexed as A (outside front), B (inside front) and C (inside heel). Each crossing of the metallic stripes form a sensing node, and the resolution of the nodes of A, B, C are $3-by-4$, $3-by-4$ and $3-by-3$. The front two patches are covered with very thin and soft protective textile sports tape, which does not significantly influence the sensors' readouts.

The sensors are powered by a hardware version similar to M-1.1 from Table 17 in Chapter 2. Every pixel is scanned with a 12-bit ADC at $> 550Hz$ per sensing node. The data is sent via Bluetooth to an Android device (Samsung Galaxy Tab 2 or S6 in this study). The smartphone/tablet saves the data for further processing. The sensors measure the pressure on the shoe surface cover material, which can be caused externally, by higher speed ball impacts or internally, by the lower speed movements from the wearer's foot (i.e., toes). As shown in Figure 78, every $1.8ms$ there is a data point, and for every actual ball impact, we have around ten observations.

While designing the system, we ensured that it would be feasible for implementation in real sports. With a small 800mAh Li-Po battery, the system can operate continuously for 8 hours. In this initial prototype, we have only one side of the shoe paired to one Android device. We also implemented connecting two shoes (a pair) with one Android device, which has a slight drop of scanning speed from $550Hz$ to $420Hz$ per sensing node. The dimensions of the printed circuit board and the battery are $2.6 \times 3.5cm$ and $4.4 \times 3.4cm$.

5.2.3 Experiment Design

To test the sensors' capability of measuring different football shot angles and forces, an experiment is conducted with a robotic leg that is designed specifically for simulating football kicking actions. Its mechanical details and the variation of ball flight paths are documented in [171]. The robotic leg has a controlled motor at the thigh joint, which offers a controllable leg swing speed υ. At the ankle joint, there are two adjustable hinges:

- the angle of the foot's rotation around the length axle of the leg cylinder α;
- the angle of the foot's tilt around the radius direction of the leg cylinder β.

Adidas Brazuca 2014 match balls are used in the experiment with 0.6 Bar pressure. The foot-vs-ball impact position is marked on the ball with a laser cross and is controlled to be at the vertical center of the ball from the view of the foot as in Figure 84. Different classes are defined by kicking

[10] This section is based on the publication [156]. As the leading author of the publication, I have only included my own work; unless other authors' contribution is an integral part of explaining my work, in which case it will be explicitly stated. Since the published work has gone through extensive wording, some sentences and passages have been quoted verbatim from the said source. I would like to express special thanks to Markus Wirth for making the experiment recording possible in the Adidas Herzogenaurach facility.

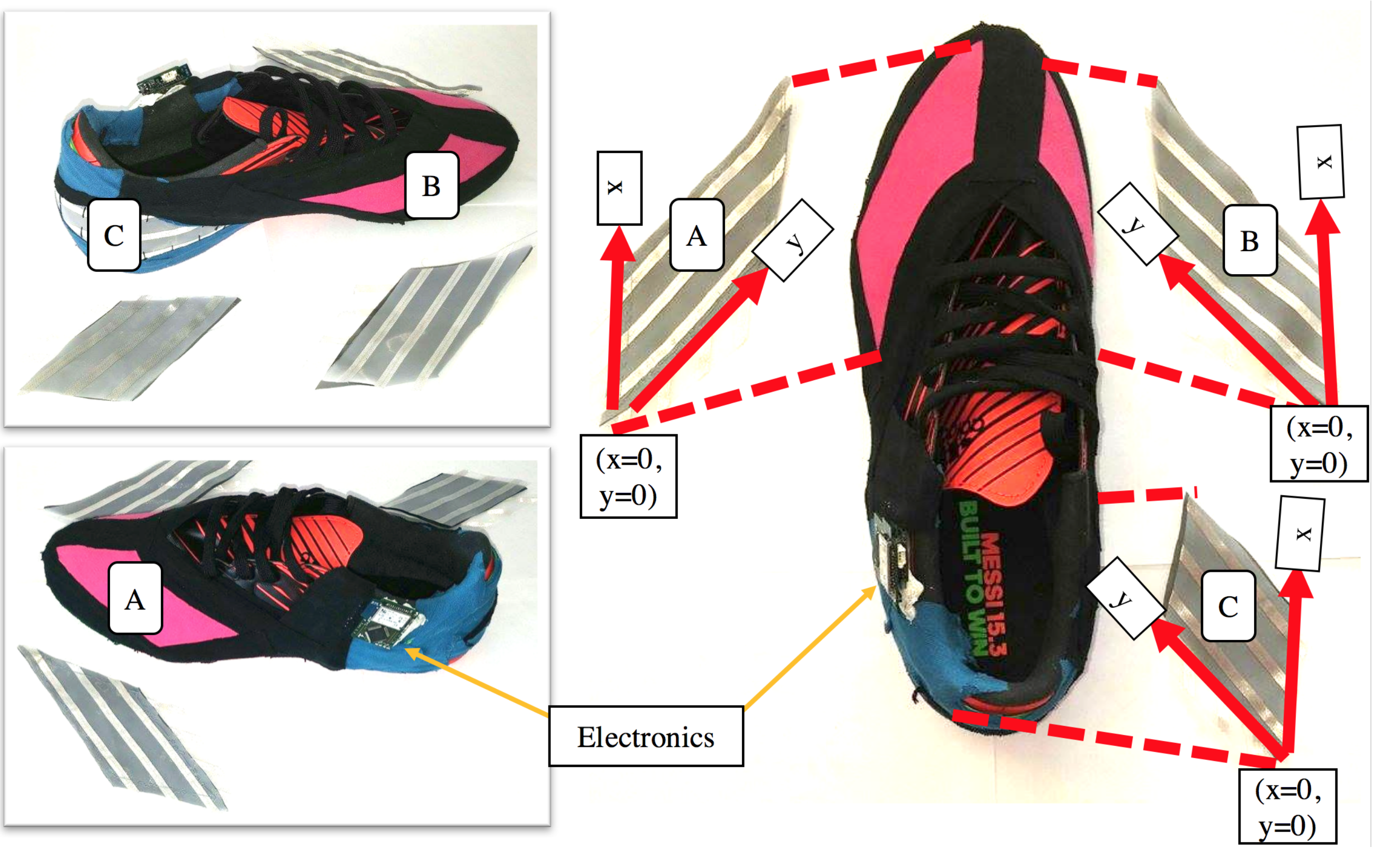


Fig. 76. Soccer shoe study: instrumented shoe with dummy sensor patches on the side

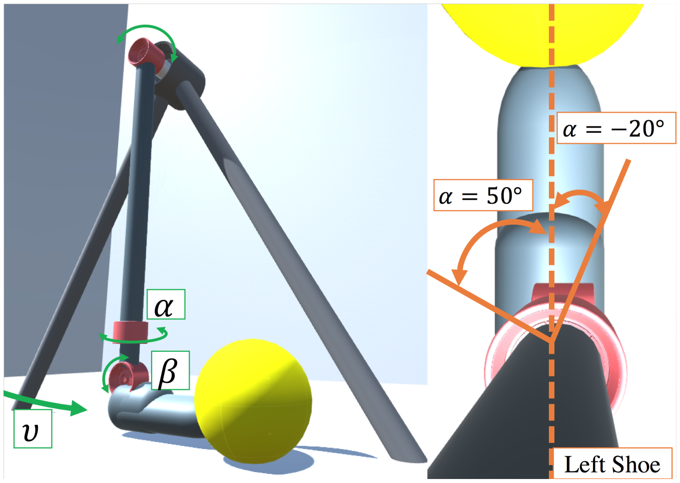


Fig. 77. Soccer shoe study: 3D illustration of the robotic leg, ankle angles α, β and leg speed $upsilon$. Free view (left) and top view (right).

types with various α and υ combinations listed in Table 15. In some the combinations, the ball would fire to the ceiling or out of the protected field at a speed of $100km/hr$, therefore the β angle, horizontal impact position of the ball, and the orientation of the robotic leg's platform are adjusted to ensure that the trajectory would point towards the goal direction like a realistic shot. With real players, the trajectory is typically controlled by the combination of multiple factors such as foot angle and body orientation of the players.

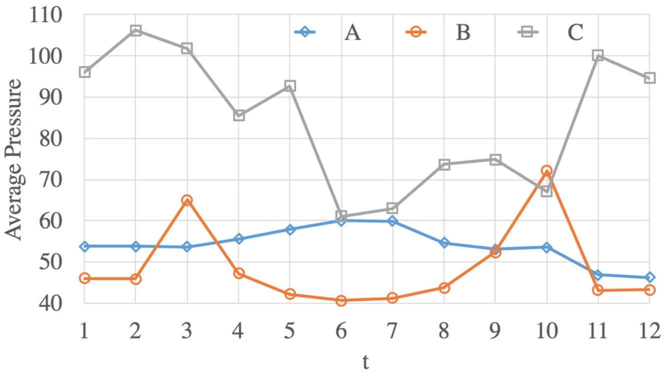


Fig. 78. Soccer shoe study: average pressure from each sensor patch during a ball impact

The speed of the ball is measured with a combination of a hawk-eye system and speed radar. The ball speeds range from around $40km/hr$ at $\upsilon = 10m/s$ to over $100km/hr$ at $\upsilon = 20m/s$, and vary with different impact insertion angle from the foot to the ball.

With multiple footballs, the ball is fired 10 times per class and overall 170 shots are recorded. Before every shot, the shoe is manually readjusted on the robot's foot to introduce some variance to the initial position. The ball speeds are shown in Figure 79; to compare the variance and offer a reference for further classification accuracy, the average $StandardDeviation/MeanValue$ ratio of ball speeds is 5.74%.

TABLE 15
Soccer Shoe Study: Class Definition

	$v = 10$	$v = 15$	$v = 20$
$\alpha = 50°$	C1	C2	C3
$\alpha = 40°$	C4	C5	C6
$\alpha = 20°$	C7	C8	C9
$\alpha = 0°$	C10	C11	C12
$\alpha = -20°$	C13	C14	C15

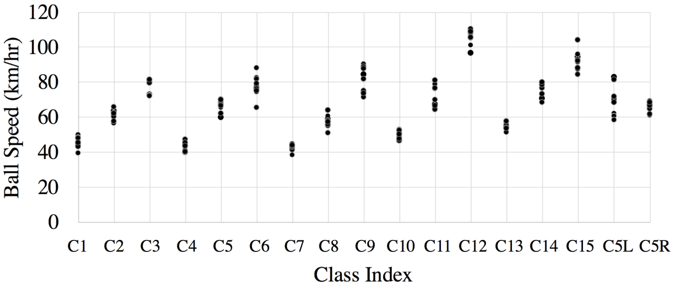


Fig. 79. Soccer shoe study: measured ball speed distribution

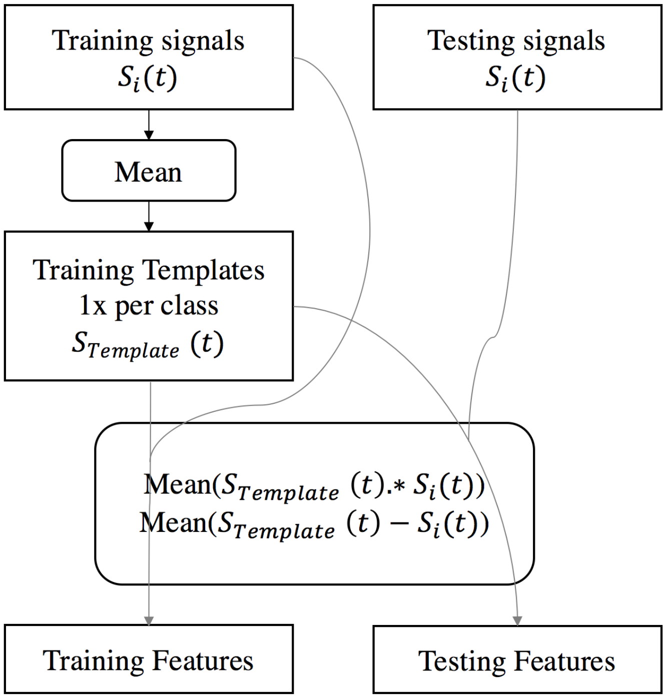

Training signals $S_i(t)$
Testing signals $S_i(t)$
Mean
Training Templates 1x per class $S_{Template}(t)$
Mean($S_{Template}(t).* S_i(t)$)
Mean($S_{Template}(t) - S_i(t)$)
Training Features
Testing Features

Fig. 80. Soccer shoe study: method 1 algorithm

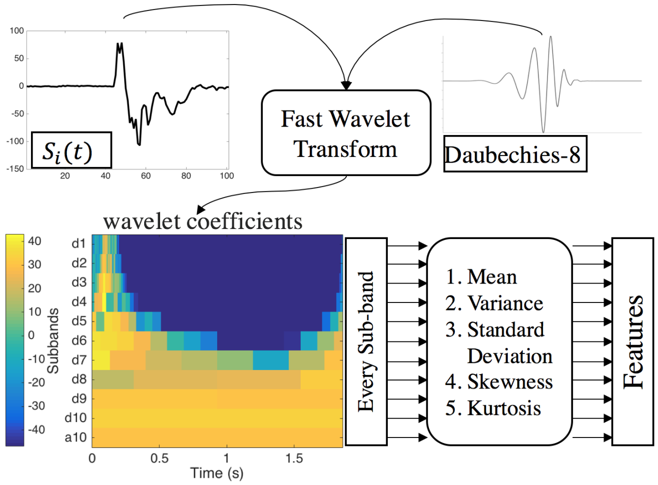


Fig. 81. Soccer shoe study: method 2 algorithm

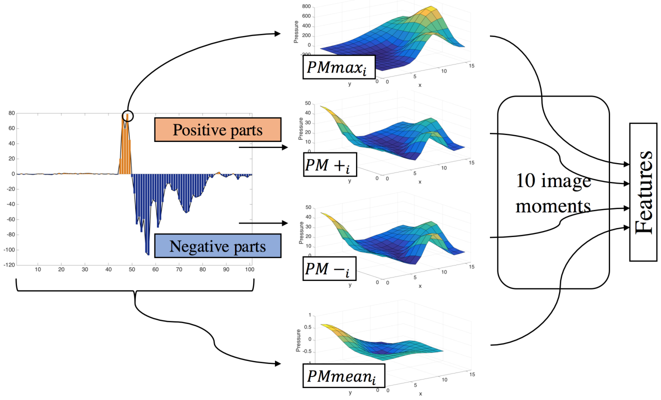


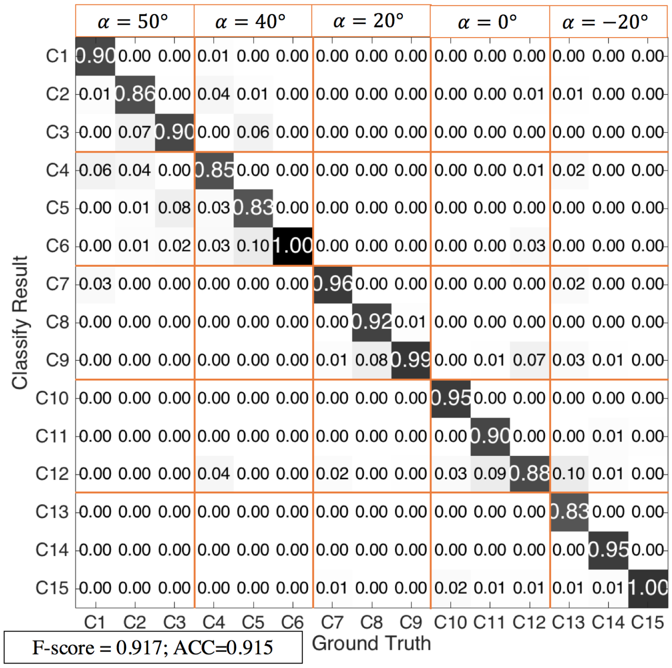
Fig. 82. Soccer shoe study: method 3 algorithm

5.2.4 Evaluation

The data retrieved from the sensors are a stream of 2-dimensional matrices, including 3 submatrices $\{A, B, C\}$ representing each patch front-outside, front-inside, and front-heel; with a time granularity of approximately $1.8ms$. As shown in Figure 87, the 3-D pressure mapping illustrates the relative pressure as well as the impact distribution and location. The fabrics are fixed onto the non-planer shoe surface with some initial pressure. Thus there will always be a minimal level of pressure present. A high-pass filter with a cut-off frequency of 2 Hz is applied to remove this offset. Then the frames are up-scaled by a factor of four on x and y dimensions. Frame descriptors - mean $w_{\{A,B,C\}}(t)$, centroid $(x, y)_{\{A,B,C\}}(t)$ and maximum $max_{\{A,B,C\}}(t)$ for each submatrix are calculated, denoting them as $S(t)$. The data recorded from the experiment are then segmented based on the peak value of the mean $w_{\{A,B\}}(t)$ during each impact. Each segment is one data sample i, corresponding to a $90ms$ window centered at the peak, the time span is named as time domain T_i. $S(t)$ of each data sample is a time-varying sequence $S_i(t), t \in T_i$.

Fig. 88 shows that for an impact of the same class, $S(t)$ have repeatable patterns. Therefore, for every $t \in T_i$ the average of $S(t)$ from all training data samples is used as a template $S_{Template}(t), t \in T_i$ for each specific class.

Combining $(x, y)_{\{A,B,C\}}(t)$, and $max_{\{A,B,C\}}(t)$, the track of the center of impact with its intensity can be visualized as in Fig. 89, where the position of circles are $(x, y)_{\{A,B,C\}}(t)$ coordinates and the radius of circles are scaled to $max_{\{A,B,C\}}(t)$. From these figures, we conclude that, as the swing speed increases, the impact intensity increases over the curve; and different α angles result in distinct tracks of impact center. These figures also illustrate that while the shoe rotates from $\alpha = 50°$ to $\alpha = -20°$ as in Figure 77, the major impact area shifts from the heel part to the front-inner, then front-outside sections of the shoe.



	$\alpha = 50°$			$\alpha = 40°$			$\alpha = 20°$			$\alpha = 0°$			$\alpha = -20°$		
	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15
C1	0.90	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
C2	0.01	0.86	0.00	0.04	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.01	0.00	0.00
C3	0.00	0.07	0.90	0.00	0.06	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
C4	0.06	0.04	0.00	0.85	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.02	0.00	0.00
C5	0.00	0.01	0.08	0.03	0.83	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
C6	0.00	0.01	0.02	0.03	0.10	1.00	0.00	0.00	0.00	0.00	0.00	0.03	0.00	0.00	0.00
C7	0.03	0.00	0.00	0.00	0.00	0.00	0.96	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00
C8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.92	0.01	0.00	0.00	0.00	0.00	0.00	0.00
C9	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.08	0.99	0.00	0.01	0.07	0.03	0.01	0.00
C10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.95	0.00	0.00	0.00	0.00	0.00
C11	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.90	0.00	0.00	0.01	0.00
C12	0.00	0.00	0.00	0.04	0.00	0.00	0.02	0.00	0.00	0.03	0.09	0.88	0.10	0.01	0.00
C13	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.83	0.00	0.00
C14	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.95	0.00
C15	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.02	0.01	0.01	0.01	0.01	1.00

Fig. 83. Soccer shoe study: example confusion matrix of $F_2(i) - NB - 2 - Fold$ with 0.915 F-score

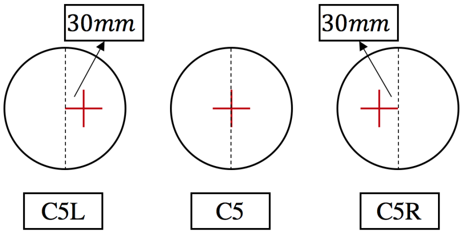


Fig. 84. Soccer shoe study: illustration of the impact center on the ball with the view from the foot, and definition of Class $\{C5L, C5, C5R\}$ ($\alpha = 40°$ and $v = 15m/s$)

5.2.5 *Data Mining Methods*

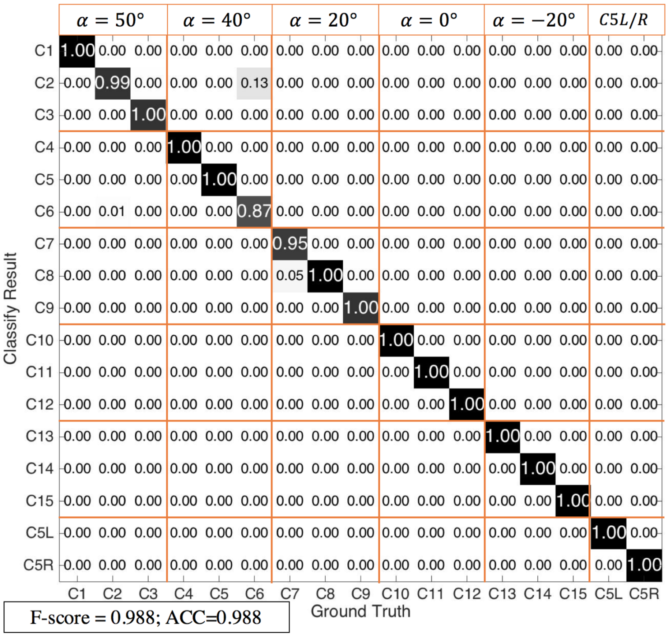
Next, to extract features that best represent the force mapping signature of different types of impact, three methods are explored.

Method 1: Template Matching

For each iteration of the cross-validation, the templates $S_{Template}(t), t \in T_i$ are calculated anew from the training samples, excluding testing samples, and each class has one set of templates. Then calculate the mean of multiplication, and the mean of subtraction between the templates and each data sample i, are calculated, and the products are used as features $F_1(i)$. Since each $S_{Template}(t), t \in T_i$ has 12 members, and overall 15 classes, $F_1(i)$ has 180 members.

Method 2: Wavelet Analysis

In Method 1, when there is a new training sample, the templates are recalculated and all the features will change accordingly. For a more generic algorithm, wavelet analysis is used with the LTFAT toolbox by [129]. The wavelet features are similar to the method introduced in Chapter 8.



	$\alpha = 50°$			$\alpha = 40°$			$\alpha = 20°$			$\alpha = 0°$			$\alpha = -20°$			$C5L/R$	
	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C5L	C5R
C1	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
C2	0.00	0.99	0.00	0.00	0.00	0.13	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
C3	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
C4	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
C5	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
C6	0.00	0.01	0.00	0.00	0.00	0.87	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
C7	0.00	0.00	0.00	0.00	0.00	0.00	0.95	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
C8	0.00	0.00	0.00	0.00	0.00	0.00	0.05	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
C9	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
C10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
C11	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00
C12	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00
C13	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00
C14	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00
C15	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00
C5L	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00
C5R	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00

Fig. 85. Soccer shoe study: confusion matrix with F_2 feature set and LDA classifier, 10-fold of 17 classes including Class $\{C5L, C5R\}$

First Mallat's fast wavelet transform (fwt) algorithm [130] is performed on every $S_i(t), t \in T_i$ with 10 iterations of the Daubechies-8 wavelet [131], producing produce a scaleogram of fwt coefficients of 11 sub-bands as shown in Fig. 81. For each sub-band, the mean of coefficients is calculated; for sub-bands $d9 - d1$, we also calculate the variance, standard deviation, skewness and kurtosis which describe the distribution of the wavelet coefficients. For each data sample, $47 \times 12 = 564$ features are used as $F_2(i)$.

2-Dimensional Analysis

While in Method 1 and 2, $(x, y)_{\{A,B,C\}}(t)$ and $max_{\{A,B,C\}}(t)$ are beneficial from the pressure mapping matrix in contrast against single-sensor FSR installments, the information from the 2-dimensional mapping is not fully used. For each data sample and the stream of 2D pressure mapping $PM_{(i,\{A,B,C\})}$, for each pixel, zero-phase high-pass filtering is applied to remove DC. Then the positive and negative values are isolated in the matrices as $PM+_{(i,\{A,B,C\})}$ and $PM-_{(i,\{A,B,C\})}$, the mean value of every pixel as $PMmean_{(i,\{A,B,C\})}$ the frame at the time when the sum of $PM_{(i,\{A,B,C\})}(t)$ is maximum within the data sample, as $PMmax_{(i,\{A,B,C\})}$. We calculate ten image moments (first three central moments and Hu's seven moments) [82], as they are image descriptors that can represent certain properties of the pixel density distribution. Overall, 120 image moments are extracted for each data sample as features, $F_3(i)$.

5.2.6 *Cross-validation*

Cross-validation is performed to evaluate if it is possible to distinguish the α angle and leg speed combinations defined during the experiment from the sensor data, with standard classifiers: kNN, Naive Bayes, linear discriminant analysis and decision tree to compare the performance of each. 10-fold, 5-fold and 2-fold cross-validation are differentiated to

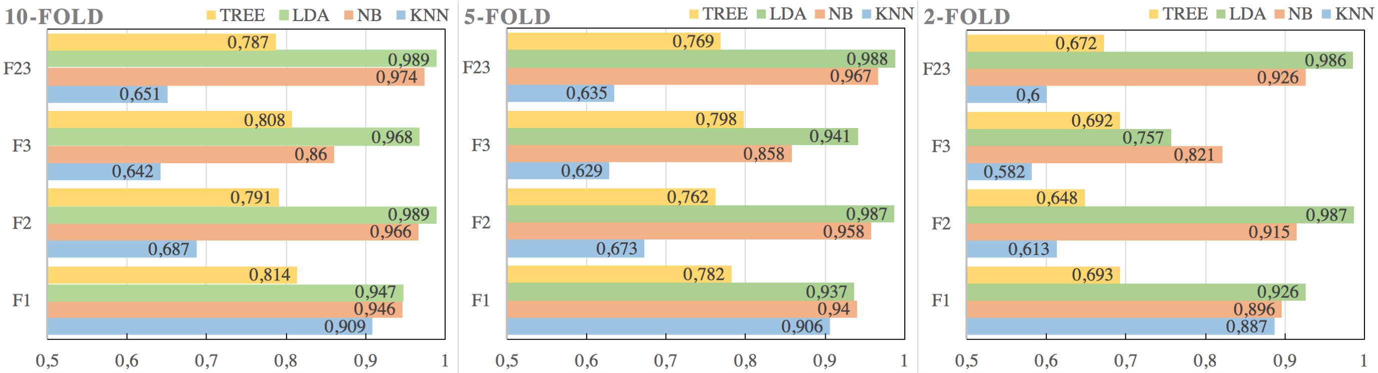


Fig. 86. Soccer shoe study: F-scores of cross-validation on different feature sets $F_1(i)$, $F_2(i)$,$F_3(i)$ and $F_{23}(i)$

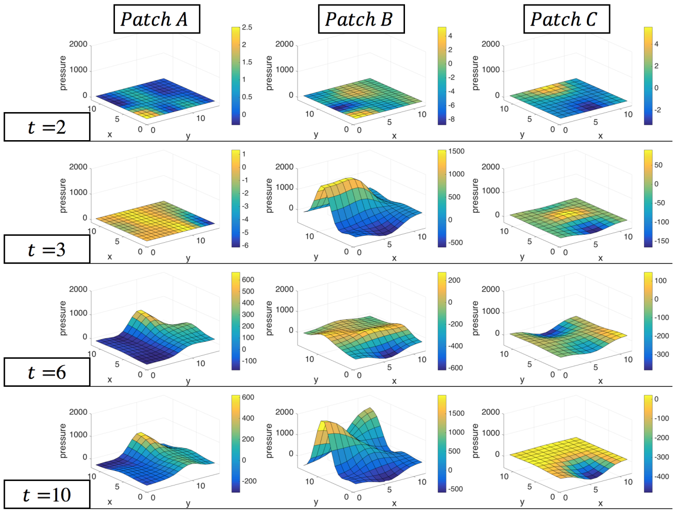


Fig. 87. Soccer Shoe Study: Example of pressure mapping readings during a ball impact, t corresponds to the data point in Figure 78; *Pressure* axis has a static range across all subplots while colormaps are adapted to individual matrix values

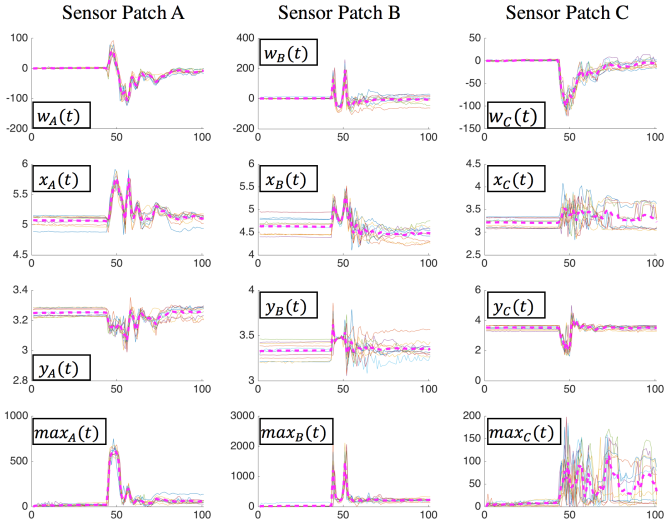


Fig. 88. Soccer shoe study: data samples of class $\alpha = 40°, v = 10$, thin solid lines are raw data $S_i(t)$, and thick dash lines are template $S_{Template}(t)$

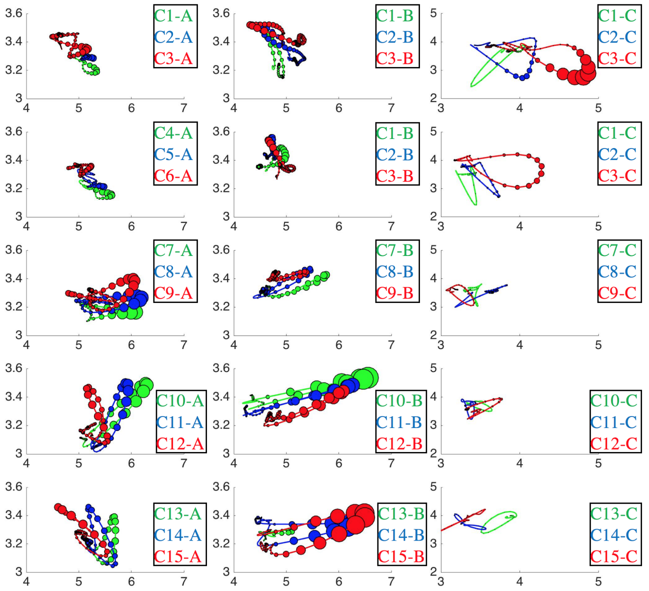


Fig. 89. Soccer shoe study: combined visualization of impact center $((x, y)_{\{A,B,C\}}(t)$ and $max_{\{A,B,C\}}(t))$ of all classes from $\{C1 - C15\}$. Bubbles' radii are scaled to the same factor.

examine the influence of the size of the training data set on the results. For each feature-classifier-fold combination the algorithm is run for 20 iterations and calculate the F-scores $(2 \times (Precision \times Recall)/(Precision + Recall))$ from the average value of the confusion matrices from the 20 iterations.

The resulting F-scores are summarized in Fig. 86. In 10-fold, for $F_1(i)$ feature set, multiple classifiers (KNN, NB, LDA) generate a similar classification result; with $F_2(i)$, $F_3(i)$, LDA has higher F-score, but the other classifiers have a poorer result; on average, combined $F_{23}(i)$ has slightly better result than individual $F_2(i)$, $F_3(i)$. As the number of folds decreases, there are less training data samples and more testing data samples, and most of the classifiers' f-scores decrease; however, with $F_2(i)$ from wavelet analysis, LDA classifier, the f-score are robust against such condition.

To show where the actual miss-classifications are located, the confusion matrix from $F_2(i) - NB - 2 - Fold$ with 0.915 F-score is taken as an example in Fig. 83. It can be seen that most miss-classifications happen between the same α angle with different leg speeds, as well as between adjacent α angles. This shows that even if the data sample is classified as a different type, the actual foot angle and swing speed lies within the adjacent possibilities.

5.2.7 Shift of Kicking Center

As explained in Section **Experiment Design**, the impact position on the ball is regulated to be at the vertical center with a laser. To evaluate whether the sensor is capable to distinguish the change of the kicking center, the ball center is moved $30mm$ away from the vertical center to both the left and right, at the combination of $\alpha = 40°$ and $v = 15m/s$, giving them the class index of $C5L$ and $C5R$ as shown in Fig. 84. This is primarily to introduce spinning variance on the ball.

With feature set $F_2(i)$ derived by wavelet analysis, and linear discriminant analysis classifier, accuracy and F-score both reach 100%. Yet it is possible that moving the center of the ball might fall into the impact condition of other α angles. Therefore $\{C5L, C5R\}$ are combined with $\{C1 - C15\}$, overall 17 classes. In a 10-fold cross-validation with $F_2(i)$ features and LDA classifier, as shown in Fig. 85, the change of the ball impact center is distinctly different from other classes by the information from our sensor.

5.2.8 Conclusion

This application has showcased using TPM sensors to cover a very irregular shaped object - the upper surface of soccer shoes. While there have been sole pressure sensors, such placement as in this application has not been observed before. It also adds to the textile integration for it uses an angled combination of the metallic stripes instead of the standard 90 degrees and utilizes the virtual surface concept to have several patches connected to a single sensing hardware module.

The data can be visualized in several ways to help understand what is happening on the shoe surface during soccer shots up to professional impact intensities. After exploring several algorithms and evaluating their performances, wavelet analysis with a linear discriminant classifier has shown near 100% accuracy with the overall 17 classes defined.

Chapter 6 Software Toolkit

The TPM framework implements a unified software toolkit which can be used across different design subspaces. The software implementation needs to consider the flexibility during algorithm exploration, as well as the real-time performance. The overall TPM software toolkit structure can be divided into:

- Tier 1 - Driver: Data receiving and logging, outputs raw data from the hardware.
- Tier 2 - Preprocessing: from raw data, performs necessary processing to parse it into the pressure mapping imagery, and improve the quality of the representation of the pressure mapping imagery, segmentation.
- Tier 3 - Machine learning: feature calculation and classification.
- Tier 4 - Output and feedback: includes data visualization (temporal waveform and spatial pressure imagery), as well as representation of the classifier results, (visual and audio) feedback based on classifier results.

The software implementation has eventually evolved into a heterogeneous implementation as in Fig. 90, that cross different platforms, and programming languages. Efforts have been taken to make sure that the software has as little dependencies as possible. They are mostly based on open-source platforms, but also licensed platforms such as Matlab. However, an implementation of the software that is completely in open-source platforms is available.

A closed homogeneous application implemented only in C++ with open source libraries is also introduced.

This chapter explains all the design concerns of the software stack, including timing and data sharing, with the goal of either a collection of heterogeneous programs or a single closed application that handles multiple tasks from the hardware level data receiving, through data processing and classification, to visualization and user feedback in real-time.

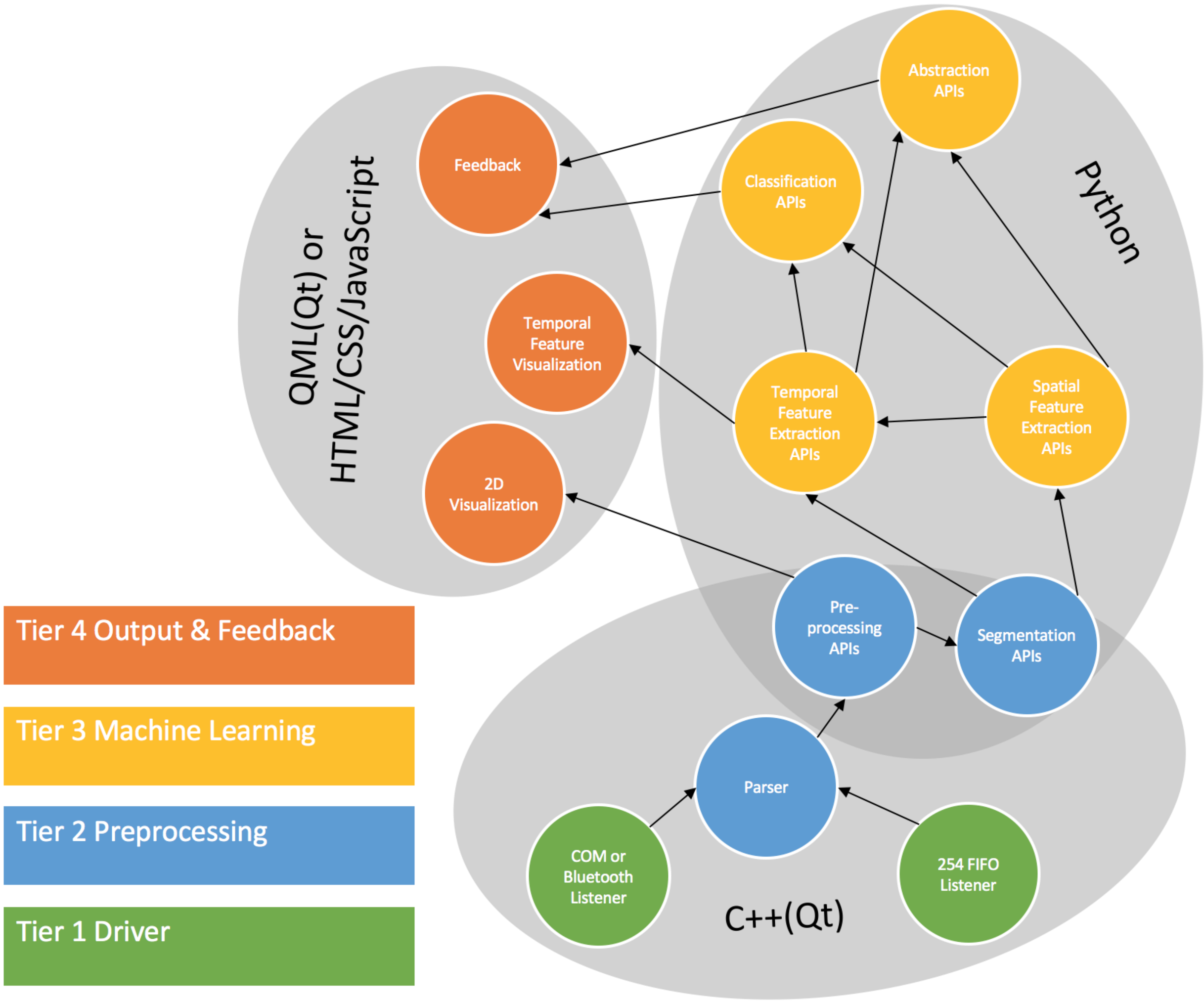


Fig. 90. Software Stack and sub-module structure

6.1 Real-time System

It is imperative to have a real-time data path to ensure the validity of the sensor data for activity recognition. The real-time goal in this part of the work is defined as:

1) Every input package (a frame) can be stored and/or processed accordingly.
2) The software should not lose new input data while processing previous packages or doing other tasks.

The hardware implementation guarantees fixed frame rate, which gives a baseline for the software framework. Yet since the software implementation is subjective to the operating system's scheduler, resources (memory size, core count, availability) and for different actual data values, the computational effort may vary, there are several key points to be taken care of during the real-time software implementation. It is possible that the computer cannot handle the amount of data, causing buffer overflow; in such occasions, the buffer should be cleared and warnings need to be registered.

Building the software has a similar challenge as the reverse of display synchronizations: the hardware is usually coded to generate pressure mapping at a fixed speed, since it is typically driven by a hardware clock and the sampling of every frame takes exactly the same clock cycles/instructions. But the software on the computer is subjective to the operating system's scheduler, the efficiency of the executing language, and sometimes the execution time of different frames may also vary based on the actual data.

Therefore, proper pipelining and multithreading are needed to guarantee the real-time performance during data collection and methodology validation in the development stage, as well as during long term evaluation and user studies in the deployment stage.

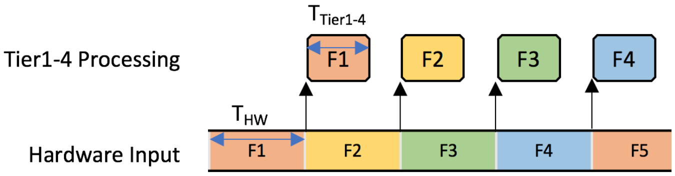


Fig. 91. Simple abstraction of Software pipeline

Pipelining is realized through multiple layers of buffers. Fig. 91 shows the naive timing scenario: assume each box is the data of a frame, different colors represent different frames, two boxes of the same color are the processing time in two software layers of the same frame. T_{HW} is the time between two frames, and $T_{Tier1-4}$ is the time needed to process a frame in Tier 1-4 of the software stack.

However, this abstraction figure is not realistic: the hardware buffer does not distinguish the frames. Instead the hardware buffer only has a linear vector of input data. Therefore Tier 1 needs to locate the frames by the frame header bytes, and this process is not synchronized with the frame sampling rate.

In Fig. 92, the software operates on a timer event with the timer interval T_{Timer1}, when a frame is located from the read buffer, the following Tier 1-4 processing will continue, otherwise if the frame is not located, the buffer is kept for the next execution of checking the read buffer. Three examples of the read buffer at $t = 3T_{Timer1}$,$t = 4T_{Timer1}$,$t = 7T_{Timer1}$ are shown at the bottom of Fig. 92. At $t = 3T_{Timer1}$ and $t = 4T_{Timer1}$, the read buffer contains an entire frame and part of the next frame, and in $t = 7T_{Timer1}$ the read buffer does not contain a full frame, therefore at that, instance the whole Tier 1-4 processing is suspended and wait for the next callback.

Basic real-time performance requires

$$T_{HW} < T_{Tier1-4}$$

Otherwise, the software cannot process and display every frame the hardware sends. In this case, the software will accumulate the frames it received in the past until the memory is filled. Yet this is the case if the operating system has suddenly less resources for the software (for example, other active applications or background tasks are using a major part of the CPU resource); or hardware is sending high data rate and the software runs on a relatively low performing machine. This can be temporally fixed by setting a fixed limit to the Read Buffer and cleanse it upon buffer overflow, so the software stack will skip frames instead of processing frames that it receives a long time ago.

A deeper look into the software stack, not all the layers need to be operated at the same frequency as the hardware input sampling rate. Typically, machine learning operates on sliding windows, of which the window step is much longer than T_{HW}. For example, in a gym exercise scenario from Chapter 5.1, to detect which exercise the users are doing, a few seconds granularity of the classification result can already be considered fine-grained. Also for the data visualization, typically 25 fps is sufficient for smooth animations in the eyes of the viewers.

Therefore, different layers of the software stack shall operate on different frequencies:

- Tier 1 Driver and Tier 2 Preprocessing should be faster than the hardware input to be able to process every frame;
- Tier 3 Machine Learning should run according to the sliding window, or the desired temporal granularity for the classification results;
- Tier 4 should follow the display output's requirement for real-time data visualization, or activity based feedback that is related to the machine learning results. (e.g., to display the classification result or confidence scores)

Pipeline separates the different layers of the software stack, so that layers can operate on different timer cycles, and also changes the real-time requirement from $T_{HW} < T_{Tier1-4}$ to

$$T_{HW} < max\{T_{Tier1}, T_{Tier2}\}$$

and

$$T_{HW} > max\{T_{Timer1}, T_{Timer2}\}$$

(assuming Tier 3 and Tier 4 do not necessarily synchronize with hardware input).

An abstracted fully pipeline flow diagram of the software stack is shown in Fig. 93. Different tiers can be executed in different threads so that different processes

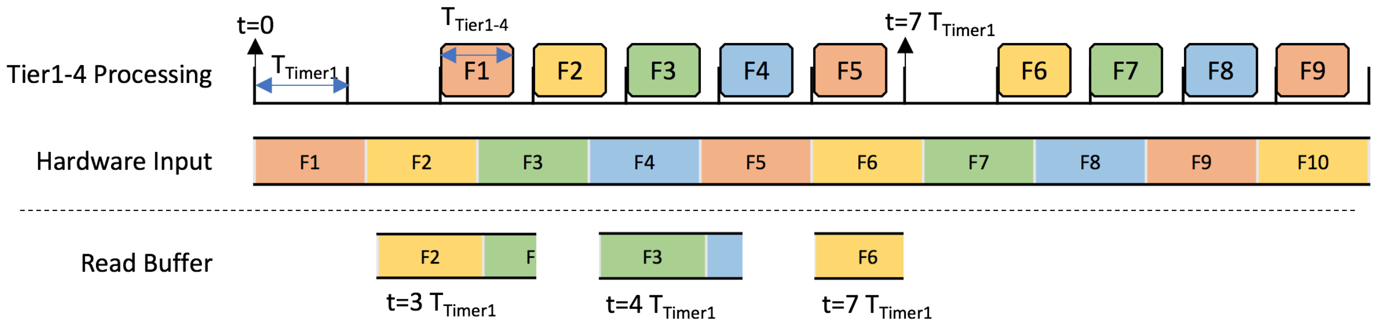


Fig. 92. Simplified practical software pipeline.

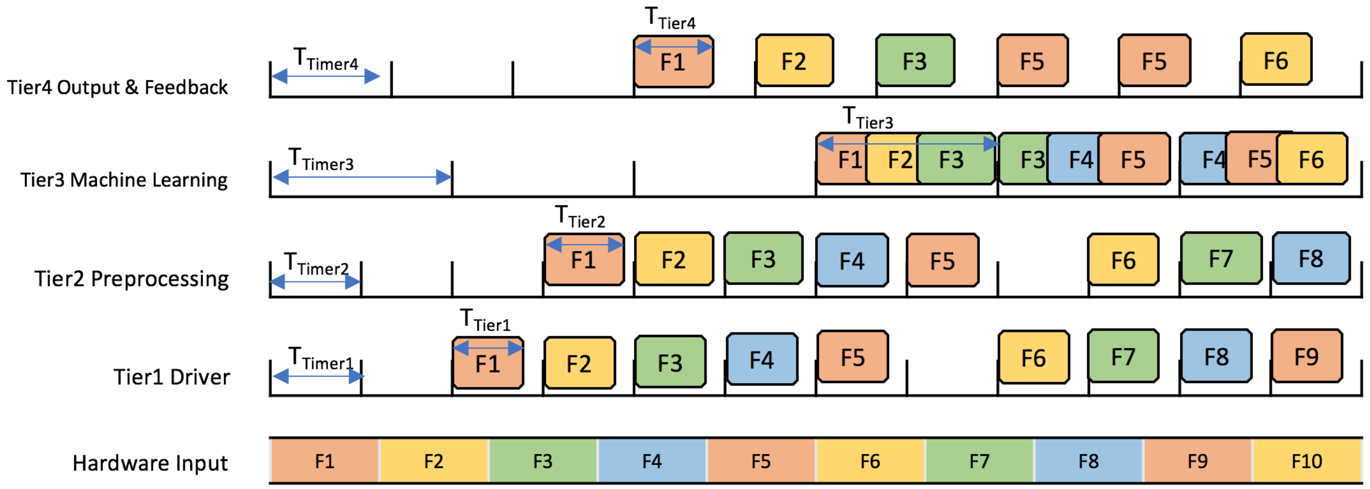


Fig. 93. Multi-threaded software pipeline. In this example assume Tier 3 needs three frames at a time.

can run concurrently to take advantage of the multicore processors in modern computers or smartphones. This way, even when it takes longer to process a frame in Tier 1 to Tier 4 together than the hardware input, but if the time needed for individual tiers is shorter than the hardware input, real-time implementation is still possible.

6.2 Layer descriptions

In order to locate and parse a frame, Tier 1 has its own read buffer, and the input buffer's data keeps append to its end at the beginning of its main execution routine. Then as soon as the read buffer is longer than a complete frame, it locates the frame headers and checks for the distance between the first and second frame headers, if it equals the present length, it is determined as a correctly received frame and moved out of the read buffer. The leftover from the read buffer is then kept for the next execution cycle. After Tier 1 locates the raw data of a frame (1D vector of the 2D frame in hex), it should save the raw hex vector into the hard-drive for further offline analysis. It also stores the frame data in a temperate variable, ready to be accessed by Tier 2.

Tier 2 transforms the 1D vector into 2D frames. Thus it requires a piece of mapping information, which can be a simple row-column combination or more specific virtual surface (Chapter 7) descriptions. Then it may perform 2D processing to the frame such as upscale, filter, crop, etc. It may also perform time-domain operations, such as temporal filtering, or removing offset from previous frames. Hence Tier 2 should have a 3-dimensional buffer, essentially a first-in-first-out time window of 2D frames.

Tier 3 calculates features from the window of processed frames. Feature calculation will be discussed in the next chapter. It also includes training classifiers and using them to predict new data. Apart from features for machine learning, there are also interpretative parameters that can be used for visualization.

Tier 4 need to visualize both the data input as videos, and the features from the time domain as waveforms. Designed visualization such as geometry changing with the interpretative parameters can also help build an interactive user experience. Classification results can be displayed, but also scripted event based on the classification results can be triggered, such as a voice reminding the user to change their posture if a 'bad' posture is detected [36].

6.3 Homogeneous All-in-One Application

C++ is relatively a low-level programming language compared with Python, Matlab, and JavaScript; however, the performance of software written in C++ is typically superior to the other languages. Also, in the Qt environment, with the help of *QThread*, multi-threading can be easily implemented to make use of multi-core CPUs and realize the multi-threading software pipeline in Fig. 93.

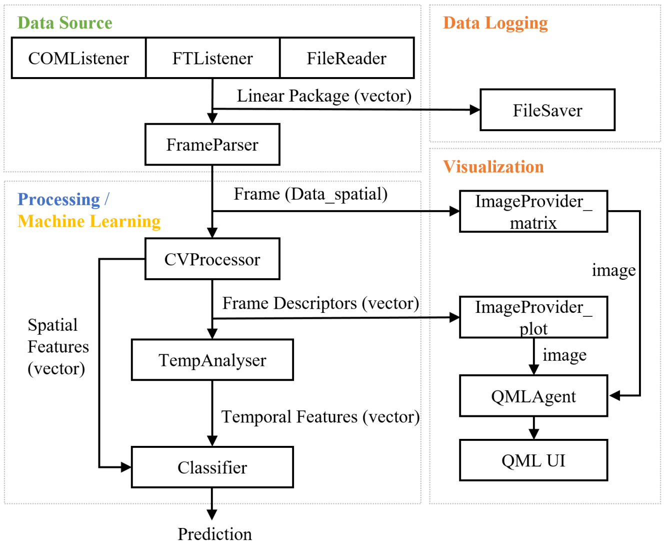


Fig. 94. The Structure of the All-in-One Qt Application.

6.3.1 Data Structures

OpenCV uses native multi-dimensional data structures as *cv::Mat*. However except for a 2-dimensional matrix, a *frame* would also include other information such as the time stamp. Temporal sequences of frame descriptors, for example, may also be accompanied with time window size, annotation, etc. Therefore three data structures are implemented as C++ objects:

- *data_spatial* is used to store and pass on *frame* data. The actual data is stored inside as a 1-D *QVector*, with the definition of the rows and columns, and methods to access single pixels by the row and column index.
- *data_temporal* is essentially a round buffer to store a single temporal sequence. The round buffer then behaves as a sliding window for continuous input data.
- *data_obj* can be used to store an event with all information, including a temporal sequence of 2-D frames and frame descriptors, and metadata such as the time window size. It is also designed to be suitable for storing a small part of the virtual surface after segmentation. Thus it includes the coordinates to locate the region of interest.

6.3.2 Process Objects

The main abstract structure of the all-in-one application's process flow is shown in Fig. 94. Every box is an object/class except for QML UI, which is a user interface designed in Qt Modeling Language (QML) for visualization. The application also has a Qt GUI (graphical user interface) as the control panel, which is also the main window thread. Settings for the TPM such as data source mode, details on the data source such as baud rate and port for the serial mode, different package descriptions such as how many bits every pixel has, dimension of the matrix, etc. can be selected in the control panel GUI. The visualization GUI and control panel GUI are separated for the concerns that:

1) the control panel can be minimized once a real-time process has started;
2) QML UI has a dedicated rendering engine and utilizes OpenGL, and thus it is more efficient with handling graphics.
3) the control panel should have very close coupling with the internal processes, which is a major difference between Qt GUI and QML GUI. The C++ code can directly access assets in the Qt GUI, and the Qt GUI can have widgets that execute call-back functions from the C++ code; while the communication between the C++ code and QML are executed with special functions.

To start the process, first, a data source needs to be chosen in the control panel. Upon start, the corresponding receiver object (*COMListener* for serial port or Bluetooth, *FTListener* for 254FIFO, *FileReader* for from a file) provides a linear package that represents a frame at a time. This package can be directly saved into a file for raw data logging. Then the *FrameParser* converts the package to a frame as a *data_spacial* object based on the information configured in the control panel.

CVProcessor contains all the processes using OpenCV. For a data analysis developer, this can be the main playground to change different algorithms without worrying much about the rest of the program. It essentially takes a frame as a *data_spacial* object and converts it to a $cv::Mat$. Inside, multiple application-specific data processing algorithms can be implemented such as filtering, segmentation, etc. The detailed algorithms are described in Chapter 8.

In Chapter 8, features for classification are categorized into *spacial features* and *temporal features*. *Spacial features* are calculated from individual *key frames*. A *key frame* is a single 2-D frame calculated from all the frames from a time window. *Temporal features* are calculated from temporal sequences of *frame descriptors* within a time window. For this all-in-one application, *Frame descriptors* can be calculated inside this *CVProcessor* class. *CVProcessor* can also store a time window of frames as either a vector of *cv::Mat* or a $data_obj$. A vector of *cv::Mat* is easier to be put into different OpenCV functions while $data_obj$ is easier to be communicated with the rest of the application, which does not have any trace of OpenCV. The decision should be made by the developer upon practical concerns. From this time window of frames, *CVProcessor* can then calculate specific *key frames*. then *spacial features* can directly be calculated from the *key frames* inside *CVProcessor*.

Thus, the outputs of *CVProcessor* can be a vector of *frame descriptors*, of a single sampling point, or a vector of *spacial features*. The *frame descriptor* vector then need to be put inside a rolling buffer as a sliding window, which is provided by $data_temporal$. And *TempAnalyser* calculates *temporal features* from the window of *frame descriptors*. Eventually, the features can be directed to a *Classifier*. The *Classifier* first needs to be able to read a training database or model from a file. And then it generates the prediction based on the input features.

During this process, the visualization is running in parallel with the data processing algorithms. The frame can be sent to an object that generates a 2-D image based on a colormap that can be changed by the developer. Then the image is sent to the QML UI by the *QMLAgent*. A

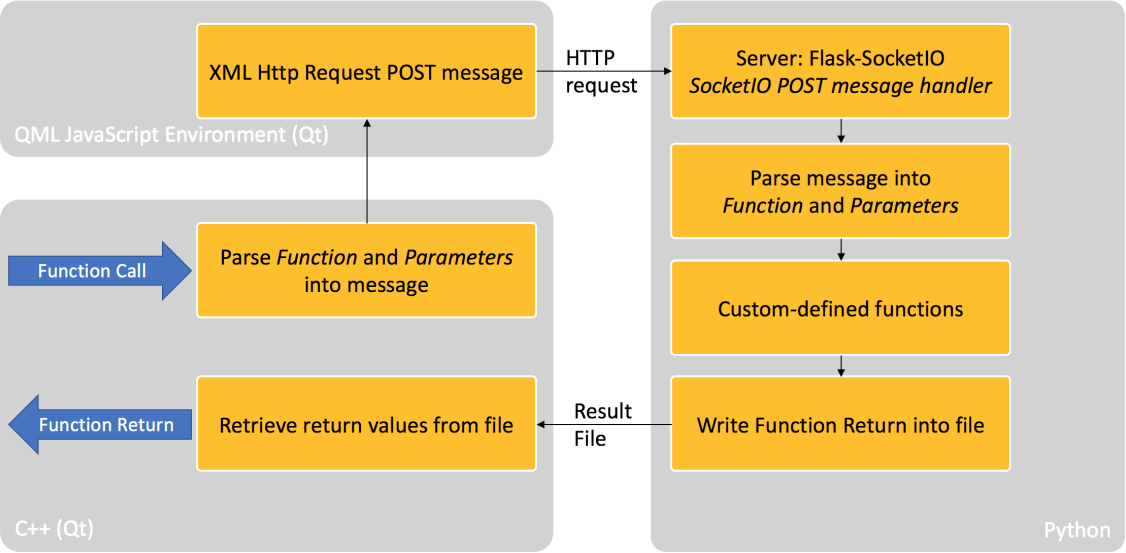

XML Http Request POST message
QML JavaScript Environment (Qt)
HTTP request
Server: Flask-SocketIO
SocketIO POST message handler
Parse message into Function and Parameters
Custom-defined functions
Write Function Return into file
Result File
Function Call
Parse Function and Parameters into message
Function Return
Retrieve return values from file
C++ (Qt)
Python

Fig. 95. The synchronous-to-asynchronous communication mechanism between C++ and Python

similar process is done to generate a time plot of *frame descriptors* such as the average value of a frame. To do this, $ImageProvider\_plot$ has a rolling buffer as a $data_temporal$ object. It uses the *QCustomPlot* library to convert the vector into a 2-axis plot that refreshes in real-time. Then this plot is sent to the QML UI by the *QMLAgent* as well. Custom functions can also be implemented to display different visualization on the QML UI in the *QMLAgent* class, such as displaying the performance specs, classifier prediction results, or changing the UI properties such as the transparency or position of a shape based on the average value of a frame.

6.3.3 Multi-Threading

Since C++ is a very efficient language, multi-threading in this all-in-one application is only carried out to separate the data source, visualization from the rest of the process. That is to say, referring to Fig. 93, Tier 1 and Tier 4 are in separated threads, and Tier 2 and Tier 3 are in one thread (as the main thread). This ensures that Tier 1 can always perform raw data logging regardless of the processing and machine learning tasks. And also makes the visualization more responsive to the sensor signal change.

6.4 Heterogeneous Software System

The entire software with 4 tiers can be implemented in C++ language in a closed executable environment as above explained, by the Qt environment, with openCV for matrix operations and machine learning, and QML for data visualization. OpenCV and C++ are very efficient and thus benefits the deployment recompile for algorithm changes. It is not convenient for exploring algorithms during the development stage. Because debugging, compiling, etc., are not as intuitive as other script-based, data analysis oriented language platforms such as Python and Matlab. Moreover, it requires specific dependencies, specific builds on different operating systems for every deployment, i.e. an OpenCV library compiled with the MinGW compiler will not work with Qt using MSVC compilers. This may also make the deployment stage more complex and raise the requirements of the machine.

Therefore, to minimize the dependencies, the example Software stack is shown as in Figure 90 as a heterogeneous system.

6.4.1 Programming Language Selection

For Tier 1, the hardware communication library is implemented in C++. While other programming languages can handle Bluetooth or serial port, the 245 parallel FIFO mentioned in Table 16 only has a library in C++, and it has the highest bandwidth compared with other methods. It is essentially the Data Source part of the all-in-one application, removing the OpenCV, processing, machine learning, and visualization. This allows the Tier 1 software to be compact and efficient.

Tier 2 and Tier 3 are mostly implemented in Python or Matlab. They are script-based, therefore it is easy to modify the programs without compiling, a wide range of powerful libraries are also available for numeric manipulation and machine learning. Tier 4 is implemented in QML if the software is a closed Qt program, or in HTML/Javascript for the heterogeneous software environment. It can run in a browser, or a *webview* in a mobile app.

Data communication between Qt and Python can be done by two methods:

- Python Server websocket callback: the python code can host a web server and handle HTTP requests.
- Qt command line: Qt can execute command line functions with parameters, and the python code can be wrapped to exchange input and output in this manner.

6.4.2 Data Sharing

Web engineering is adopted to achieve connecting different parts of the software seamlessly with minimal dependencies.

In this structure, the HTML/JavaScript UI or Python back-end can be modified without interrupting the Qt program that is handling the real-time hardware reception. This setting is based on the considerations that: in real-world development, it is more common that the UI in Tier 4 and back-end algorithms in Tier 2 and Tier 3 need frequent modification as soon as the hardware is sending stable data; restarting Tier 1 would often need reconnecting the actual hardware or selecting the port, with wireless connection it is also possible the operating system loses the device configuration after frequent reconnecting.

In the actual implementation, the C++ part operates on a synchronous mechanism: from the receiver listener which assembles data packages to the processing routines which operate on every frame and the visualization, those methods all operate on respective timers to ensure a fluent operation. While in the Python part, functions are called on demand, therefore an asynchronous option would be preferable.

6.5 Android Application

An Android application is developed for logging data, which enables a range of mobile use case studies. The application only has Tier 1, Tier 2 and Tier 4, without the machine learning part from Tier 3. It supports both Bluetooth and USB-OTG with an emulated serial port for data source, while Bluetooth is the preferable choice of connection. Tier 2 calculates some basic frame descriptors such as the average value, and visualized in Tier 4. The visualization is implemented either with the 'Androidplot' library, or with a *webview* widget displaying an HTML/JavaScript based visualization. With Androidplot, it is more efficient and stable during runtime, but *webview* can essentially use the same visualization code from one written for computer web browsers, and is more flexible in terms of changing the plotting.

In the future, it will be interesting to implement feature calculation and classification on the Android application, so that everything runs in the mobile phone applications and services. This can be possible with the recently released TensorFlow Lite library, which is a library supporting neural networks optimized for mobile applications.

Chapter 7 Hardware: Generic Architecture, Design, and Implementations

This chapter discusses an unified hardware architecture and the design concerns in specific applications. The hardware implementations are also described in details. Some implementations are used for the empirical studies in Chapter 3, Chapter 4, and Chapter 5; some implementations are further improvements based on the lessons learnt from these studies.

Previously, Chapter 2 introduced the textile sensing principles and the types of materials that can be used. An overall view of the design space was discussed regarding how they can be used to acquire information about user activities.

The electronics include all the functionality needed for implementing a TPM application except for heavy computation which is done on computers or smartphones. The goal of the electronics is to generate pressure mapping imagery at high refresh rates, which is similar to a camera. However, for decades of development, video systems benefit from a mature range of dedicated, specialized ICs (integrated circuits) and data compression/transmission/encoding protocols to accommodate the parallel computation needs; while there is none for TPM hardware.

A major part of the hardware of this work focuses on how to implement the system with off-the-shelf ICs and components [11] in a compact footprint. An architecture is published in [172] to discuss the theoretical architecture designs; however, many practical concerns are encountered during the actual implementation and development of two directions of systems: large scale and ultra-portable. The large scale direction points towards table, floor carpet, sports mat, or even room-sized applications. In this direction, power consumption is less of a concern, but parallelism and scalability are given higher priority. The portable direction aims at promoting wearable garment and movable objects. Thus power efficiency, wireless connectivity, and footprint are more important. Efforts have been taken in reaching both directions of high performance that can detect normal human motions with $> 20Hz$, nominal $40Hz$ refresh rate of the whole sensing area.

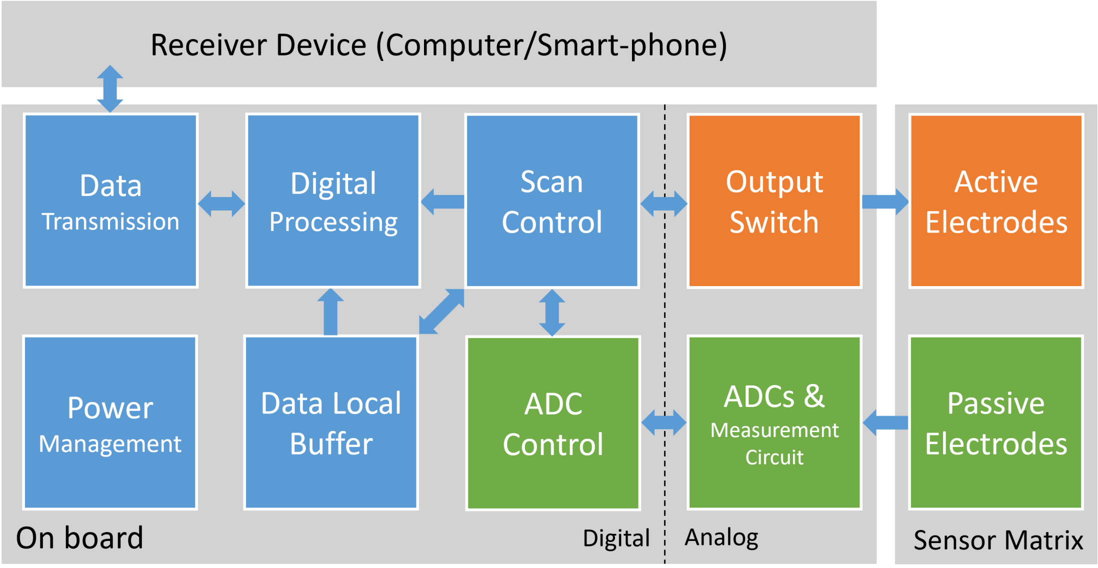


Fig. 96. General hardware architecture

Structure of this chapter:

- The unified top-down hardware architecture that supports on all the empirical studies' apparatus and other implementation from this work is described;
- Several key details of how the architecture can be versatile in various application settings are discussed;
- Actual implementations of the architecture for portable and large-scale applications are introduced with discussions on various design concerns over the time span of this work.

[11] The off-the-shelf components date from the year 2013, the start of this work.

7.1 General Architecture

A unified general hardware architecture is described in Figure 96 and in [172] to facilitate the two-stage scanning process of the sensor matrix.

During the scanning of an entire frame, the *Active Electrodes* are turned on one by one. And the voltage levels from the *Passive Electrodes* of the voltage dividers are measured at the *ADCs & Measurement Circuit*. *Active Electrodes* and *Passive Electrodes* each represent one dimension of the pressure mapping frame, and their cross-points are the corresponding pixels. The *Output Switches* drive the *Active Electrodes* and offer a fast and stable voltage level when it's turned on, and connect to ground or high impedance when it's turned off. The actual implementation may have an influence on the imagery result, as an example shown in Section 7.2.1. The number and volume resistance of sensitive points on each *Active Electrode* will influence the absolute maximum load of the *Output Switches*.

The ADC sampling rate is crucial to the final refresh rate, therefore high speed ADCs are required. In this work, one underlying principle is always a large amount of ADC channels; this can be achieved by either combining high speed analog multiplexers with a single channel high speed ADC, or ADCs with multiple input channels. Both approaches have been attempted and evaluated in the following subsections. In general, the combination of high speed analog switch and high speed ADC offers higher sampling rates than most multi-channel ADCs, at a cost of adjacent channel cross-talk. If the analog switch and ADCs are not perfectly synchronized, the voltage from the previous sample may be partially included in the current sample . This may result in a 'shadow' of the object on the imagery.

The *Scan Control* triggers turning on and off the *Output Switch* arrays, and receives measurement data from the *ADC Control*. It then streams out the data via *Data Transmission* to the *Receiver Device* if immediate data out is programmed in the firmware. Since the combination of multiple *Passive Electrodes* and *Active Electrodes* forms an entire *frame* of the pressure mapping imagery, the *Scan Control* also reconstructs the complete frames in an internal *Local Frame Buffer*. In some operation modes differentiated by the firmware, the *Data Transmission* sends entire *Local Frame Buffer* as packets instead of streaming transient ADC data.

General ADCs have a sample and hold (S/H) operation cycle, where the input signal is captured by a buffer to a capacitor, and then discharged during the analog-to-digital conversion [173]. To have a consistent measurement across all the sensitive points, the *Scan Control* needs to control that the ADC has an identical sampling time. This is first achieved by the parallel capability of the FPGA, where the scanning logic is driven by a fixed clock and interface with the rest of the logic through buffers, and additional digital logic does not influence the scanning logic. On a micro-controller with embedded ADCs, carefully designed firmware with fixed software triggers and without using interrupts can also achieve a constant sampling time; since they run on sequential instructions, the digital processing, and data transmission needs to be taken special care so that they do not introduce noticeable variance of the number of instructions between samples.

In a workflow of a novel application with TPM, typically there are two stages that can be abstracted down to the root: **development** and **deployment**. During development, the raw data is sent to the *Receiver Device* for analyzing, for offline evaluation through signal processing, segmentation, feature extraction, and cross-validation. Then a complete algorithm chain is formed from the development. Deployment is to test the system with the algorithm chain operating online with real-life usage conditions or demonstrations. The on-board *Digital Processing* in this case can take over some of the primitive stages of the algorithm chain, such as: removing pixel offset; pixel-wise oversampling-temporal average filter for the entire frame; essential feature calculations including the center, average, and area of pressure. This can help the deployment by means of:

- decreasing the workload of the *Data Transmission* by reducing the packet size (e.g. sending features instead of entire frame) or sending fewer packets (e.g. after average filtering the over-sampled frames).
- decreasing the processing effort of the *Receiver Device*. This can be especially important for mobile devices.

In some applications, all the data processing should be on board. For example, as a Bluetooth textile touchpad for controlling a PC, the operating system recognizes the system as a Human Interface Device Profile (HID) and only receives pointer movement and keystrokes; therefore, all the pressure center tracking, gesture recognition, fault rejection needs to be implemented onboard. As manufacturers continue to improve higher performance embedded processors, some of the more sophisticated workloads may be possible to be implemented on-board without impairing the timing requirements of the *Scan Control*.

While it is a straight-forward and intuitive architecture, it is essential that the architecture scales up well for larger systems, and the implementation of each submodule can also influence the performance and measurement results. At first, FPGA driven systems are developed to ensure the real-time performance. And high performance, 24-bit ADCs are used, because first, we would like to see if there is any advantage of using the highest possible analog resolution, then after trimming the resolution inside the software and no information loss is present, it would suggest that with lower resolution ADCs, it would not compromise the useful information. This helps the further development and optimization of the hardware since lower resolution ADCs are typically faster, and models with higher channel numbers are available.

This architecture is also designed to be easily up-scaled. The largest implementation has four parallel ADCs, each is paired with a 32-channel multiplexer (Passive Electrodes), the whole matrix has a maximum size of 128-by-128, with 40 fps scanning rate. In principle, FPGA based systems do not have an upper limit of the number of parallel ADCs, unless the frame is cached on the hardware before sending it to the receiver computer. Therefore the said 128-by-128 system

TABLE 16
Data Communication Methods

Method	IC	Compatible Host	Receiver	Speed
Normal UART	FT232R[1]	uC[2] or FPGA	Computer (COM Port) / Android (USB OTG)	$\leq$ 92KB/s
High baudrate UART	FT232H[3]	FPGA	Computer (COM Port) / Android (USB OTG)	$\leq$ 800KB/s
Quad UART	FT4232H[3]	FPGA	Computer (COM Port)	$\leq$ 3.2MB/s
245 parallel FIFO	FT232H, FT2232H, FT4232H[3]	FPGA	Computer (C++ driver)	$\leq$ 40MB/s
Bluetooth Classic	RN42[4]	uC[2] or FPGA	Computer / Android	$\leq$ 30KB/s

[1] Serial-to-USB Cables
[2] uC: Micro-Controller
[3] On-board
[4] SMD module

TABLE 17
Hardware Versions

Version	Logic Processor	ADC	Stimuli pins	Resolution [1]	Refresh rate
F-1	Microchip + FPGA	24bit single channel + MUX input	FPGA I/O	32x32	20
F-2	2x FPGA	24bit single channel + MUX input	FPGA I/O	128x128	40
F-2.1	2x FPGA	24bit single channel + MUX input	FPGA I/O	128x64	40
F-3	FPGA	2 x 24bit 16 channel ADC	32ch MUX	32x32	50
F-3.1	FPGA	2 x 24bit 16 channel ADC	FPGA I/O	32x32	50
F-4	FPGA	2 x 24bit 16 channel ADC	FPGA I/O	64x128	25
M-1	Microchip	Integrated 12 bit 32 channel ADC	Microchip I/O	32x32	40
M-1.1	Microchip	Integrated 12 bit 32 channel ADC	Microchip I/O	32x32	40
M-2	Microchip	Integrated 12 bit 9 channel ADC	Microchip I/O	20x9	40

[1] The resolution is written as active electrodes x passive electrodes. Thus 128x64 and 64x128 are different.

can be further scaled up to N-by-128 (N is a multiple of 32) without decreasing the scanning rate.

Data Transmission can be the bottleneck when the amount of *Active Electrodes* and *Passive Electrodes* scales up. Since they form a matrix, the increase is on the level of power of two. While typical in computer science for sending sensor prototype data to a computer usually use the simple serial port, it does not have sufficient data bandwidth. In this work, several data transmission methods have been implemented and thoroughly tested, including: Bluetooth Classic, multiple Bluetooth Classic to one *Receiver Device*, quad serial ports through a single USB, 254FIFO over USB highspeed.

7.2 Portable and Wireless Implementation

As mentioned in Section 1.2 and to the best of our knowledge, there is no existing small form factor that is suitable for hand-held (size of a smartphone or less) and completely wireless electronic systems. To enable a wider range of wearable application that is suitable for high mobility (such as sport exercise tracking) and mobile furniture (such as office swivel chairs), a system for rapid application prototyping needs to consider the following aspects:

- comprehensive power management with charging, and battery protection
- stable, high-speed wireless data transmission
- sufficient Analog-to-Digital converter channels and sampling rate
- sufficient fast switch output pins for offering stimuli at the active pins.

Typically, the sensing area of on-body sensors ranges from as small as 1-by-5 (e.g., one finger) to 16-by-8 (e.g., a stripe across big muscles such as chest or quadriceps) for individual sensor patches, a patch that covers the back of a jacket or a chair could also be 32-by-32. Therefore, after several attempts, it has come to the conclusion that an ideal solution is to build the system around a microprocessor with embedded ADC, dsPIC33FJ256GP710A is chosen as the center of the portable implementation.

7.2.1 Early prototpyes

The early wearable prototypes are implemented with small FPGAs and dedicated 24 bit ADCs. This allows the best possible analog resolution to initially investigate how precise is enough for the activity recognition goal. An example is Version F-3 and F-3.1. F-3.1 is used in the Leg Band study in Chapter 5, from which, a step was taken to truncate the 24-bit data to 12-bit, yielding similar recognition results. This then further confirms that 12-bit ADC is sufficient for wearable applications. As 12-bit ADCs are integrated into many power efficient micro-controllers.

7.2.2 Milestone Version

The milestone version M-1.1 has a dsPIC33FJ256GP710A as the logic controller and analog conversion. It is powered by a Li-Po battery, with USB 5V charging and under voltage (3V threshold) cut-off protection. An inertial measurement unit IC MPU-9250 is added for possible sensor fusion, connected to the micro-controller via I2C. The data from the micro-controller can be transmitted via a UART-USB bridge (FT232HL) (up to 92KB/s) or via Bluetooth Classic (RN42) (up to 30KB/s). A microSD card reader is also implemented on the board, driven directly by the micro-controller. To prevent ESD damage to the hardware, ESD protection arrays (TPD8S009DSMR) are designed to protect every IO pin. For indicating the working status, there is an RGB LED, every channel is connected to a pulse-width modulation output pin. Therefore any color can be instantly displayed on the LED indicator. The firmware is written in C, applying the firmware flowchart in Fig. 99.

There are 32 analog input pins (ANx), all of them can be re-purposed as digital I/O; there are another 32 digital I/O pins (DOx). Since the ANx pins can be changed to drive active electrodes of the sensor matrix, this board can also support very long matrix structure, for example, a matrix of 2 ANx and 62 DOx, which can be implemented on a belt or seat belt.

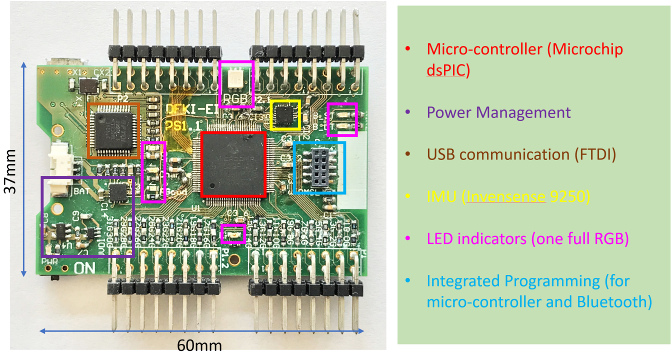

37mm
60mm
Micro-controller (Microchip dsPIC)
Power Management
USB communication (FTDI)
IMU (Invensense 9250)
LED indicators (one full RGB)
Integrated Programming (for micro-controller and Bluetooth)

Fig. 97. Hardware Version M-2.1 (front)

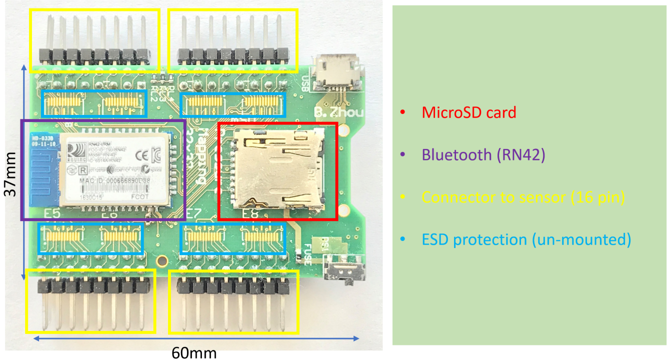


Fig. 98. Hardware Version M-2.1 (back)

7.3 Large-scale implementation

For the large-scale direction, the goal is to create a solution that can be driving not only a single large piece (e.g., sports mat sized, carpet sized or bed sized) matrix, but also possibly room-sized TPM matrix.

7.3.1 *Early Prototypes*

The first problem comes to drive a large number of active electrodes and passive electrodes. At the early stages of this work, active electrodes are driven by analog switches as in Version F-2 Fig. 101. Every switch is controlled by an FPGA Input/Output (I/O) port. It was soon concluded that the raw FPGA I/O ports achieve a similar effect as analog switches. Thus the use of costly analog switches was dismissed in further versions.

For driving more active electrodes, another approach was investigated to use bi-directional 1-to-N analog multiplexers (MUX), because to control which port the MUX is routing takes much less digital I/O than N. In Version F-3, a 32-to-1 channel MUX routes one single power output to multiple external active electrodes. However, the resulting hardware has a strong ghosting effect. Take the indexing range from Fig. 12 for example, when two different pixels G/26 and I/28 are being triggered, the pixels G/28 and I/26 will appear to be triggered even when they are not. This is possibly because when the 32 channels in the MUX are not selected, the circuit has a varying high impedance. However, from the simulation, it is ideal that the un-powered active electrodes to be pulled to ground. Therefore the Version F-3 is modified to Version F-3.1, using an FPGA to power the active electrode to avoid such ghosting effects. And it is not recommended to use analog MUX for driving active electrodes.

For driving a large number of passive electrodes, the first solution candidate is to combine a 32-channel MUX with a single channel ADC. Version F-2 and F-2.1 are implemented with this solution. Since usually, ADC operates at a faster sampling rate when it is set to automatic continuous sampling, the MUX should switch the channels just at the time when ADC finishes sampling. The firmware needs to switch the MUX channel and then read the ADC sample-conversion result. This allows one ADC-MUX pair to sample multiple channels at a fast rate, in Version F-2 and F-2.1, every pair samples 128-by-32 of the virtual surface at 40Hz, which is essentially 163,840 samples per second (sps).

7.3.2 *Milestone Version*

However, the ADC-MUX combination has a problem that the MUX usually is not switching at the exact time between ADC samples. This will result in that the previous pixel's value will influence the next pixel. On the resulting mapping, this appears as a 'shadow' of the actual object. To solve this, the alternative is using native multi-channel ADCs. However, such ADCs normally have less sampling rate. In Version F-3, F-3.1 and F-4, ADS1258 is used, which is one of the fasted 16-channel ADC available. Yet the resulting hardware scans a 32-by-16 area at 50Hz or 64-by-16 area at 25Hz, which is 25,600 sps, much less than the 163,840 sps with the ADC-MUX combination. But since ADS1258 simplifies the implementation and, it communicates with SPI, and thus requires less timing controlling and fewer pins than the ADC-MUX combination, because the MUX requires another bundle of pins to select the active channel. Therefore ADS1258 has become the more favorable choice in the later hardware versions.

7.3.3 *Scaling*

When designing even larger surface areas, the major limiting technical factors are the ADC channels, sampling rate, and the transmission data-bandwidth. The cost, refresh rate, and power consumption is the trade-off factors that should be considered. For example, the ADS1258's limiting factor is the 25,600 sps sampling rate. Thus to let the active electrode dimension to increase from 64 in Version F-4, while keeping the same 25Hz sampling rate, the simple solution can be using only 8 channels instead of 16 channels for every ADC. However, this will make the 8 ADCs in Version F-4 only sample 64 passive electrodes instead of 128. Then to make up for the loss of passive electrodes, 16 ADCs are needed

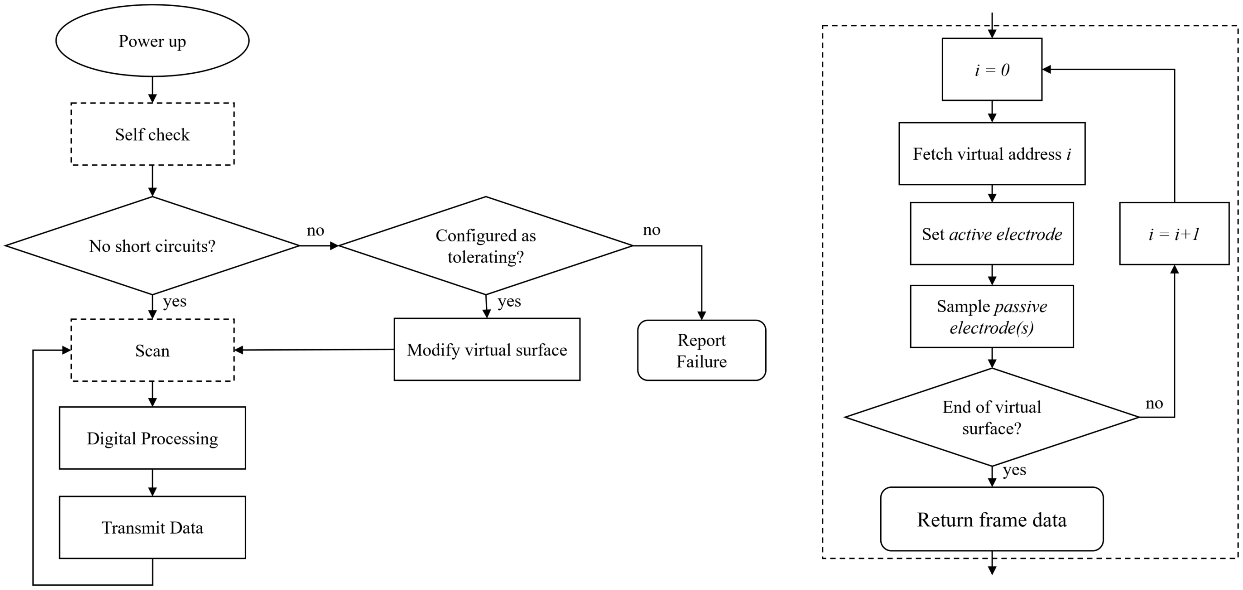


Fig. 99. Simplified firmware flowchart

instead of 8. This would then double the cost, analog power consumption, as well as the SPI lanes connected to the controlling FPGA.

Another scaling direction is on the modular system level, that multiple devices can be connected to a single computer at once. This will then require considering the actual sensor fabrication, especially the wires from the matrix to the electronics, because the cannot be overlapping the sensing area, which will block the pressure from going onto the sensor. Take equipping an entire room or hall, for example, for such occasions it is inevitable to have multiple systems, each taking care of a small area of one or two square meters. Fine spacial granularity may not be necessary when the overall area is big, thus using a larger pitch is also a solution to have one system covering more area. If it is not critical to have a seamless sensing area, software interpolation can be investigated to cover the gaps, such as predicting movement trajectory when the objects or persons are out of the sensing area. Normally those wires can be placed at one side of the matrix on each direction, then four matrices can be placed together, with the side without wires facing each other, to form a larger system.

7.4 Virtual Surface

Up to now, the pressure sensing matrix is assumed as a two-dimensional, rectangular shape. However, the principle can be applied in more versatile ways. Hence here the concept of the *virtual surface* is introduced. Fig. 106 shows an example of the virtual surface configuration of a complex touch-sensitive glove, with a dedicated sensor patch on every finger and the palm. From the perspective of the hardware board, there are only two types of pins connected to the sensing fabric - *Active Electrodes* and *Passive Electrodes*. They virtually form a two-dimensional matrix, every point on the same row or column should be electrically connected together. This virtual matrix is however only the possibilities of how the sensing electronics can be connected with the ECPC (electrically conducting polymer composites) physical sensor construct; it is also possible that any arbitrary points from the virtual matrix are connected to the physical sensor. The framework in this dissertation treats every possible virtual surface as a generic coordinate combination, instead of a 2-D matrix.

One major advantage is the prevention of cross-talk in designs where cross-talk between different sensing area is ideally isolated. Take the glove, for example, every finger is physically separated, joining only at the palm. As explained in [44], every sensing point in the matrix directly influences the sensing points that are directly connected to it in the same row and column. Therefore, if the fingers share the rows and columns, touch on one finger can cause a slight change in the adjacent fingers. When planning the configuration of the virtual surface, different patches can be separated to prevent such cross-talk caused by points sharing the same row or column. In Fig. 106, assume the columns (A, B, C, ect.) are connected to *Active Electrodes* and the rows (1, 2, 3, etc.) to *Passive Electrodes*. For the green finger A1-A4 and the blue finger B5-B9, Active Electrode A drives four sensing elements while B drives five; while this layout has a small scale, it shall apply to scaled-up sensor designs.

In the separated fingers example, there is also a difference in choosing either use active electrodes or passive electrode for the common wire and to each individual sensing node on the same finger. Fig. 107 shows simulation schematics and the voltage values of two cases from two fingers (blue and green) from Fig. 106 using active electrodes for the columns in one case (a and b), and for the rows in another case (c and d). One finger has 4 (a and c) and the other has 5 (b and d) sensing nodes. Fig. 107-a and

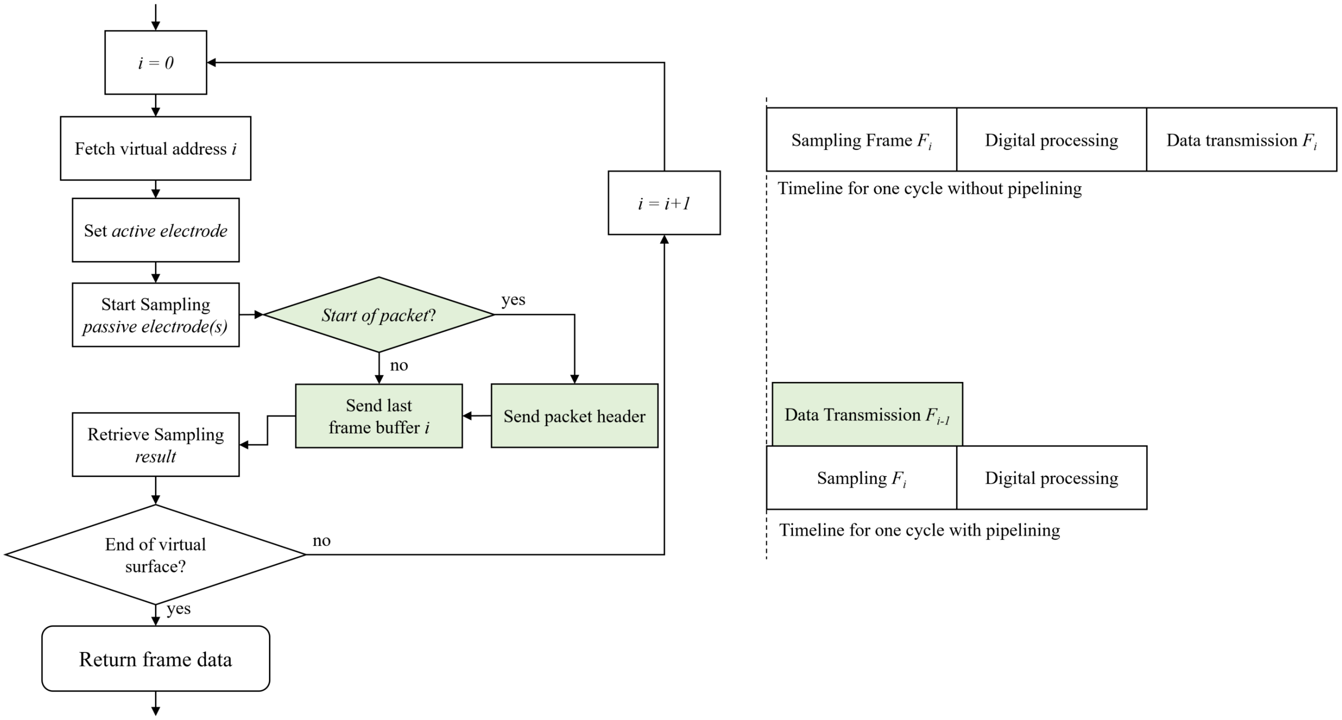


Fig. 100. Firmware pipelining

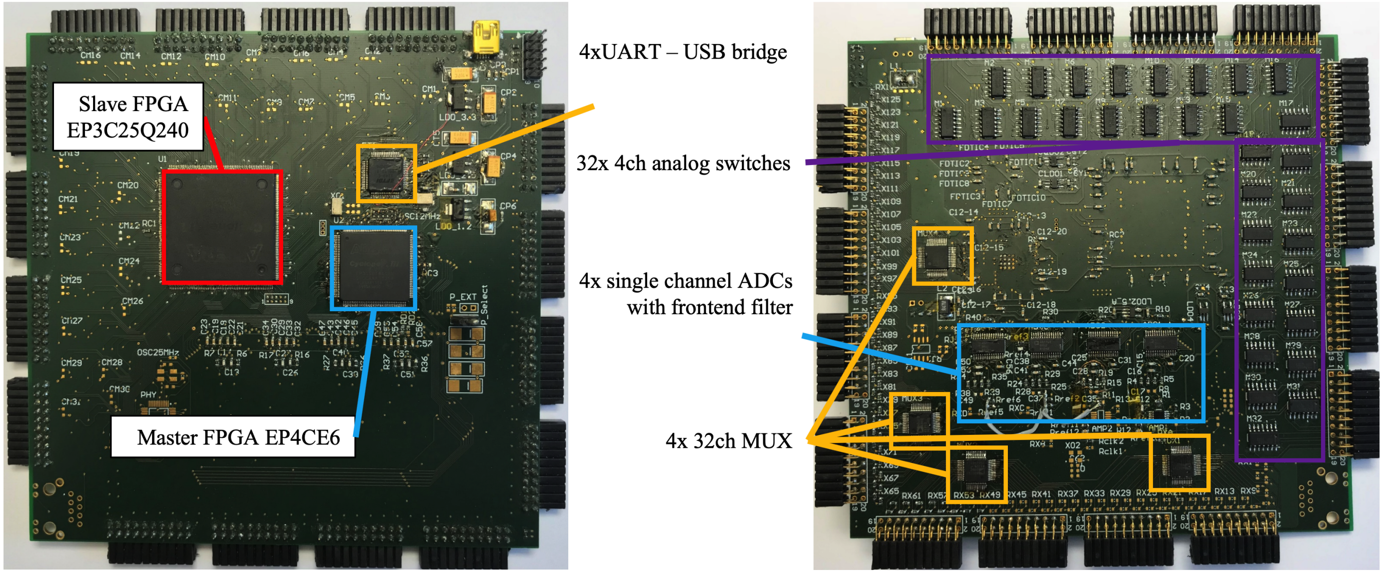


Fig. 101. Hardware version F-2 PCB

Fig. 107-b together show that when the individual nodes are separately connected to active electrodes, the change of one sensing resistor does not influence the rest of the sensor patch, and the unchanged sensing nodes have the same voltage value as the default values. And the same change in resistance returns the same change in voltage, regardless of the one more sensing node in Fig. 107-b. While in Fig. 107-c and Fig. 107-d, the unchanged resistors have a different voltage value from the default value, caused by the changed sensing resistor. Also even though in both Fig. 107-c and Fig. 107-d, the triggered sensing resistors changed the same values, but have different voltage values.

We can measure the quality of the output by the following formulas:

the normalized absolute contrast

$$\frac{V_{sense} - V_{default}}{V_{CC}}$$

or the normalized relative contrast

$$\frac{V_{sense} - V_{remain}}{V_{CC}}$$

V_{sense} is the voltage value of the sensing node which the resistor is changed, and $V_{default}$ is the voltage when no

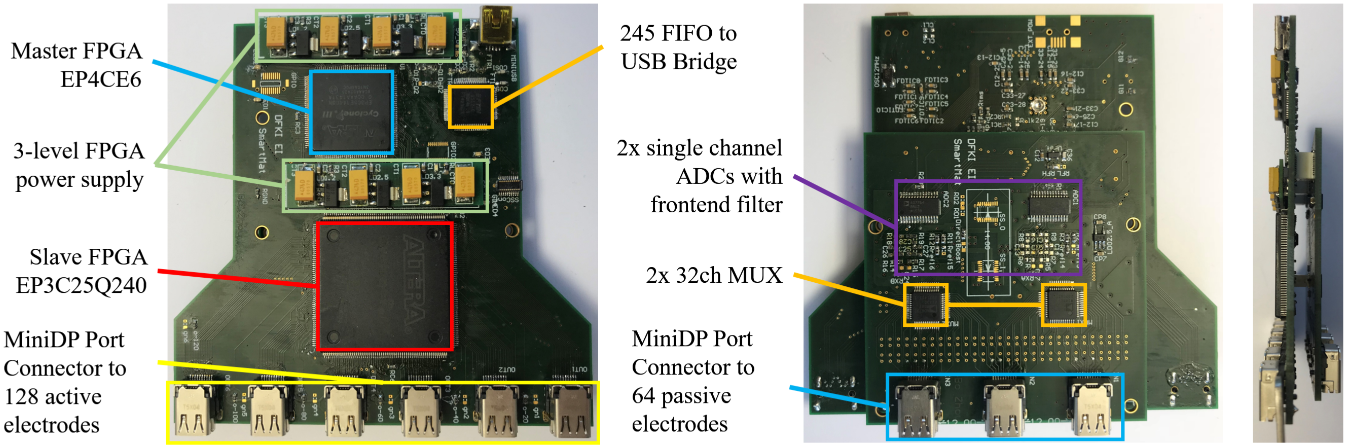

Master FPGA EP4CE6
3-level FPGA power supply
Slave FPGA EP3C25Q240
MiniDP Port Connector to 128 active electrodes
245 FIFO to USB Bridge
2x single channel ADCs with frontend filter
2x 32ch MUX
MiniDP Port Connector to 64 passive electrodes

Fig. 102. Hardware version F-2.1 PCB: from left to right: digital module, analog module, side view.

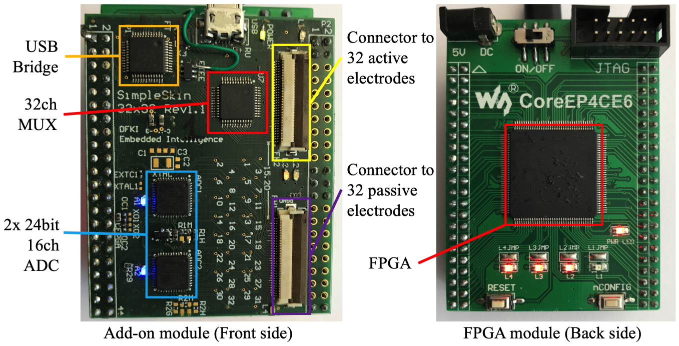

USB Bridge
32ch MUX
2x 24bit 16ch ADC
Connector to 32 active electrodes
Connector to 32 passive electrodes
FPGA
Add-on module (Front side)
FPGA module (Back side)

Fig. 103. Hardware version F-3 PCB

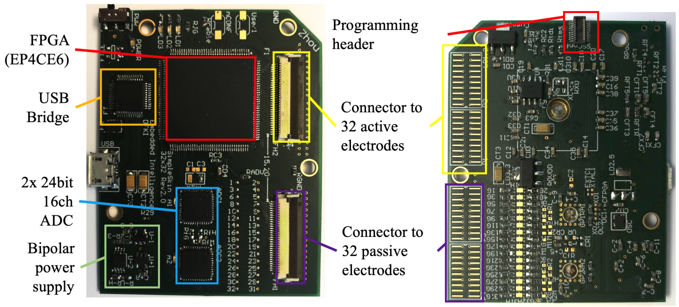


Fig. 104. Hardware version F-3.1 PCB

node is changed. V_{remain} is the voltage of the sensing nodes without change, when some other nodes are changed (i.e. the background of a pressure profile). Both of these two formulas should be accompanied with how much the sensing node has been changed ΔR_{Sense}. Those normalized contrast measures indicate how the end readout shows the pressure mapping. The absolute is compared with when there is nothing pressing the entire surface; and the relative describes how the changed sensing points 'pop out' compared with the rest of the surface.

In the case of Fig. 107 $\Delta R_{Sense} = 0.5K\Omega$ for node A1 and B5. In Fig. 107 a and b, both the normalized absolute and relative contrast is 0.167. This means for the same ΔR_{Sense} both contrast measures are stable regardless of the number of connected sensing nodes, when they are connected to a single active electrode and have individual passive electrodes. For c, the normalized absolute contrast is 0.133 and normalized relative contrast is 0.167; for d, the values are 0.119 and 0.143. This shows that the contrast becomes unstable if there are multiple sensing nodes on one passive electrode and multiple active electrodes, and the contrast becomes smaller as the number of nodes increases.

This suggests that even with the same physical sensor design, changing the passive electrodes and the active electrodes will also have an influence on the signal quality.

7.5 Further Remarks

There have been a positive development in the open-source, generic microcontroller market, mainly oriented at prototyping and hobbyists, such as the Arduino (arduino.cc) and Mbed (mbed.com) platforms. These prototyping boards can also be used to power smaller scale TPM sensors. They usually have many built in features such as wireless modules, power and battery management, indicator LEDs, etc. They also have programming suites that are aimed at fast and efficient coding. For example, an Arduino Feather HUZZAH32 with its 21 GPIOs, out of which 13 can be configured as analog inputs to its embedded 12-bit ADC, can power TPM matrices up to a size of 13-by-8. The only extra modifications needed are resistors for the sampling voltage dividers.

However, one disadvantage should be noted that, when using their simplified C programming environment, it is not possible to implement firmware-level pipelining to improve the data and sampling throughput such as the one shown in Fig. 100.

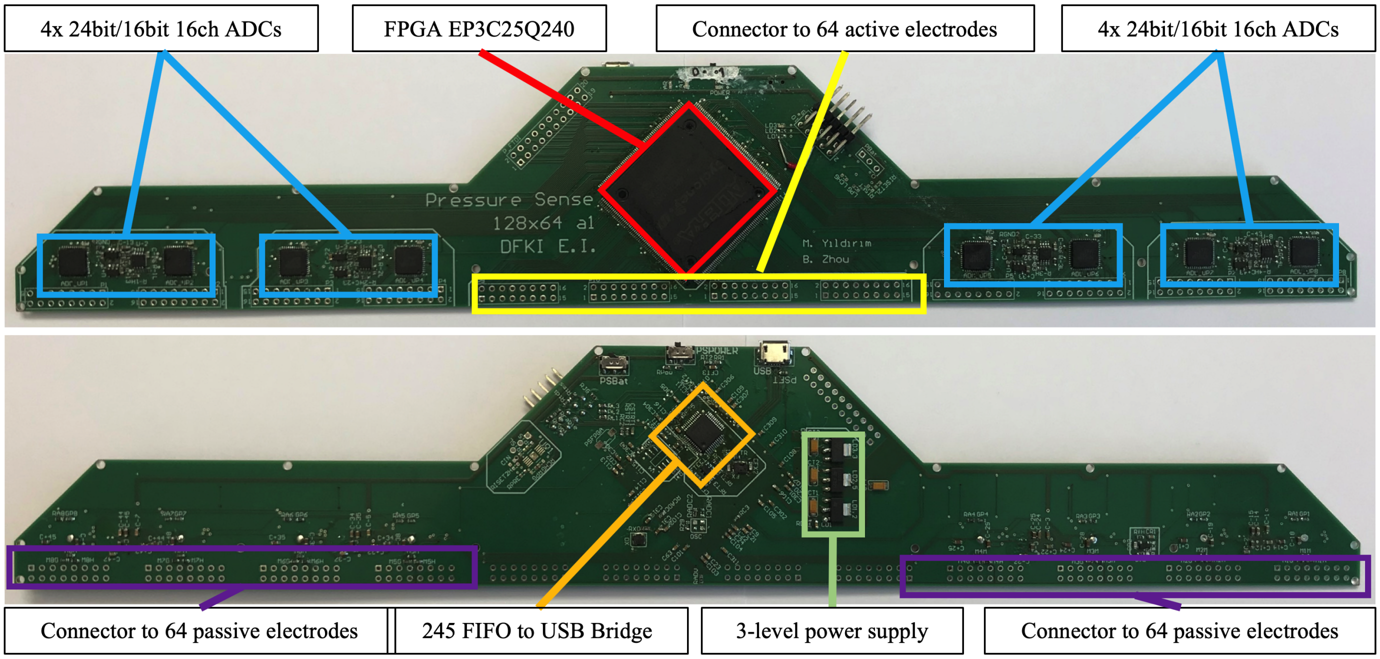


Fig. 105. Hardware version F-4 PCB

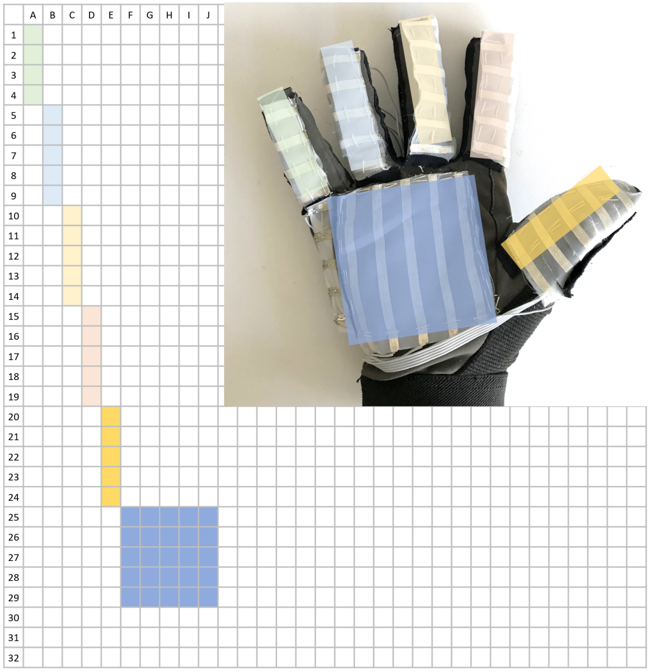


Fig. 106. Virtual surface of a smart glove

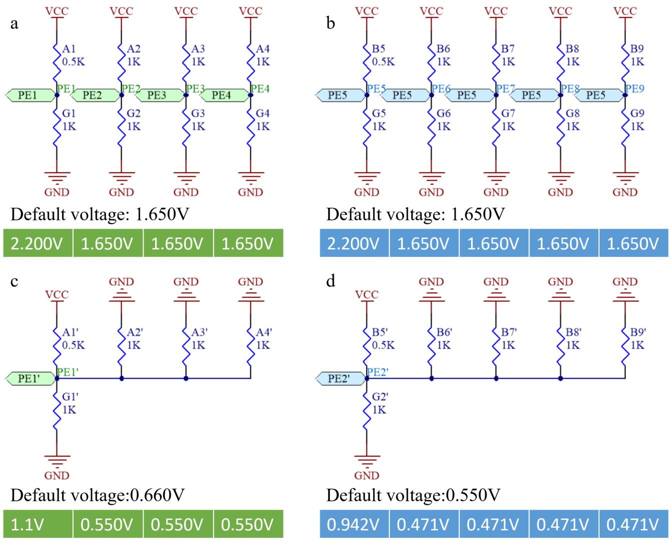


Fig. 107. Circuit model for swapping the active and passive electrodes examples. PEx are terminals connected to passive electrodes, and the colors match the virtual space configuration in Fig. 106; Ax, Bx match the row and column indexes in Fig. 106. The simulated scenario is that only $A1$ and $B5$ are changed equally to $0.5k\Omega$ and the others remain $1k\Omega$. Default voltage means the voltage of scanning every sensitive node when all the sensing resistors are equal to $1k\Omega$.

Chapter 8 Universal TPM Data Mining Algorithms

Data processing and data mining algorithms of the pressure mapping data are discussed in this chapter. Since the general data has a spatial-temporal format (a temporal sequence of the spatial pressure distribution), many data processing methods are present in all applications. However, in different applications the data present diverse properties, some special applications would require specific steps. For example, as in the studies of *Table Cloth* (Chapter 3.1) and *Smart Mat* for gait analysis (Chapter 3.3), objects cast distinguishable shapes inside a larger area, thus spatial segmentation may be required. While in the *Leg Band* study the shape makes less sense because all of the points are being pressed; but due to possible sensor shifting, a region of interest is selected based on the most active points prior to feature calculation. At its root, in the former case the sensing area is a planar surface bigger than, or matching the moving objects and people, and the latter, the sensing area is smaller than the objects and body parts, as illustrated in Fig. 15.

This chapter will start with the generic algorithms that are shared across various empirical aspects and then go into some specific algorithms dealing with special application tasks. Details of the empirical studies are already explained in previous chapters. This chapter takes the datasets from the empirical studies for evaluation and discussion. This chapter, in general, discuss the algorithms on a mathematical level as unified methods; as the implementation for the software toolkit are chosen is already discussed in Chapter 6 .

Structure of this chapter:

- Introduction to the general format of the pressure mapping data and low level methods to improve the signal quality.
- Discussion of spatial pattern recognition and temporal pattern recognition. As activity recognition through pressure mapping is essentially a spatial-temporal domain problem. Information of both the spatial distribution at a specific time and the temporal motion during a time period should be considered.
- A generic method, the TPM feature set, to extract information from the space and time domains is explained. It offers a base set of spatial and temporal features.
- The spatial and temporal domains are discussed in depth to investigate which features are more important for recognition.

In the TPM feature set, one of the space and time domains needs to be trimmed down in order to eventually calculate features from the remaining domain. Depending on the eventual domain, the features are categorized as temporal features and spacial features. Through feature selection method, neighborhood component analysis, and evaluation with datasets from different applications, it is obvious that the relevant features are different based on the nature of the application.

When other developers and researchers are evaluating TPM technology in a novel application, they can directly use the algorithms, including the data processing and the TPM feature set for initial validations. A crude optimization can also be achieved by choosing the more relevant features with the methods described in this chapter.

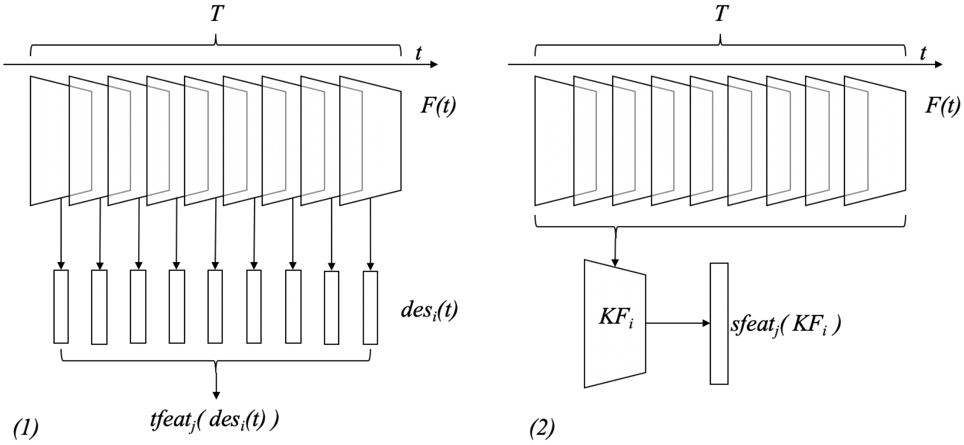


Fig. 108. (1) Temporal and (2) spacial feature extraction process.

8.1 TPM Data in the Spatial-Temporal Domains

TPM sensors generate a multi-channel, spatial-temporal data format, which describes the localization of the pressure distribution along time.

Every sensing point is defined as a *pixel*

$$p(x, y, t)$$

where x is the active electrode index, y is the passive electrode index, and t is the specific time.

At any time t, the entire mapping M of the sensor is defined as a *Frame*:

$$F(t) = \{p(x, y, t) \mid (x, y) \in \{M\}\}$$

A *Frame* may be a 2-D matrix, while it can also be an arbitrary array of points from the *Virtual Surface* as explained in section 7.4. A temporal sequence from a time window T of *Frames* is defined as a data *Stream*.

$$S_T = \{F(t) \mid t \in \{T\}\}$$

Individual sensing points have limited information about the activity; therefore some descriptive features are usually calculated as *Frame Descriptors*.

$$des_i(t) = Func_i(F(t))$$

Every des_i has its own calculation function and it is usually selected depending on the use-cases. Simple descriptors can be standard statistical descriptions of individual values of $F(t)$, such as mean value, standard deviation, kurtosis, skewness, etc. Since $F(t)$ is a 2D imagery in many cases, methods from image processing and computer vision can also be used to derive *Frame Descriptors* by leveraging the 2D nature of $F(t)$. For example, coordinates and (average) pixel value of points of interest or specific parts of the imagery can express the change of objects and pressure points.

Another approach to abstract the *stream* from a time window is to perform per-pixel operations along the time axis, resulting in individual frames that represent the stream. We call these frames as *Key Frames*:

$$KF_i(t) = Func_i(S_T))$$

8.2 Pre-processing

Before further data mining, some pre-processing methods can help improve the quality of data.

8.2.1 Row Offset

As mentioned in Chapter Chapter 2, Section 2.3.1, one changed pixel may reduce the values of the pixels of the entire row (including itself). The reduction is the same per changed pixel, and the reduction value depends on the changed pixel. In practical applications, it is common that multiple points are pressed, and in theory, the reduction should add up evenly. This phenomenon is more obvious in some hardware versions such as F-1, F-2 and F-3 from Table 17 and not observable in the other versions. If this phenomenon is present, it can be removed by subtracting the minimum value of every row from every pixel. This also brings the background of the imagery to zero, thus it can also be applied when the pixels-row phenomenon is not observable. One possible problem is that when an entire row is under pressure, the smallest pressure point would be brought down to zero. It can be compensated by conditions such as: whether the row minimum is not significantly smaller than the higher values from the row, or is significantly bigger than the other rows' minimum values, then it is possible that this is the case that the entire row is sensing pressure.

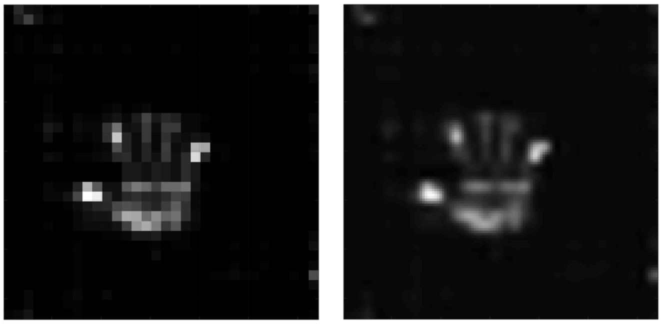


Fig. 109. Up-scaling of the pressure imagery.

8.2.2 Up-scaling

Up-scaling or super-resolution is a common technique in computer graphics and computer vision, by constructing a higher resolution image from a lower resolution image input. Since the TPM data per frame can be seen as an imagery of the pressure mapping, up-scaling can also be applied. For the observers, a higher resolution imagery can provide better understanding of the pressing objects' shape; for the further algorithms, super-resolution imagery can also help to increase the granularity. For example, when calculating the angle of a line between two pixels, finding the contour of a cluster of pixels, or for fitting eclipse algorithms, a greater resolution can provide more refined locations and angles. Through various studies, bicubic interpolation [174] for up-scaling has proven to be appropriate, since the super-resolution output expands the shape in all directions, in contrary to that nearest-neighbor interpolation only generates a square and pixelated version of the input imagery and bilinear interpolation generates a start-like result.

8.2.3 Thresholding

One particular merit of pressure profiling is that the background of the imagery, where there is no pressure, is of small value with minor fluctuations compared to the pressed parts of the sensor, which can be caused by weak contact of the fabric material or the weight of the covering material. Thus a simple process is used for completely removing the background. First, a threshold is defined, and the imagery is passed through the threshold to form a binary map $F_{binary}(t)$ that describes which pixel is above the threshold. If in the case when a majority part of the surface is the background, the threshold can be chosen by sorting the pixel values of a frame into several bins to form a histogram, and set the threshold depending on the upper limit of the bins containing the most counts. Such a threshold is dynamic. Alternatively, some initial trials can be made to decide a static threshold empirically for separating when there is an external force. The background can be removed by the

Hadamard product of the original frame and the binary map after thresholding $F_{binary}(t) \circ F(t)$. The binary mapping $F_{binary}(t)$ that represents the contours of the pressure map is also useful in further algorithms.

8.2.4 Morphological Transformation

Sometimes one object may have several clusters of pressure points, the clusters are very close to each other but not connected. To connect those clusters, a simple morphological transformation can be performed upon $F_{binary}(t)$, by dilution followed by erosion with the same kernel size. The kernel size is usually empirically defined based on the application.

8.2.5 Sensor sliding and segmentation

To cope with sensor sliding, first of all, we can seek a specific point from a time window T of the data stream, such as the point with the most significant value or standard deviation value. For example, in the study of detecting leg muscle activity [61] such a point is located as it resembles the peak location point of the muscle motions. Then a spatial window ROI can be selected around this point, and then every frame is cropped by the ROI:

$$F(t) \mid t \in T = \{p(x, y, t) \mid (x, y) \in \{ROI\}\}$$

To segment the target in an excessive coverage design, blob detection from the binary mapping generated by thresholding can be used. to further remove small gaps between blobs, the binary mapping can also go through morphological transformation before blob detection. One major problem here is that such blob detection is carried out in a single time point t, thus for a time window T, the same object may have different shapes and positions of blobs in the stream.

8.2.6 Normalization and Colormap

TPM sensors are not particularly fit for measuring absolute force values, but suitable for measuring the distribution and the dynamic change of such pressure distribution. Thus normalizing the frames is a common technique. Normalization follows the simple equation:

$$Norm(F) = \frac{F - \mu}{\sigma}$$

Normally μ is the average of F and σ is the standard deviation of F. However, when there is no point triggered and the sensor is occupied by the background noise, this normalization will generate an exaggerated noisy frame. Therefore it is necessary to empirically determine a threshold of σ of a minimum standard deviation of F, when the standard deviation of F is below this threshold, it is taken as a noise filled background and this threshold is used as σ for normalization.

For visualization, and many computer vision algorithm implementations and neural network implementations take a color or grey scale image, instead of a matrix of any range of integer or float types, converting the frame to a color image with a certain colormap is necessary in such cases. However converting to a colormap may introduce losses of data accuracy, since gray scale is normally 8 bit, or 0-255 integers, and RGB colors are three 8-bit channels, compared to 12 bit, 16 bit or 24 bit ADCs that are normally used in this work. The usual strategy is to use an RGB colormap with linearly increasing intensity for visualization, and convert the RGB frame to grayscale for image processing. Another possibility to improve the utilization of the colormap space is to introduce a nonlinear colormap made of two linear parts in combination with the thresholding introduced above: from the background noise value to the lower end of the threshold from thresholding, a wide range of sensor values correspond to a small range of colors; from above the threshold to the maximum of the frame, the sensor values are given the remaining of the colormap space.

8.2.7 Fault Tolerance

During actual usage, TPM sensors can experience two types of sensor failures: shorted pixel and open line. Shorted pixel happens when the top and bottom electrode of a pixel pierce through the middle ECPC layer. For the fine-mesh ECPC CARBOTEX, this failure is temporary and can normally be fixed by properly shaking and separating the layers; for the polymer film based ECPC materials, this failure can be permanent. Open line failure is because the electrical connection from a line electrode to the electronics circuit hardware is broken. It can be fixed by opening the wire routing part of the hardware and securing the connection.

Temporary or not, such failures can be bypassed on the software and ensure the operation for the rest of the surface. Shorted pixel behaves as the faulty pixel always at a saturated value and the other pixels on the same passive electrode at very low values, similar to the simulation result in Fig. 12. Open line behaves as either the disconnected active or passive electrode line gives no change on external pressure and the pixels of the line usually have values close to zero. When failure occurs, the affected line cannot be used to generate information anymore. A self-diagnosis procedure such as the following can remove the affected part of the matrix and continue normal operation:

1) accumulate several consecutive frames;
2) look for shorted pixels: a saturated point within a almost zero passive line;
3) look for open lines: a passive line or active line without any change and have almost zero values;
4) remove the faulty lines, replace the values with a spacial filter with the kernel (assuming the line is on the horizontal dimension):

$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} or \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$$

After the faulty lines and pixels are located, step 4 of the above procedure needs to be performed on every frame during the sensor's operation.

8.3 the TPM Feature Set

Fig. 108 shows the general workflow of calculating the TPM feature set from the space and time domains. Temporal features are extracted from sequences of simple frame descriptors $des_i(t)$. Spatial features are calculated from 2-dimensional key frames $KF_i(t)$. The initial version of the

TPM feature set includes 663 (17 × 39) temporal features and 80 (8 × 10) spatial features.

8.3.1 Frame Descriptors

Temporal features are extracted from sequences of simple frame descriptors $des_i(t)$. Treat $F(t)$ as a set, the TPM feature set calculates the following $des_i(t)$:

- average value

$$des_1(t) = mean(F(t)) = \frac{1}{|M|}\sum_{(x,y)}^{\{M\}} p(x,y,t)$$

- variance

$$des_2(t) = \frac{1}{|M|}\sum_{(x,y)}^{\{M\}} (p(x,y,t) - mean(F(t)))^2$$

- range

$$des_3(t) = p_{MAX}(t) - p_{MIN}(t)$$

- entropy

$$des_4(t) = -\sum_{(x,y)}^{\{M\}} p(x,y,t)\cdot log_2\, p(x,y,t)$$

- mean absolute deviation

$$des_5(t) = \frac{1}{|M|}\sum_{(x,y)}^{\{M\}} (p(x,y,t) - mean(F(t)))$$

- the center of mass (CoM) coordinate x and y (weighted by pixel value) $des_6(t)$ and $des_7(t)$
- the centroid coordinate (unweighted, only considering the contour after filtering the frame with a threshold). $des_8(t)$ and $des_9(t)$. Here the threshold is defined as

$$mean(F(t)) - 0.25\cdot(mean(F(t)) - p_{min}(t))$$

- area (the count of pixels that are above the threshold) $des_{10}(t)$
- $des_{11}(t)$ to $des_{17}(t)$ Hu's seven image moments [82]

For a matrix of binary values, the CoM is identical to the centroid; but for a matrix with multi-values that describes a profile, the CoM shows how the pixel value density is focused while the centroid shows only the geometric center. $des_1(t)$, $des_6(t)$ and $des_7(t)$ are mathematically identical to the first three central moments in the literature on image moments.

8.3.2 Temporal Features

Any sequence of frame descriptors is denoted as $des_i(t) \in \{des_1(t), des_2(t), ...\}$. Then from the temporal sequence within a window of length T (sliding window or spotted event), temporal features can be calculated:

- average

$$tfeat_1 = \frac{1}{|T|}\sum_{t}^{\{T\}} des_i(t)$$

- variance

$$tfeat_2 = \frac{1}{|T|}\sum_{t}^{\{T\}} (des_i(t) - tfeat_1)^2$$

- range

$$tfeat_3 = des_{i_{MAX}} - des_{i_{MIN}}$$

- skewness, that describes the asymmetry of the data

$$tfeat_4 = \frac{\frac{1}{|T|}\sum_t^T (des_i(t) - tfeat_1)^3}{\left(\frac{1}{|T|}\sum_t^T (des_i(t) - tfeat_1)^2\right)^{3/2}}$$

- kurtosis, that measures how outlier-prone the temporal sequence's distribution is

$$tfeat_5 = \frac{\frac{1}{|T|}\sum_t^T (des_i(t) - tfeat_1)^4}{\left(\frac{1}{|T|}\sum_t^T (des_x(t) - tfeat_1)^2\right)^{2}}$$

- waveform length [175] [176]

$$tfeat_6 = \sum_{t}^{T} -1(des_i(t+1) - des_i(t))$$

- sum of values greater than mean

$$tfeat_7 = \sum_{t}^{T} (des_i(t) \mid des_i(t) > tfeat_1)$$

- the power spectrum density of des_i is calculated with fast Fourier transform as $PSD(n)$, $n \in N$ is the frequency in the spectrum. Following features are calculated from $PSD(n)$: average magnitude

$$tfeat_8 = \frac{1}{N}\sum_{n}^{N} PSD(n)$$

- mean frequency

$$tfeat_9 = \frac{\sum_n^N n \cdot PSD(n)}{\sum_n^N PSD(n)}$$

- N is divided to 5 equal frequency bands, the average values of each band is $tfeat_{10}$, $tfeat_{11}$, $tfeat_{12}$, $tfeat_{13}$, $tfeat_{14}$.
- A wavelet transform scalogram is calculated with the LTFAT toolbox [129], with $J = 4$ filterbank iterations. The coefficient vector of each filterbank is $C(j), j \in [0, 4]$.
- $tfeat_{15}$, $tfeat_{20}$, $tfeat_{25}$, $tfeat_{30}$, $tfeat_{35}$ are the mean value of each coefficient vector;
- $tfeat_{16}$, $tfeat_{21}$, $tfeat_{26}$, $tfeat_{31}$, $tfeat_{36}$ are the variance of each coefficient vector;
- $tfeat_{17}$, $tfeat_{22}$, $tfeat_{27}$, $tfeat_{32}$, $tfeat_{37}$ are the range of each coefficient vector;
- $tfeat_{18}$, $tfeat_{23}$, $tfeat_{28}$, $tfeat_{33}$, $tfeat_{38}$ are the skewness of each coefficient vector;
- $tfeat_{19}$, $tfeat_{24}$, $tfeat_{29}$, $tfeat_{34}$, $tfeat_{39}$ are the kurtosis of each coefficient vector;

Even though frame descriptors and temporal features may have apparently similar equations, they are different concepts. The differences between temporal features and frame descriptors are that:

1) Relationship: temporal features are calculated from a temporal sequence of frame descriptors

2) Scope: temporal features are in the scope of an entire time window, while a frame descriptor is calculated from a single frame.
3) Dimension: for one time window, one method to calculate the feature of a single frame descriptor $des_i(t)$, there is one value of temporal feature. In a time window of size T, one frame descriptor is a function of time t and is a 1-dimensional array of length T.

8.3.3 Key Frames

Spatial features can be calculated from 2-dimensional frames. Since classifiers and non-convolution neural networks take features from a time window, spatial features need to be calculated from frames that are representative of its time window. These frames are called *key frames*.

From a time window, a *key frame* can be one particular frame that has special frame descriptor values such as the maximum or minimum of $des_i(t)$. A *key frame* can also be calculated from the stream of the window through pixel-wise operations. 8 key frames are calculated in the TPM feature set:

- per pixel average of all frames

$$KF_1 = \frac{1}{|T|}\sum_{t}^{\{T\}} F(t)$$

- sum of per pixel differences

$$KF_2 = \sum_{t}^{\{T-1\}} (F(t+1) - F(t))$$

- sum of only the positive or negative values of per pixel differences

$$KF_3 = \mid \sum_{t}^{\{T-1\}} ((F(t+1) - F(t)) > 0) \mid$$

$$KF_4 = \mid \sum_{t}^{\{T-1\}} ((F(t+1) - F(t)) < 0) \mid$$

- the frame which has the maximum mean pixel value as KF_5 and the frame with the minimum mean value as KF_6
- the frame with the maximum standard deviation from the stream as KF_7
- the per pixel average of the frames, whose pixel value is greater than the frame pixel average

$$KF_8 = \frac{1}{|T|}\sum_{t}^{\{T\}}(F_p(t))$$

$$F_p(t) = \begin{cases} p(x,y,t) & \text{if } p(x,y,t) \geq mean(F(t)) \\ 0 & \text{if } p(x,y,t) < mean(F(t)) \end{cases}$$

8.3.4 Spatial Features

Various image processing techniques can then be used to extract information from those key frames. Image moments are proven to be a helpful shape descriptor for spatial features. The commonly used ones are 3 central moments plus Hu's 7 invariant moments [82], which are rotation, translation and scale invariant.

8.4 Feature Selection and Cross-Validations

In this section, an evaluation of how different combinations of frame descriptors - temporal feature pairs, and key frame - spatial feature pairs contribute to the machine learning is performed. The datasets used are from various studies in different setting scenarios. This chapter uses a top-down perspective to investigate those datasets. For how different datasets are gathered, and their processing uniqueness are explained case-by-case in Chapter 3, Chapter 4 and Chapter 5.

8.4.1 General Approach

The evaluation process can be divided into four parts:

Part 1: convert the data stream into features. From the time domain, first, temporal sequences of the 17 frame descriptors $des_i(t)$ are calculated from every stream. Then within every $des_i(t)$, a sliding window is performed. Every window is denoted as $n \in N$. The data in the window is multiplied with a Tukey window with $r = 0.2$, to bring the start and end of the window to zero. Then for every sliding window, 39 temporal features $tfeat_j(des_i), j \in 1, 2, ...39, i \in 1, 2, ...17$ are calculated. In the spatial domain, first the input data stream is cropped with the same window size and window step as the sliding window for $des_i(t)$, but the outputs are the smaller length of streams, and no Tuken window is applied. Then within each window of streams, 8 key frames KF_i are calculated. Overall 10 spacial features is calculated from every key frame $sfeat_j(KF_i), j \in 1, 2, ...10, i \in 1, 2, ...8$.

Part 2: baseline cross-validation. To carry out balanced training, all classes are trimmed to the same amount of windows by random selecting. The amount of windows is the class that has the least windows. K-fold cross-validation is performed with multiple classifiers, and the accuracy is used to compare different classifier's results.

Part 3: feature selection. TThe feature weight evaluation is performed using neighbourhood component analysis (NCA) [177]. The method ranks the most relevant features that contribute to the classification. Since the features are calculated from two levels of information: temporal features are calculated first by reducing the space domain to frame descriptors, then to the time domain features; as spatial features are calculated first by reducing the time domain to key frames. Thus the feature weight result can either be presented as a *feature weight vector* or as a *feature weight matrix* for either the temporal or spacial feature methods.

Part 4: feature reduction. The top weighted features are selected to perform the same cross-validation. For comparison, the least weighted features are also evaluated separately.

Principle component analysis (PCA) [178] is another commonly used technique for reducing feature dimensions. The method removes redundancy and outputs a set of eigenvectors that best describes the variance of the dataset. Each component is orthogonal to the preceding one so that the eigenvectors are uncorrelated and thus without redundancy. However, PCA itself does not take the class label information, it only analyses the data distribution to remove redundancy but not irrelevant features. Typically PCA is used as a step after calculating the features, and

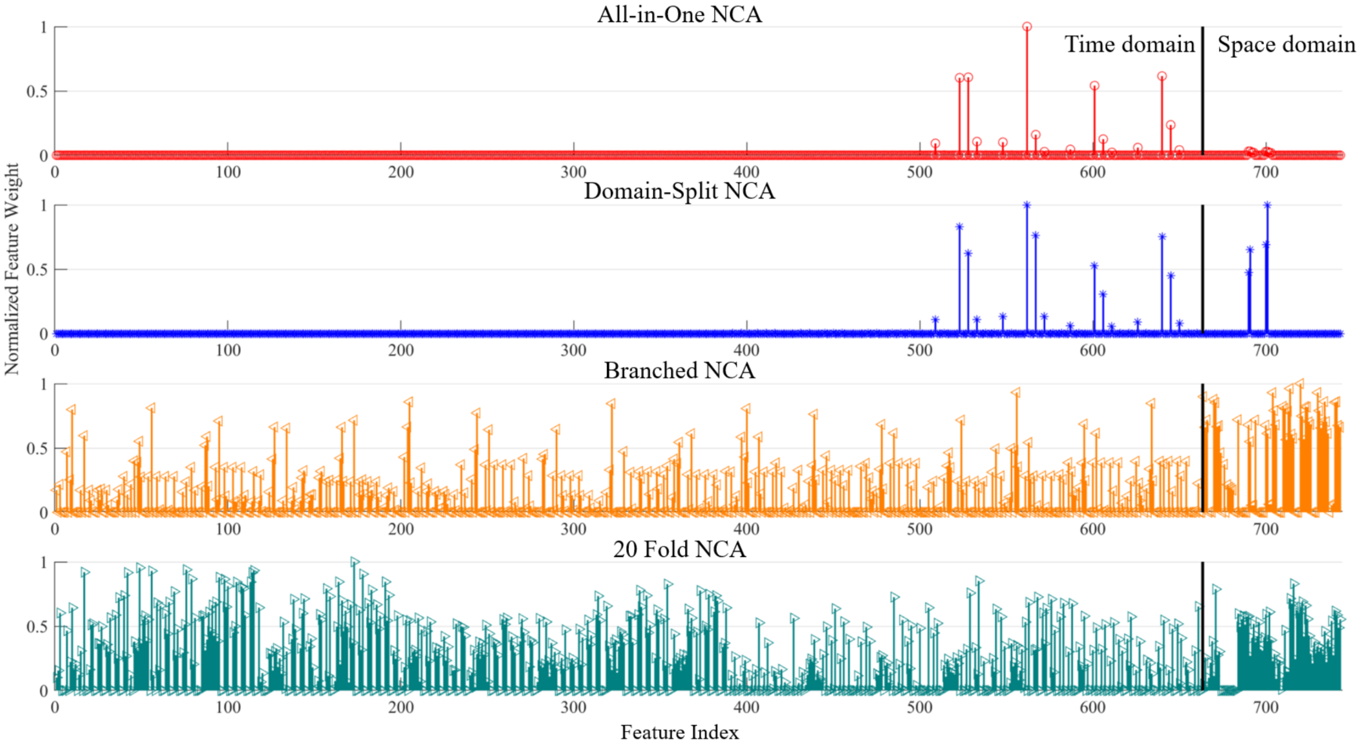


Fig. 110. Feature weight distributions of different NCA division methods (table cloth dataset)

before feeding the information to classifiers. Therefore, we use NCA instead of PCA to find the features that are more contributive to distinguishing different classes.

8.4.2 Datasets

3 past studies are taken for comparison, they are code-named as: *table cloth* [64], *robot skin* [62], and *leg band* [56].

In the table cloth study [64], a TPM fabric with a 30-by-42 matrix is placed on a dining tablet to detect dining related actions. A main dish plate, a salad bowl, and a glass are placed on it. Participants eat various food of different textures, that would require different actions for dining the food with a knife and a fork. The force of the actions can propagate through the cutlery and plates to the table cloth surface. The 7 action classes are: stir, scoop, cut, poke, scoop, collect and replace. The sliding window is chosen with 2 second period and 1 second window step. 10 participants each took part in 8 recordings.

In the robot skin study [62], a TPM fabric with a 20-by-20 matrix is used to detect 7 emotionally related touch gestures onto a dummy arm or a surface, including grab, poke, press, push, scratch, pinch and stroke. The gestures are already segmented based on matrix activation, since when there is no gesture, the matrix is not being pressed. In total, 24 participants took part in 2 recordings. Each recording includes 16 repetitions of every gesture.

In the leg band study [56], a TPM fabric with an 8-by-16 matrix is embedded in an elastic compression band that is placed on the thigh as users take part in gym leg exercises. The sensor detects the surface pressure of the leg muscles as planar pressure mechanomyography. The 5 activity classes are: working out with a cross trainer, leg press, seated leg curl and leg extension, plus a class contains all non-workout activities. Based on the activity's characteristic, the sliding window is chosen as 4 seconds wide, the window step is 20% of the window size. 6 participants have recorded 4 sessions each.

In this chapter, all the participants' data are merged together as one dataset per application (person dependent - inclusive case). Leave-one-out is carried out separately in the corresponding sections of Chapter 3, Chapter 4 and Chapter 5. Every sliding window or gesture is one sample. The table cloth dataset has 10815 samples, robot skin 5376 samples, and leg band 28425 samples.

8.4.3 Neighborhood Component Analysis (NCA)

The NCA method [177] assumes a feature weight vector w as a variable for the features, and use an approximate solver to find the optimal weight vector that maximizes the correct classification probability (the objective function). (In this subsection, the mathematical symbols are not related to the rest of this thesis.)

For a d-dimensional dataset of N training points, all the points from the training point are taken as a query point once x_i. For each query point, the other points can be taken as its reference point as a probability p_{ij} derived from their weighted distance enclosed in a kernel function. The probability that this query point x_i is correctly classified is then defined as the probability summation of the reference points that has the same class.

The objective function is then the average of all the points' correct classification probability. After unfolding the relationship, the objective function can be written as a differentiable function of the feature weight vector, with a

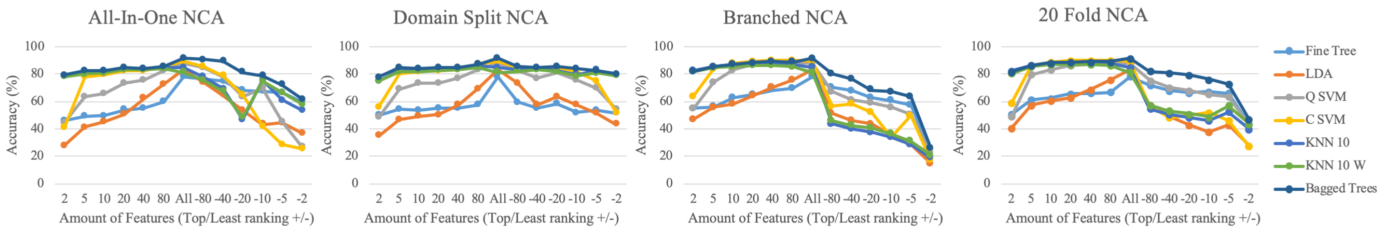


Fig. 111. Accuracy with varying amount of selected features comparison of four NCA division methods (table cloth dataset)

tunable parameter λ which is multiplied with the weight vector's term in the objective function:

$$F(w) = \sum_{i=1}^{N}\left(\sum_{j=1,j\neq i}^{N} P_{ij}y_{ij} - \lambda\sum_{l=1}^{d} w_l^2\right)$$

where $y_{ij} = 1$ if the query point and the reference point has the same class. Since $F(w)$ is differentiable, its maxima can be approximated with algorithms such as stochastic gradient descent (SGD) [179], to find out the feature weight vector w that maximizes the objective function $F(w)$.

For a certain NCA model, the λ value is fixed. However, a parametric sweep of λ can further improve the result by finding the λ that minimizes the *loss* of the NCA model. K-fold cross-validation splits the dataset into training to approximate the NCA model and testing to calculated the *loss*. The average *loss* value of all the K folds is the average loss for a specific λ.

8.4.4 NCA division approaches on high dimensional features

A problem of NCA is that when most of the features contribute to the classification, the approximation may return to only very few highly weighted features while the others remain close to zero weight. This leaves the classification result relatively low with selected high weighted features. Our solution is to segment these features and perform NCA on smaller batches, then combine the feature weights. Therefore, in this work, four NCA approach is investigated:

- All-in-One: all the features are taken under NCA as once.
- Space-time domain split: features are split into two groups: spatial domain features and temporal domain features.
- Branched: features are more detailed separated into branches. In the time domain, a branch is all the temporal features from one frame descriptor; in the space domain, a branch is all the spatial features from one key frame.
- K-fold: all features are randomized and split into K equal partitions. One NCA is performed for each partition.

In the segmented feature groups, the resulting feature weight vectors are normalized within each group before being concatenated into one vector. The results of the three different approaches on the smart table cloth data are shown in Fig. 110. From the result, All-in-One and domain-split NCA return similar weight for the time domain features. The domain split NCA gives higher weight on the spatial features as a result of normalization before merging, but the feature indexes that are higher than approximate zero are the same between the two approaches. In the branched NCA, however, many more features are given higher weight.

8.4.5 Feature Weight Criteria

To compare which approach is better, cross-validation with the highest ranking features, in comparison with the lowest ranking features can be used. A better approach should meet the following criteria:

- Higher accuracy with the same number of top ranking features compared to other approaches.
- Greater difference between highest accuracy and the accuracy with the least ranking features, than the difference between highest accuracy and the accuracy with the top ranking features.
- With the same amount of features, top ranking features should in general result in higher accuracy than least ranking features.

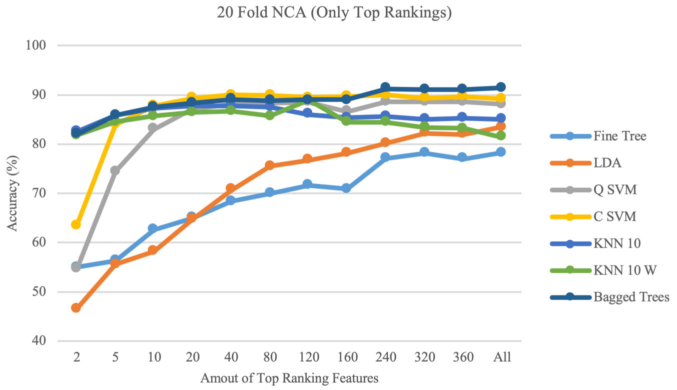


Fig. 112. Top ranking features of the 20 fold NCA on the table cloth dataset to locate the optimal amount of features.

8.4.6 NCA Evaluation

Every time, an incremental number of either the top or least ranking features are selected from the feature weights for 5 fold cross-validation. The accuracy of different validations are plotted in Fig. 111. Comparing four different approaches, all-in-one and domain-split NCA returns much fewer features with significant weight than the branched or 20-Fold NCA, as most of the features are weighted close to zero.

To evaluate which approach yields better feature selection, cross-validation from the top or least ranking features are performed. For performance reasons, top or least 2, 5, 10, 20, 40 and 80 features are chosen. The NCA algorithm should have greater influence on the KNN classifiers since the basic principle is similar (Euclidean distance to the training data samples). In this evaluation, a variety of classifiers are chosen:

1) classification tree with 100 maximum splits and Gini's diversity index split criterion (Fine Tree)
2) linear discriminant analysis (LDA)
3) support vector machine with quadratic kernel function (Quadratic SVM)
4) support vector machine with cubic kernel function (Cubic SVM)
5) K-nearest neighbor with equally weighted Euclidean distance and K=10 (Fine KNN)
6) K-nearest neighbor with squared inversely weighted Euclidean distance and K=10 (Weighted KNN)
7) Ensemble of 30 decision tree learners (Bagged Trees)

The results are shown in Fig. 111. For many classifiers, all-in-one and domain split NCA has a near symmetric accuracy distribution centered at all features; sometimes with the least ranking features, there are higher accuracy points than the corresponding top ranking features. From this, we concluded that the feature weights derived by these two methods are no better than random selection. Branched and 20 Fold NCA, on the other hand, in general, meet the criteria listed above, and have a similar trend of the accuracy values. The highest ranking features result in higher accuracy values than the lowest ranking features.

The top 2 ranking features already result in over 80% accuracy for Bagged Trees and the two KNN classifiers. While for the other classifiers, Fine Tree, LDA and SVM, the accuracy values are significantly lower. This may because these classifiers work by separating the feature space with modelled boundaries, while KNN and bagged trees do not use such boundaries to distinguish different classes. The data's nature may not fit very well with the classifiers' algorithms, e.g., the data may not have clean-shaped boundaries, or the same class may have several clusters. However, this cannot be further investigated at this point due to the high dimensionality.

The least 2 ranking features result in close to chance level (14.3% for 7 classes) accuracy values, thus means the NCA successfully identify the less relevant features. As the number of features taken grows, the accuracy of both top and least ranking features increase, but the top ranking features give higher accuracy than the least ranking ones.

As branched NCA is not a generic approach, and K-Fold NCA can be performed on any feature sets, this work will continue with K-Fold NCA. Fig. 112 shows the top ranking features but with more amount of taken features until all of them are chosen. From it, the accuracy has come to a stable level close to 90% between 10 to 160 features for most classifiers except for LDA and Fine Tree; while from 240 features on, the accuracy has another increase that is on the similar level with all the features. This shows that only the top 10 features are sufficient for this dataset for moderately high accuracy, and 240 features are adequate to explain all the class discriminant as good as with all features.

TABLE 18
Accuracy Comparison of the Original Studies and the TPM Feature Set

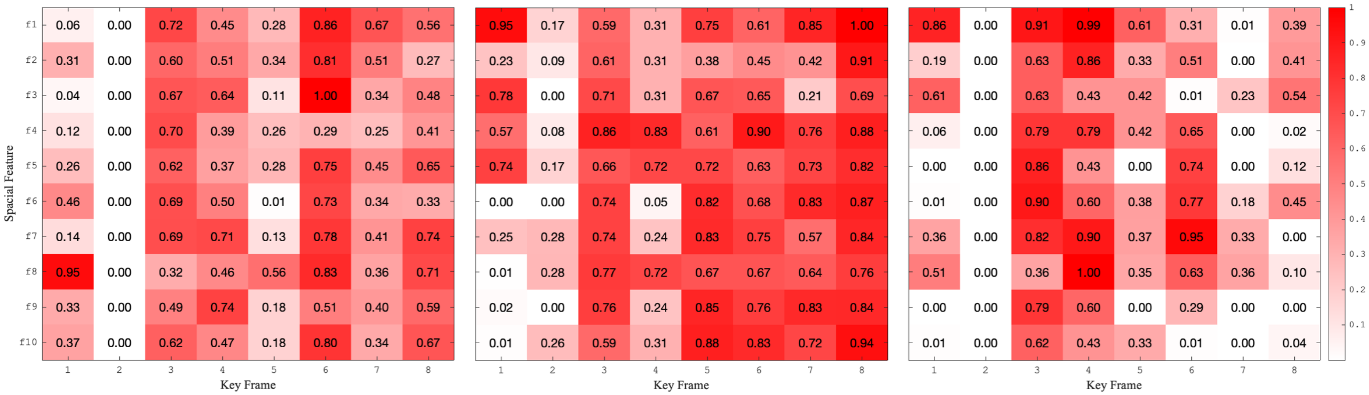
Dataset	Original Study	TPM Feature Set
Table cloth	91.2%	91.4 %
Robot Skin	92.7%	94.7 %
Leg Band	81.7 %	98.2 %

8.4.7 Application Variance

To be displayed only as a linear vector of values as in Fig. 110 is not sufficient to tell which feature calculation method is more relevant. Therefore, the feature weight vector is reshaped into two 2-dimensional matrices according to the frame descriptor - temporal feature combination or key frame - spacial feature combination as a feature weight matrix (FWM). For the table cloth dataset, the temporal feature weight matrix FWM_t is shown in Fig. 115, and the spacial feature weight matrix FWM_s is in Fig. 113(1). From FWM_t, it can be seen that some temporal features have no contribution such as skewness, kurtosis, including the skewness and kurtosis for the wavelet transform. Some frame descriptors are more important such as des_2 variance, des_3 range, des_5 mean absolute deviation. All the 7 Hu's moments des_{11} to des_{17} are less important. It is possibly a result that in this dataset, the objects are all plates or glasses, and their footprints are all circular. Hence the shape descriptors are not contributing to the activity. From FWM_s, the key frames describe the static values such as KF_1 and KF_2 are less contributive, while the key frames that describe the dynamic changes all have greater feature weights.

Two other datasets are evaluated with the same process, and the resulting plots of 'number of features' - accuracy plots are in Fig. 114 and Fig. 118. (SVM classifiers are not used for evaluation here due to performance constraints.) Referring to the criteria of NCA evaluation, NCA has effectively located relevant features in all of the datasets. Feature weight matrix are shown in Fig. 116, Fig. 117, and Fig. 113(2)(3). Table 18 compares the accuracy of the original studies with the TPM feature set. The top 20 features from each dataset are further listed in Table 19.

Comparing the FWM_t and FWM_s of all datasets, important features are very different for different applications. And there are only three features that are present in all datasets' top 40 features. For example, skewness and kurtosis have relatively high weight for the robot skin dataset, and also have higher weights in some of the frame descriptors for the leg band dataset. The FFT features have almost no weight in the robot skin dataset, while these features are significantly relevant in the table cloth and leg band dataset. And when FFT features have more weight, wavelet transform features also have more weight. In the leg band dataset, Hu's 7 moments as frame descriptors have significantly higher weight than the other two dataset. Spatial features on average have less weight in the table cloth and robot skin datasets than in the leg band datasets.



(1) Table Cloth Dataset *(2) Robot Skin Dataset* *(3) Leg Band Dataset*

Fig. 113. Spatial feature weight matrices of the three datasets.

TABLE 19
Top Ranking Features

Table Cloth dataset

Ranking	1	2	3	4	5	6	7	8	9	10
Des/KF	5	2	2	3	2	1	3	2	5	3
Feature	17	10	37	37	17	17	36	3	22	17
Domain	T	T	T	T	T	T	T	T	T	T
Ranking	11	12	13	14	15	16	17	18	19	20
Des/KF	3	3	14	5	5	3	3	6	10	3
Feature	1	20	27	35	8	35	27	3	3	22
Domain	T	T	T	T	T	T	T	**S**	T	T

Robot Skin dataset

Ranking	1	2	3	4	5	6	7	8	9	10
Des/KF	3	3	4	4	4	3	4	6	3	4
Feature	21	16	34	8	1	31	15	7	1	22
Domain	T	T	T	**S**	**S**	T	T	**S**	T	T
Ranking	11	12	13	14	15	16	17	18	19	20
Des/KF	3	2	4	4	4	3	4	3	3	4
Feature	7	1	27	39	16	1	7	6	15	2
Domain	T	T	T	T	T	**S**	**S**	**S**	T	**S**

Leg Band dataset

Ranking	1	2	3	4	5	6	7	8	9	10
Des/KF	2	7	6	5	5	15	8	15	9	8
Feature	32	37	22	17	22	35	31	32	25	1
Domain	T	T	T	T	T	T	T	T	T	**S**
Ranking	11	12	13	14	15	16	17	18	19	20
Des/KF	8	9	3	15	12	15	8	4	8	6
Feature	15	35	31	3	17	22	16	3	17	32
Domain	T	T	T	T	T	T	T	T	T	T

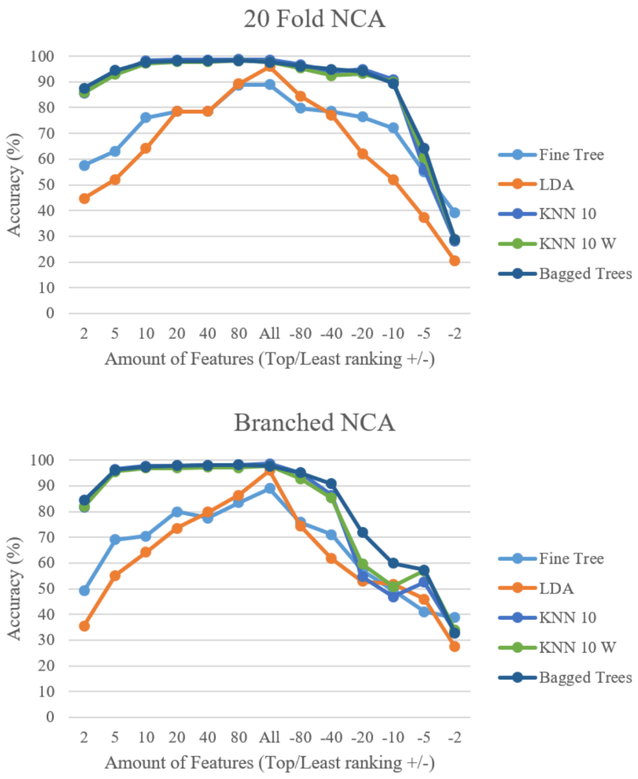
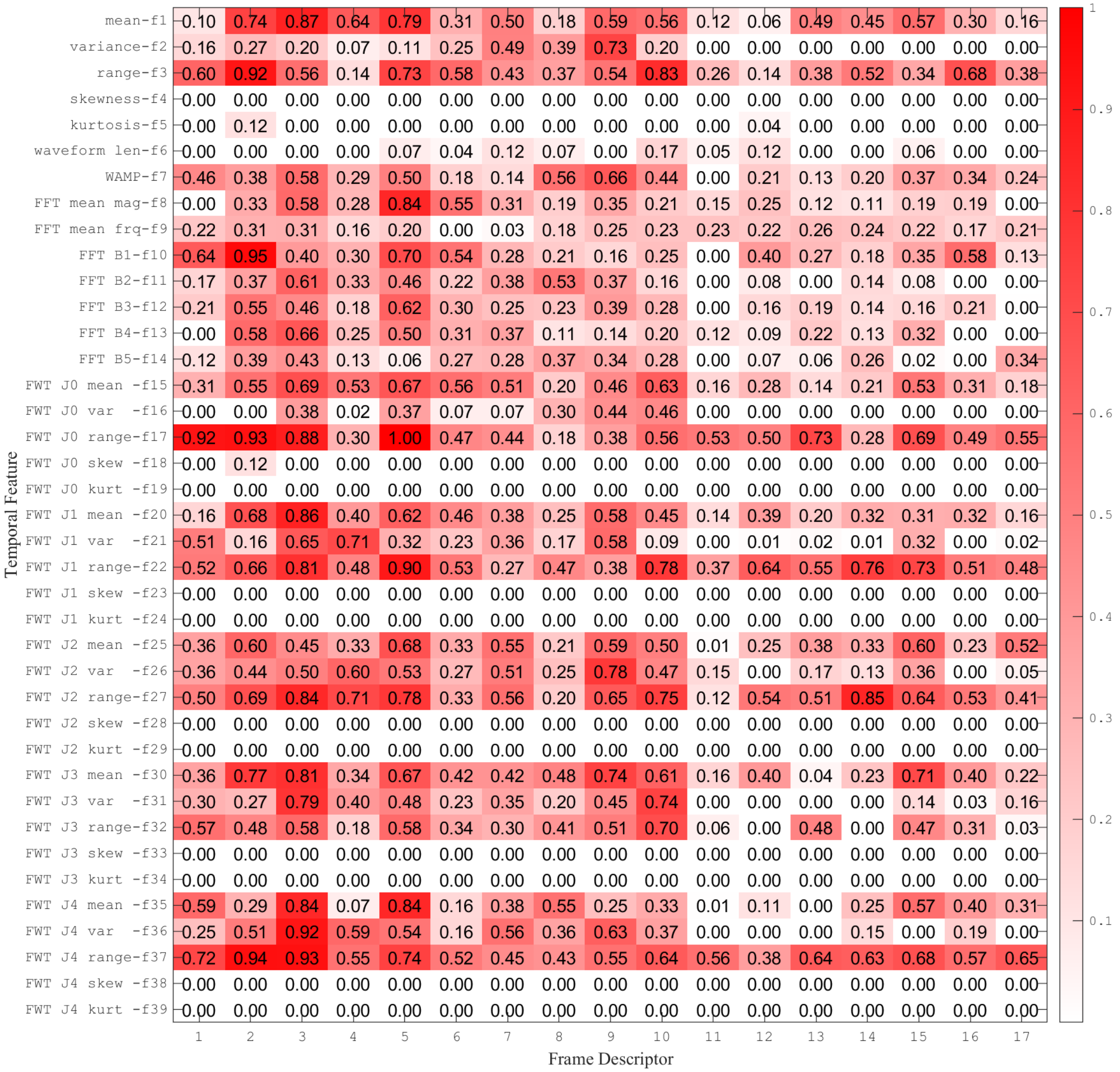
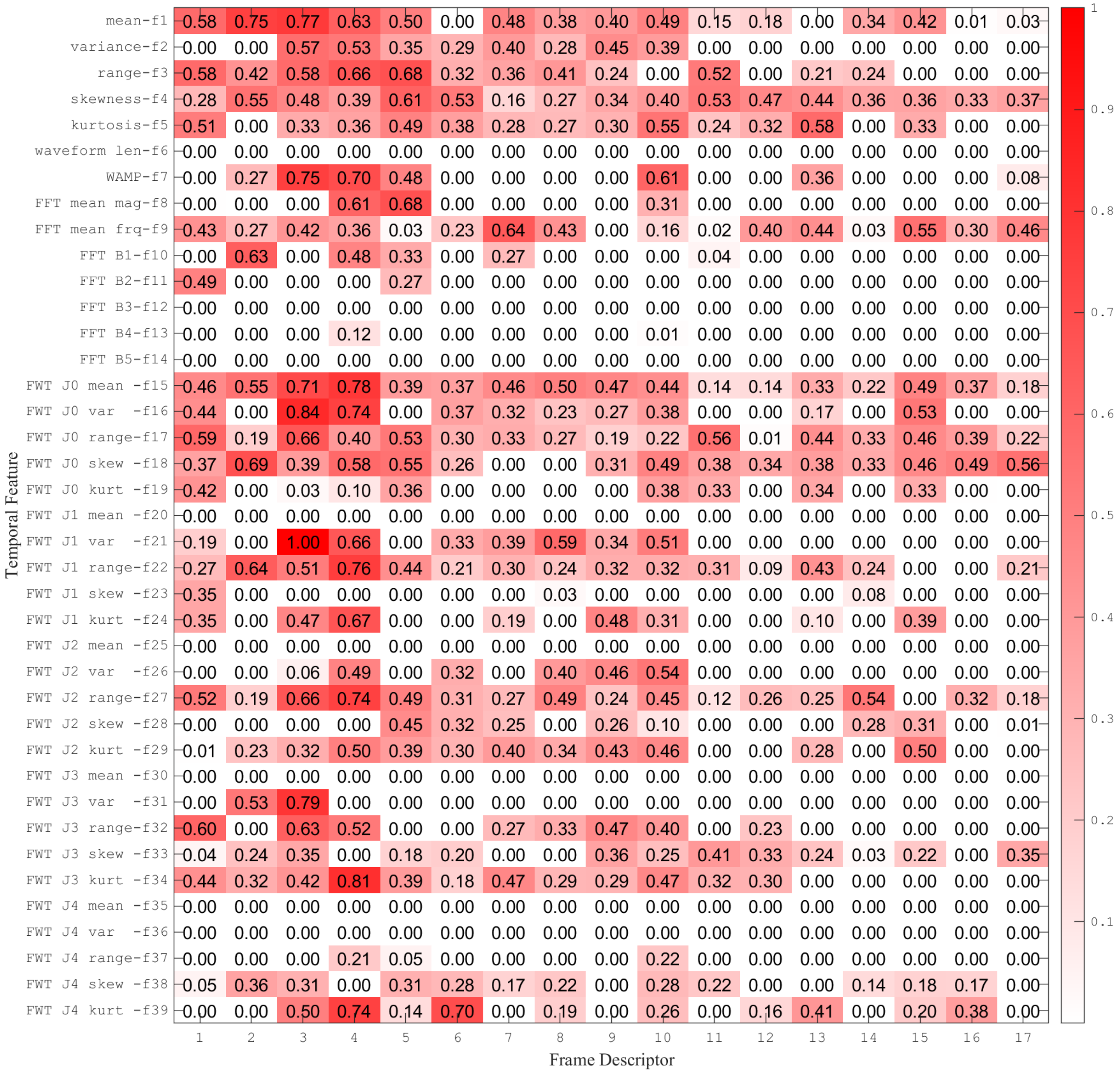


Fig. 114. Feature number against accuracy for the Leg Band dataset.



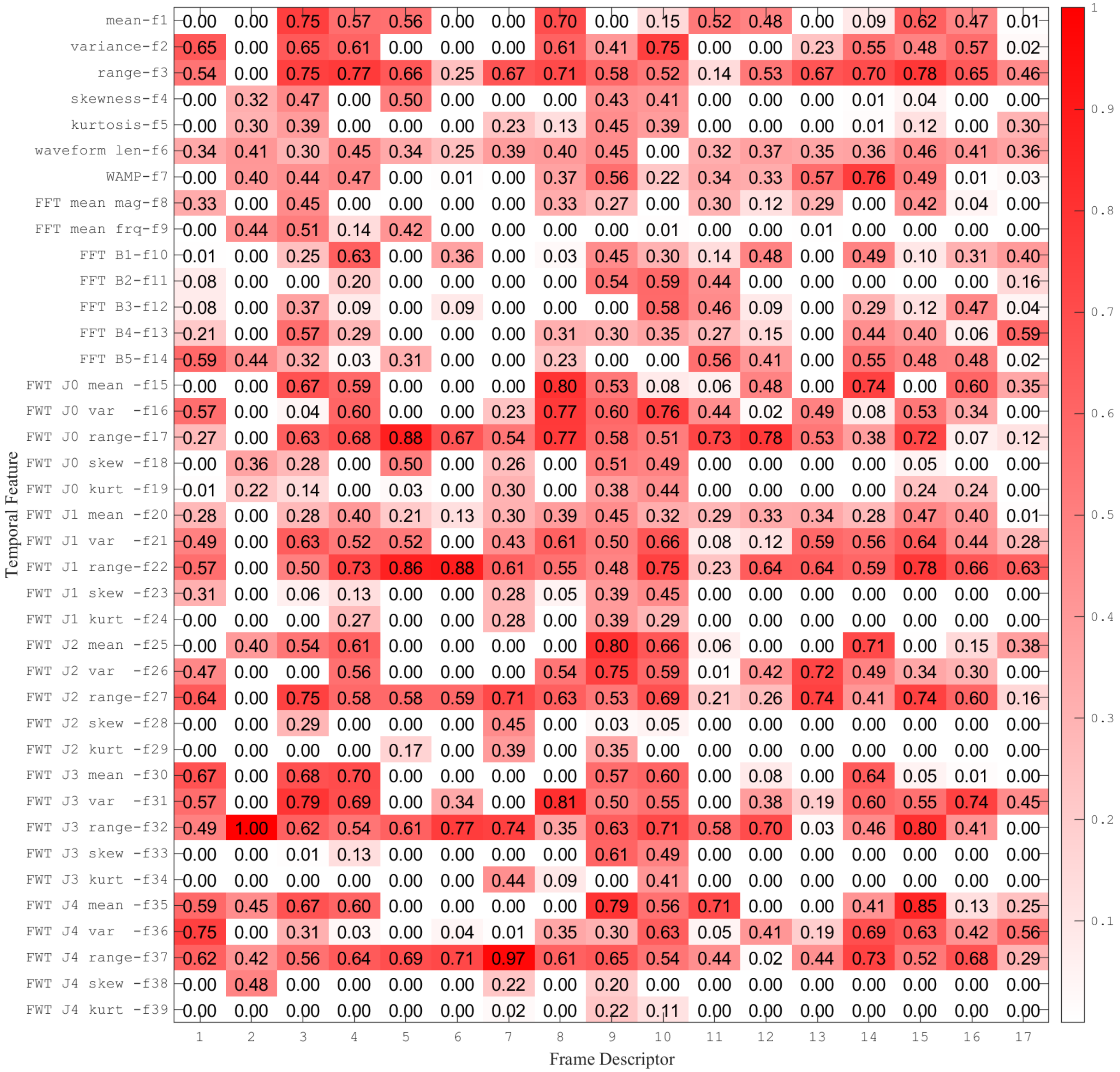
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
mean-f1	0.10	0.74	0.87	0.64	0.79	0.31	0.50	0.18	0.59	0.56	0.12	0.06	0.49	0.45	0.57	0.30	0.16
variance-f2	0.16	0.27	0.20	0.07	0.11	0.25	0.49	0.39	0.73	0.20	0.00	0.00	0.00	0.00	0.00	0.00	0.00
range-f3	0.60	0.92	0.56	0.14	0.73	0.58	0.43	0.37	0.54	0.83	0.26	0.14	0.38	0.52	0.34	0.68	0.38
skewness-f4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
kurtosis-f5	0.00	0.12	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.04	0.00	0.00	0.00	0.00	0.00
waveform len-f6	0.00	0.00	0.00	0.00	0.07	0.04	0.12	0.07	0.00	0.17	0.05	0.12	0.00	0.00	0.06	0.00	0.00
WAMP-f7	0.46	0.38	0.58	0.29	0.50	0.18	0.14	0.56	0.66	0.44	0.00	0.21	0.13	0.20	0.37	0.34	0.24
FFT mean mag-f8	0.00	0.33	0.58	0.28	0.84	0.55	0.31	0.19	0.35	0.21	0.15	0.25	0.12	0.11	0.19	0.19	0.00
FFT mean frq-f9	0.22	0.31	0.31	0.16	0.20	0.00	0.03	0.18	0.25	0.23	0.23	0.22	0.26	0.24	0.22	0.17	0.21
FFT B1-f10	0.64	0.95	0.40	0.30	0.70	0.54	0.28	0.21	0.16	0.25	0.00	0.40	0.27	0.18	0.35	0.58	0.13
FFT B2-f11	0.17	0.37	0.61	0.33	0.46	0.22	0.38	0.53	0.37	0.16	0.00	0.08	0.00	0.14	0.08	0.00	0.00
FFT B3-f12	0.21	0.55	0.46	0.18	0.62	0.30	0.25	0.23	0.39	0.28	0.00	0.16	0.19	0.14	0.16	0.21	0.00
FFT B4-f13	0.00	0.58	0.66	0.25	0.50	0.31	0.37	0.11	0.14	0.20	0.12	0.09	0.22	0.13	0.32	0.00	0.00
FFT B5-f14	0.12	0.39	0.43	0.13	0.06	0.27	0.28	0.37	0.34	0.28	0.00	0.07	0.06	0.26	0.02	0.00	0.34
FWT J0 mean -f15	0.31	0.55	0.69	0.53	0.67	0.56	0.51	0.20	0.46	0.63	0.16	0.28	0.14	0.21	0.53	0.31	0.18
FWT J0 var -f16	0.00	0.00	0.38	0.02	0.37	0.07	0.07	0.30	0.44	0.46	0.00	0.00	0.00	0.00	0.00	0.00	0.00
FWT J0 range-f17	0.92	0.93	0.88	0.30	1.00	0.47	0.44	0.18	0.38	0.56	0.53	0.50	0.73	0.28	0.69	0.49	0.55
FWT J0 skew -f18	0.00	0.12	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
FWT J0 kurt -f19	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
FWT J1 mean -f20	0.16	0.68	0.86	0.40	0.62	0.46	0.38	0.25	0.58	0.45	0.14	0.39	0.20	0.32	0.31	0.32	0.16
FWT J1 var -f21	0.51	0.16	0.65	0.71	0.32	0.23	0.36	0.17	0.58	0.09	0.00	0.01	0.02	0.01	0.32	0.00	0.02
FWT J1 range-f22	0.52	0.66	0.81	0.48	0.90	0.53	0.27	0.47	0.38	0.78	0.37	0.64	0.55	0.76	0.73	0.51	0.48
FWT J1 skew -f23	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
FWT J1 kurt -f24	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
FWT J2 mean -f25	0.36	0.60	0.45	0.33	0.68	0.33	0.55	0.21	0.59	0.50	0.01	0.25	0.38	0.33	0.60	0.23	0.52
FWT J2 var -f26	0.36	0.44	0.50	0.60	0.53	0.27	0.51	0.25	0.78	0.47	0.15	0.00	0.17	0.13	0.36	0.00	0.05
FWT J2 range-f27	0.50	0.69	0.84	0.71	0.78	0.33	0.56	0.20	0.65	0.75	0.12	0.54	0.51	0.85	0.64	0.53	0.41
FWT J2 skew -f28	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
FWT J2 kurt -f29	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
FWT J3 mean -f30	0.36	0.77	0.81	0.34	0.67	0.42	0.42	0.48	0.74	0.61	0.16	0.40	0.04	0.23	0.71	0.40	0.22
FWT J3 var -f31	0.30	0.27	0.79	0.40	0.48	0.23	0.35	0.20	0.45	0.74	0.00	0.00	0.00	0.00	0.14	0.03	0.16
FWT J3 range-f32	0.57	0.48	0.58	0.18	0.58	0.34	0.30	0.41	0.51	0.70	0.06	0.00	0.48	0.00	0.47	0.31	0.03
FWT J3 skew -f33	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
FWT J3 kurt -f34	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
FWT J4 mean -f35	0.59	0.29	0.84	0.07	0.84	0.16	0.38	0.55	0.25	0.33	0.01	0.11	0.00	0.25	0.57	0.40	0.31
FWT J4 var -f36	0.25	0.51	0.92	0.59	0.54	0.16	0.56	0.36	0.63	0.37	0.00	0.00	0.00	0.15	0.00	0.19	0.00
FWT J4 range-f37	0.72	0.94	0.93	0.55	0.74	0.52	0.45	0.43	0.55	0.64	0.56	0.38	0.64	0.63	0.68	0.57	0.65
FWT J4 skew -f38	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
FWT J4 kurt -f39	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Fig. 115. Temporal feature weight - Smart Table Cloth dataset.



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
mean-f1	0.58	0.75	0.77	0.63	0.50	0.00	0.48	0.38	0.40	0.49	0.15	0.18	0.00	0.34	0.42	0.01	0.03
variance-f2	0.00	0.00	0.57	0.53	0.35	0.29	0.40	0.28	0.45	0.39	0.00	0.00	0.00	0.00	0.00	0.00	0.00
range-f3	0.58	0.42	0.58	0.66	0.68	0.32	0.36	0.41	0.24	0.00	0.52	0.00	0.21	0.24	0.00	0.00	0.00
skewness-f4	0.28	0.55	0.48	0.39	0.61	0.53	0.16	0.27	0.34	0.40	0.53	0.47	0.44	0.36	0.36	0.33	0.37
kurtosis-f5	0.51	0.00	0.33	0.36	0.49	0.38	0.28	0.27	0.30	0.55	0.24	0.32	0.58	0.00	0.33	0.00	0.00
waveform len-f6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
WAMP-f7	0.00	0.27	0.75	0.70	0.48	0.00	0.00	0.00	0.00	0.61	0.00	0.00	0.36	0.00	0.00	0.00	0.08
FFT mean mag-f8	0.00	0.00	0.00	0.61	0.68	0.00	0.00	0.00	0.00	0.31	0.00	0.00	0.00	0.00	0.00	0.00	0.00
FFT mean frq-f9	0.43	0.27	0.42	0.36	0.03	0.23	0.64	0.43	0.00	0.16	0.02	0.40	0.44	0.03	0.55	0.30	0.46
FFT B1-f10	0.00	0.63	0.00	0.48	0.33	0.00	0.27	0.00	0.00	0.00	0.04	0.00	0.00	0.00	0.00	0.00	0.00
FFT B2-f11	0.49	0.00	0.00	0.00	0.27	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
FFT B3-f12	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
FFT B4-f13	0.00	0.00	0.00	0.12	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00
FFT B5-f14	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
FWT J0 mean -f15	0.46	0.55	0.71	0.78	0.39	0.37	0.46	0.50	0.47	0.44	0.14	0.14	0.33	0.22	0.49	0.37	0.18
FWT J0 var -f16	0.44	0.00	0.84	0.74	0.00	0.37	0.32	0.23	0.27	0.38	0.00	0.00	0.17	0.00	0.53	0.00	0.00
FWT J0 range-f17	0.59	0.19	0.66	0.40	0.53	0.30	0.33	0.27	0.19	0.22	0.56	0.01	0.44	0.33	0.46	0.39	0.22
FWT J0 skew -f18	0.37	0.69	0.39	0.58	0.55	0.26	0.00	0.00	0.31	0.49	0.38	0.34	0.38	0.33	0.46	0.49	0.56
FWT J0 kurt -f19	0.42	0.00	0.03	0.10	0.36	0.00	0.00	0.00	0.00	0.38	0.33	0.00	0.34	0.00	0.33	0.00	0.00
FWT J1 mean -f20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
FWT J1 var -f21	0.19	0.00	1.00	0.66	0.00	0.33	0.39	0.59	0.34	0.51	0.00	0.00	0.00	0.00	0.00	0.00	0.00
FWT J1 range-f22	0.27	0.64	0.51	0.76	0.44	0.21	0.30	0.24	0.32	0.32	0.31	0.09	0.43	0.24	0.00	0.00	0.21
FWT J1 skew -f23	0.35	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.00	0.00	0.00	0.00	0.00	0.08	0.00	0.00	0.00
FWT J1 kurt -f24	0.35	0.00	0.47	0.67	0.00	0.00	0.19	0.00	0.48	0.31	0.00	0.00	0.10	0.00	0.39	0.00	0.00
FWT J2 mean -f25	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
FWT J2 var -f26	0.00	0.00	0.06	0.49	0.00	0.32	0.00	0.40	0.46	0.54	0.00	0.00	0.00	0.00	0.00	0.00	0.00
FWT J2 range-f27	0.52	0.19	0.66	0.74	0.49	0.31	0.27	0.49	0.24	0.45	0.12	0.26	0.25	0.54	0.00	0.32	0.18
FWT J2 skew -f28	0.00	0.00	0.00	0.00	0.45	0.32	0.25	0.00	0.26	0.10	0.00	0.00	0.00	0.28	0.31	0.00	0.01
FWT J2 kurt -f29	0.01	0.23	0.32	0.50	0.39	0.30	0.40	0.34	0.43	0.46	0.00	0.00	0.28	0.00	0.50	0.00	0.00
FWT J3 mean -f30	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
FWT J3 var -f31	0.00	0.53	0.79	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
FWT J3 range-f32	0.60	0.00	0.63	0.52	0.00	0.00	0.27	0.33	0.47	0.40	0.00	0.23	0.00	0.00	0.00	0.00	0.00
FWT J3 skew -f33	0.04	0.24	0.35	0.00	0.18	0.20	0.00	0.00	0.36	0.25	0.41	0.33	0.24	0.03	0.22	0.00	0.35
FWT J3 kurt -f34	0.44	0.32	0.42	0.81	0.39	0.18	0.47	0.29	0.29	0.47	0.32	0.30	0.00	0.00	0.00	0.00	0.00
FWT J4 mean -f35	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
FWT J4 var -f36	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
FWT J4 range-f37	0.00	0.00	0.00	0.21	0.05	0.00	0.00	0.00	0.00	0.22	0.00	0.00	0.00	0.00	0.00	0.00	0.00
FWT J4 skew -f38	0.05	0.36	0.31	0.00	0.31	0.28	0.17	0.22	0.00	0.28	0.22	0.00	0.00	0.14	0.18	0.17	0.00
FWT J4 kurt -f39	0.00	0.00	0.50	0.74	0.14	0.70	0.00	0.19	0.00	0.26	0.00	0.16	0.41	0.00	0.20	0.38	0.00

Fig. 116. Temporal feature weight - Robot Skin dataset.



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
mean-f1	0.00	0.00	0.75	0.57	0.56	0.00	0.00	0.70	0.00	0.15	0.52	0.48	0.00	0.09	0.62	0.47	0.01
variance-f2	0.65	0.00	0.65	0.61	0.00	0.00	0.00	0.61	0.41	0.75	0.00	0.00	0.23	0.55	0.48	0.57	0.02
range-f3	0.54	0.00	0.75	0.77	0.66	0.25	0.67	0.71	0.58	0.52	0.14	0.53	0.67	0.70	0.78	0.65	0.46
skewness-f4	0.00	0.32	0.47	0.00	0.50	0.00	0.00	0.00	0.43	0.41	0.00	0.00	0.00	0.01	0.04	0.00	0.00
kurtosis-f5	0.00	0.30	0.39	0.00	0.00	0.00	0.23	0.13	0.45	0.39	0.00	0.00	0.00	0.01	0.12	0.00	0.30
waveform len-f6	0.34	0.41	0.30	0.45	0.34	0.25	0.39	0.40	0.45	0.00	0.32	0.37	0.35	0.36	0.46	0.41	0.36
WAMP-f7	0.00	0.40	0.44	0.47	0.00	0.01	0.00	0.37	0.56	0.22	0.34	0.33	0.57	0.76	0.49	0.01	0.03
FFT mean mag-f8	0.33	0.00	0.45	0.00	0.00	0.00	0.00	0.33	0.27	0.00	0.30	0.12	0.29	0.00	0.42	0.04	0.00
FFT mean frq-f9	0.00	0.44	0.51	0.14	0.42	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.01	0.00	0.00	0.00	0.00
FFT B1-f10	0.01	0.00	0.25	0.63	0.00	0.36	0.00	0.03	0.45	0.30	0.14	0.48	0.00	0.49	0.10	0.31	0.40
FFT B2-f11	0.08	0.00	0.00	0.20	0.00	0.00	0.00	0.00	0.54	0.59	0.44	0.00	0.00	0.00	0.00	0.00	0.16
FFT B3-f12	0.08	0.00	0.37	0.09	0.00	0.09	0.00	0.00	0.00	0.58	0.46	0.09	0.00	0.29	0.12	0.47	0.04
FFT B4-f13	0.21	0.00	0.57	0.29	0.00	0.00	0.00	0.31	0.30	0.35	0.27	0.15	0.00	0.44	0.40	0.06	0.59
FFT B5-f14	0.59	0.44	0.32	0.03	0.31	0.00	0.00	0.23	0.00	0.00	0.56	0.41	0.00	0.55	0.48	0.48	0.02
FWT J0 mean -f15	0.00	0.00	0.67	0.59	0.00	0.00	0.00	0.80	0.53	0.08	0.06	0.48	0.00	0.74	0.00	0.60	0.35
FWT J0 var -f16	0.57	0.00	0.04	0.60	0.00	0.00	0.23	0.77	0.60	0.76	0.44	0.02	0.49	0.08	0.53	0.34	0.00
FWT J0 range-f17	0.27	0.00	0.63	0.68	0.88	0.67	0.54	0.77	0.58	0.51	0.73	0.78	0.53	0.38	0.72	0.07	0.12
FWT J0 skew -f18	0.00	0.36	0.28	0.00	0.50	0.00	0.26	0.00	0.51	0.49	0.00	0.00	0.00	0.00	0.05	0.00	0.00
FWT J0 kurt -f19	0.01	0.22	0.14	0.00	0.03	0.00	0.30	0.00	0.38	0.44	0.00	0.00	0.00	0.00	0.24	0.24	0.00
FWT J1 mean -f20	0.28	0.00	0.28	0.40	0.21	0.13	0.30	0.39	0.45	0.32	0.29	0.33	0.34	0.28	0.47	0.40	0.01
FWT J1 var -f21	0.49	0.00	0.63	0.52	0.52	0.00	0.43	0.61	0.50	0.66	0.08	0.12	0.59	0.56	0.64	0.44	0.28
FWT J1 range-f22	0.57	0.00	0.50	0.73	0.86	0.88	0.61	0.55	0.48	0.75	0.23	0.64	0.64	0.59	0.78	0.66	0.63
FWT J1 skew -f23	0.31	0.00	0.06	0.13	0.00	0.00	0.28	0.05	0.39	0.45	0.00	0.00	0.00	0.00	0.00	0.00	0.00
FWT J1 kurt -f24	0.00	0.00	0.00	0.27	0.00	0.00	0.28	0.00	0.39	0.29	0.00	0.00	0.00	0.00	0.00	0.00	0.00
FWT J2 mean -f25	0.00	0.40	0.54	0.61	0.00	0.00	0.00	0.00	0.80	0.66	0.06	0.00	0.00	0.71	0.00	0.15	0.38
FWT J2 var -f26	0.47	0.00	0.00	0.56	0.00	0.00	0.00	0.54	0.75	0.59	0.01	0.42	0.72	0.49	0.34	0.30	0.00
FWT J2 range-f27	0.64	0.00	0.75	0.58	0.58	0.59	0.71	0.63	0.53	0.69	0.21	0.26	0.74	0.41	0.74	0.60	0.16
FWT J2 skew -f28	0.00	0.00	0.29	0.00	0.00	0.00	0.45	0.00	0.03	0.05	0.00	0.00	0.00	0.00	0.00	0.00	0.00
FWT J2 kurt -f29	0.00	0.00	0.00	0.00	0.17	0.00	0.39	0.00	0.35	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
FWT J3 mean -f30	0.67	0.00	0.68	0.70	0.00	0.00	0.00	0.00	0.57	0.60	0.00	0.08	0.00	0.64	0.05	0.01	0.00
FWT J3 var -f31	0.57	0.00	0.79	0.69	0.00	0.34	0.00	0.81	0.50	0.55	0.00	0.38	0.19	0.60	0.55	0.74	0.45
FWT J3 range-f32	0.49	1.00	0.62	0.54	0.61	0.77	0.74	0.35	0.63	0.71	0.58	0.70	0.03	0.46	0.80	0.41	0.00
FWT J3 skew -f33	0.00	0.00	0.01	0.13	0.00	0.00	0.00	0.00	0.61	0.49	0.00	0.00	0.00	0.00	0.00	0.00	0.00
FWT J3 kurt -f34	0.00	0.00	0.00	0.00	0.00	0.00	0.44	0.09	0.00	0.41	0.00	0.00	0.00	0.00	0.00	0.00	0.00
FWT J4 mean -f35	0.59	0.45	0.67	0.60	0.00	0.00	0.00	0.00	0.79	0.56	0.71	0.00	0.00	0.41	0.85	0.13	0.25
FWT J4 var -f36	0.75	0.00	0.31	0.03	0.00	0.04	0.01	0.35	0.30	0.63	0.05	0.41	0.19	0.69	0.63	0.42	0.56
FWT J4 range-f37	0.62	0.42	0.56	0.64	0.69	0.71	0.97	0.61	0.65	0.54	0.44	0.02	0.44	0.73	0.52	0.68	0.29
FWT J4 skew -f38	0.00	0.48	0.00	0.00	0.00	0.00	0.22	0.00	0.20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
FWT J4 kurt -f39	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.22	0.11	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Fig. 117. Temporal feature weight - Leg Band dataset.

8.5 Performance Benchmark

We evaluated the computational performance with a dataset recording file (.mat format) of 286MB (Leg Band dataset person 1 recording 1). The benchmark was carried out on a 2018 MacBook Pro with a six-core 2.6GHz Intel Core i7 processor, and Matlab 2019a. The total frame descriptors calculation took 43.87s and total key frames 2.47s. All the temporal features from all frame descriptors took 350.05s and the spatial features 0.765s. During the temporal feature calculation, the most time consuming process is the fast wavelet transform, which takes 281.35s out of the 350.05s. The 20 fold NCA with all the recordings from the leg band dataset took 926.83s.

However, since the TPM feature set is meant to help explore the useful features for specific data set offline, the computational requirement is less important. With the NCA optimization method, developers can further reduce and select the features to be computed based on their specific requirements.

8.6 Conclusion

A generic algorithm, including data processing and the TPM Feature Set, which is built upon various studies, can be used to extract information from both the space and time domains. Through our evaluation, our approach shows superior accuracy compared to the original studies in which the datasets were published with ad hoc algorithms. The key conclusion is that, for different applications, the TPM sensor data exhibit different natures. Not all features contribute equally, and the feature weights vary with different applications. Neighborhood component analysis can be used to locate those useful features and further optimise a system by reducing feature calculation efforts.

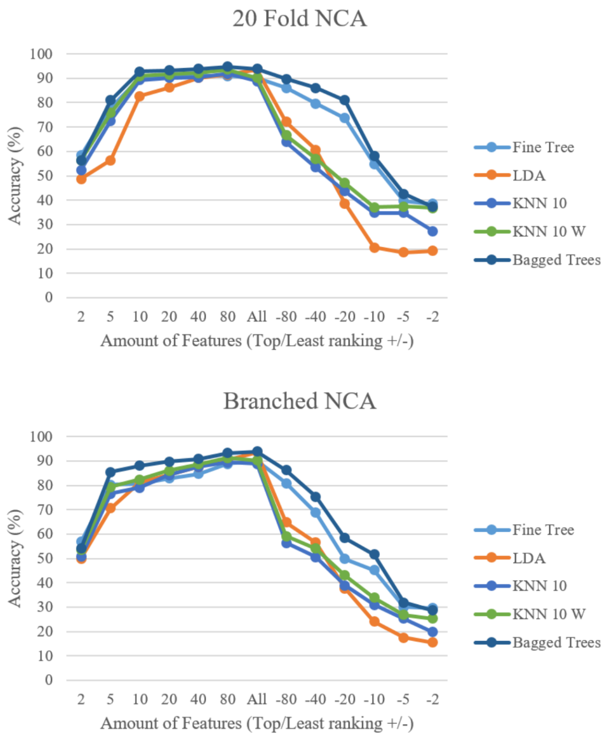


Fig. 118. Feature number against accuracy for the Robot Skin dataset.

Chapter 9 Conclusion

This dissertation has explored using textile pressure mapping (TPM) sensors in pervasive and wearable activity recognition tasks. The work is carried out through an intertwined process between refining a general unified framework and implementing, testing and validating the framework in novel empirical studies. For example, hardware version F-3.1 with the Android application using USB-OTG as the data link, which is a wired yet portable version of the system, has enabled the leg band study. The study has then inspired a fully wireless system that is suitable for sports applications, hardware version M-1 with Bluetooth communication to Android applications. Then this wireless system with a high 550Hz sampling rate made the soccer shoe study possible.

The contributions can be summarized into two folds:

1) A unified TPM framework that includes sensing hardware, software, and algorithms for ambient, object and wearable activity recognition.
2) Various empirical studies that validate TPM' capability in different activity recognition scenarios with cross-validated results.

9.1 The Unified TPM Framwork

The initial goal of this work was set on exploring various application scenarios. This would require a versatile, low-cost, open and easy-to-use method of deploying pressure sensing technology. However, a method that fits all those expectations did not exist prior to this work. Thus this work started on building such a unified framework in order to carry out empirical studies. Every application study has shined a new light on how the framework can be improved. And every improved iteration of the framework would enable and inspire new applications. Through this iterative process, the TPM framework thus ultimately becomes the backbone of this collection of work. It is both an abstract collection of design knowledge, and solid implementations that future developers can make use of. The framework covers four scopes: the design space (Chapter 2), the hardware architecture (Chapter 7), the software toolkit (Chapter 6), and the algorithms (Chapter 8).

Design Space

From observing the fundamental physical interactions, sensing principle of TPM technology and specific activity recognition contexts, the design space points out two core dimensions: propagation and intention. For designing the practical systems, the design space also includes three other dimensions: coverage, mobility, and sampling. The design space is also divided into three subspaces based on the activity contexts: ambient, object and wearable. The design space requirements in its five dimensions guide all further development of a TPM sensing system. The TPM framework is a generic framework that can cover all design space dimension ranges and subspaces.

Hardware Architecture

A general hardware architecture is implemented into two major directions:

- Small and portable hardware that can power 32 × 32 or smaller matrix, which is less than half a square meter with 1.5 cm or 1 cm pitch. The focus is to enable wearable or movable object applications, prioritizing portability and power consumption.
- Large scale hardware that powers 128 × 128, which is over 1 or 2 square meters with a similar above-mentioned pitch. This direction aims at exercise mat, carpet, bed, or even room-sized applications. By its parallel hardware nature, the large scale hardware can be further scaled up until the envelope set by the sampling design space dimension in individual systems.

For both directions of hardware, the versatility is addressed with concepts such as profile reconfiguration and virtual surface, that the physical sensor does not have to be a square matrix. Scaling of multiple systems is also possible by connecting several pieces of hardware to a single computer via either wired or wireless protocols.

Software Toolkit

The software toolkit of the framework is consisted of four Tiers as in Fig. 90. They are called 'Tiers' instead of 'layers' is because every Tier is independent of each other, and they operate in parallel. It is designed this way to handle from basic-level hardware operation, middle-level data processing, machine learning to high-level output and feedback. The complete software stack from Fig. 90 is implemented in multiple programming environments including C++(Qt), Python and HTML/CSS/JavaScript. On the other hand, the entire software can also be implemented in C++(Qt) with OpenCV as a closed homogeneous application such as the demonstrator applications for the Smart-Mat. The software handles all tasks in real-time, making use of multiple CPU cores and threads in either the heterogeneous software system or the closed homogeneous C++(Qt) application.

Algorithms and the TPM Feature Set

The algorithms part of the framework is an abstract collection of mathematical methods of extracting features from the the TPM sensor data that has both the space domain and the time domain. First, one of the domains is reduced, reducing space domain results in temporal sequences of *Frame Descriptors*, reducing time domain results in individual *Key Frames*. Then temporal features are calculated from *Frame Descriptors*, and spacial features are calculated from *Key Frames*. Chapter 8 has given a base collection of 17 × 39 temporal features and 8 × 10 spatial features as examples. With datasets from three applications, the features' importance to the classification result is analyzed with NCA. The results strongly suggest that for different applications, the useful features are very diverse. Some features may be irrelevant for some application with zero feature weight, while has higher significance in other application. Features extracted by pre-trained deep neural networks is also compared with the spatial and temporal feature subsets in Section 3.2.

The TPM framework can help either experts with knowledge of textile pressure mapping sensors or beginner developers to implement activity recognition studies in every aspect: assembling the physical sensing material, implementing or designing new electronics hardware, recording and analyzing data with a comprehensive set of algorithms and a heterogeneous software environment.

9.2 Key Findings in Empirical Studies

Various applications have been evaluated to validate the potential of using TPM to detect activities. The empirical studies are categorized according to the three design subspaces: ambient, object and wearable.

TABLE 20
Application Summary

Application	Participants×Recordings	Classes/Chance Level	Accuracy(internal/external)[1]
Table Cloth	10 × 8	8 / 12.5%	94.6% / 77.7%
Leg Band	6 × 4	7 / 14.3%	81.7% / 74.9%
Seat Cover	16 × 2	10 / 10%	79.5% / -
Robot Skin	24 × 2 + 5 [2]	7 / 14.3%	88.8% / 89.1%
Soccer Shoe	1 robot leg	17 / 5.9%	98.8% / -
Smart Mat	7 × 2	10 / 10%	88.7% / 86.4%
Gait Analysis	13 × 12	13 / 7.7%	87.6% / -

[1] internal refers to leave-recording-out cross-validation within all participants' data pool; external refers to leave-one-person-out unless otherwise stated.
[2] the experiment consists of two mutually blinded groups.

The range of empirical studies in this dissertation covers very different sensing aspects and shall give insight for the future development of how planar surface pressure signatures can be used to detect activities. For example, the *Table Cloth* demonstrates the idea that by surface pressure we can detect indirect activities that are propagated through objects, in this case, the force from the plates to the table cloth is related to different actions with the cutlery. The *Leg Band* and other wearable garment shows that while we move our body, our body and the fabric we wear have an interactive force which is distinct to different actions. Different activities cast distinct 'body-print' onto the supporting surface, such as the application of the *Seat Cover* or the *Smart Mat*. Even with seemingly similar footprints, when the morphing footprints can be examined in a fine time granularity, they can be used to identify people as shown in the *Carpet* application. With the help of the virtual surface concept, TPM sensors can also be tailored to fit on irregular contour shapes such as the upper surface of the *Soccer Shoe* to evaluate ball impact.

9.3 Limitations

Using surface planner pressure to detect activities has its own limitations. Some of them require multi-disciplinary collaborations to solve including material, textile, and manufacturing engineering.

From the activity recognition field, the major limitation is that it is difficult to detect activities that are further away from the sensing surface, i.e., when indirect propagation and direct propagation both exist in a scenario. For example, in the *Smart Mat* application, exercise calf raise (Class 7) and squat (Class 8) are often miss-classified. Which may be caused by that both activities have a foot shape with a repeating change of the average pressure.

As pointed out in Chapter 2 Section 2.2, flexible pressure sensors are not good at repetitively measuring the absolute force, which is also tested by other research endeavors such as [53]. To measure precise force, rigid form sensors such as force gauges should be used. However, the thinness and flexibility allow flexible pressure sensors to be implemented as matrix forms to measure the distribution and dynamic change of the pressure.

As mentioned in Chapter 2 Section 2.3 Since stretch sensors are also made of carbonated materials and based on the similar resistance change, how to isolate or whether it is needed to isolate pressure from elongation is an aspect that should not be ignored.

A major practical problem is wiring the connections from the fabric to the electronics. In this work, most of the connections are implemented with parallel ribbon cables, which are copper wires with PVC isolation. The reasoning behind is both the cost and ease of implementation. Obviously, a more fabric based connection solution is needed for the transition from research prototypes to consumer usage. The connection solution should include three parts: how to easily connect the wires to the fabric, how to embed the wires into the rest of the garment, and how to easily plug the wires to the electronics. All of the three parts should consider a large number of connections (64 connections for a 32 × 32 matrix).

9.4 Outlook

From the various applications, it can be concluded that dynamic planar surface force distribution can be used as a sensing modality. On a higher level argument, the force is normally the actual direct cause or consequence of activities or interactions. To compare, the measurements from popular motion sensors such as inertial measurement units or proximity sensors, and various vision methods such as plain cameras or the recently more popular depth cameras, are observations of activities or interactions. Thus planar surface force can be on the same level, if not more fundamental, as the motion in determining the type of activities. Hence it is possible to see TPM sensing as part of the future of pervasive and wearable sensing.

The TPM framework is proposed as an enabler to push forward planar pressure sensing technology in the field of ubiquitous and wearable computing. The framework is aimed to free other interested developers from the necessary and non-trivial workloads needed to develop and evaluate TPM systems. Thus, the design space and empirical studies from this work shall be seen as inspirations, not limiting boundaries.

Section 3.2 and 3.3.3 gave a peek into using deep neural networks to process the pressure imagery from the TPM sensors. Using transfer learning, a neural network previously trained with a large image database can be used to perform classification on TPM sensor imagery, to bypass the large dataset requirement. For the specific dataset, recurrent neural network (RNN) offers significantly superior results. This may because that RNN models make predictions based on a sequence of images, which takes consideration of the time order - this is the information that conventional features are missing. In the collection of conventional features in Chapter 8, the kurtosis and the skewness of the frame descriptors and their wavelet coefficients describe the time order only on a rough level. An outline of the future work in this aspect is already written in the respective section. What is worth emphasizing is the vision of building a unified neural network model that can determine the application from the data, and proceed on with application-specific classification models. A generic 'TPM AI' consisting a hierarchical structure of these models can be an interesting research direction.

In the future, a software environment fully in a mobile operating system is especially relevant for portable applications. For Android, applications are already developed with Bluetooth data streaming, recording and visualizing, and has been used in some studies such as the *Soccer Shoe*. In those data recording applications, Tier 1 and Tier 4 are essentially implemented. What are missing are the data processing and machine learning parts from Tier 2 and Tier 3. Previously these processes were considered resource consuming and lacking library support. Recently with the release of Android Neural Networks API (NNAPI) and TensorFlow Lite, and hardware accelerations in the mobile CPU, mobile applications that encloses all four tiers may be possible and may benefit a wide range of user studies with portable TPM systems.

Publication List

Conference Proceedings

‘From smart clothing to smart table cloth: Design and implementation of a large scale, textile pressure matrix sensor.’
In *International Conference on Architecture of Computing Systems*, pp. 159-170. Springer, Cham, 2014.
Zhou, Bo, Jingyuan Cheng, Mathias Sundholm, and Paul Lukowicz.

‘Smart table surface: A novel approach to pervasive dining monitoring.’
In *Pervasive Computing and Communications (PerCom), 2015 IEEE International Conference on*, pp. 155-162. IEEE, 2015.
Zhou, Bo, Jingyuan Cheng, Mathias Sundholm, Attila Reiss, Wuhuang Huang, Oliver Amft, and Paul Lukowicz.

‘Smart soccer shoe: monitoring foot-ball interaction with shoe integrated textile pressure sensor matrix.’
In *Proceedings of the 2016 ACM International Symposium on Wearable Computers*, pp. 64-71. ACM, 2016.
Honorable Mention
Zhou, Bo, Harald Koerger, Markus Wirth, Constantin Zwick, Christine Martindale, Heber Cruz, Bjoern Eskofier, and Paul Lukowicz.

‘Never skip leg day: A novel wearable approach to monitoring gym leg exercises.’
In *Pervasive Computing and Communications (PerCom), 2016 IEEE International Conference on*, pp. 1-9. IEEE, 2016.
Zhou, Bo, Mathias Sundholm, Jingyuan Cheng, Heber Cruz, and Paul Lukowicz.

‘Smart Blanket: A Real-Time User Posture Sensing Approach for Ergonomic Designs.’
In *International Conference on Applied Human Factors and Ergonomics*, pp. 193-204. Springer, Cham, 2017.
Zhou, Bo, Monit Shah Singh, Muhammet Yildirim, Ivi Prifti, Heber Cruz Zurian, Yorman Munoz Yuncosa, and Paul Lukowicz.

‘Transforming sensor data to the image domain for deep learning — An application to footstep detection.’
In *Neural Networks (IJCNN), 2017 International Joint Conference on*, pp. 2665-2672. IEEE, 2017.
Singh, Monit Shah, Vinaychandran Pondenkandath, **Bo Zhou**, Paul Lukowicz, and Marcus Liwickit.

‘Smart chair: What can simple pressure sensors under the chairs legs tell us about user activity.’
In *UBICOMM13: The Seventh International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies*, pp. 81-84. 2013.
Cheng, Jingyuan, **Bo Zhou**, Mathias Sundholm, and Paul Lukowicz.

‘TPM Framework: a Comprehensive Kit for Exploring Applications with Textile Pressure Mapping Matrix.’
In *UBICOMM17: The Eleventh International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies* 2017
Best Paper
Zhou, Bo, Jingyuan Cheng, Ankur Mawandia, Yujiang He, Zhixin Huang, Mathias Sundholm, Muhammet Yildrim, Heber Cruz, and Paul Lukowicz.

‘TPM Feature Set: a Universal Algorithm for Spatial-Temporal Pressure Mapping Imagery Data.’
In *UBICOMM19: The Thirteenth International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies* 2019
Zhou, Bo, and Paul Lukowicz.

‘Recognizing subtle user activities and person identity with cheap resistive pressure sensing carpet.’
In *Intelligent Environments (IE), 2014 International Conference on*, pp. 148-153. IEEE, 2014.
Cheng, Jingyuan, Mathias Sundholm, **Bo Zhou**, Matthias Kreil, and Paul Lukowicz.

‘Application exploring of ubiquitous pressure sensitive matrix as input resource for home-service robots.’
In *Robot Intelligence Technology and Applications 3*, pp. 359-371. Springer, Cham, 2015.
Cheng, Jingyuan, Mathias Sundholm, Marco Hirsch, **Bo Zhou**, Sebastian Palacio, and Paul Lukowicz.

‘SimpleSkin: towards multipurpose smart garments.’
In *Adjunct Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2015 ACM International Symposium on Wearable Computers*, pp. 241-244. ACM, 2015.
Schneegass, Stefan, Mariam Hassib, **Bo Zhou**, Jingyuan Cheng, Fernando Seoane, Oliver Amft, Paul Lukowicz, and Albrecht Schmidt.

‘Smart-chairs: ubiquitous presentation evaluation based on audience’s activity recognition.’

In *Proceedings of the 11th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, pp. 350-351. ICST, 2014.
Wang, Bing, Jingyuan Cheng, **Bo Zhou,** Orkhan Amiraslanov, Paul Lukowicz, and Mengfan Zhang.

'Smart-mat: Recognizing and counting gym exercises with low-cost resistive pressure sensing matrix.'
In *Proceedings of the 2014 ACM international joint conference on pervasive and ubiquitous computing*, pp. 373-382. ACM, 2014.
Sundholm, Mathias, Jingyuan Cheng, **Bo Zhou**, Akash Sethi, and Paul Lukowicz.

Journals

'Monitoring dietary behavior with a smart dining tray.'
IEEE Pervasive Computing 14, no. 4 (2015): 46-56.
Zhou, Bo, Jingyuan Cheng, Paul Lukowicz, Attila Reiss, and Oliver Amft.

'Measuring muscle activities during gym exercises with textile pressure mapping sensors.'
Pervasive and Mobile Computing 38 (2017): 331-345.
Zhou, Bo, Mathias Sundholm, Jingyuan Cheng, Heber Cruz, and Paul Lukowicz.

'Textile Pressure Mapping Sensor for Emotional Touch Detection in Human-Robot Interaction.'
Sensors 17, no. 11 (2017): 2585.
Zhou, Bo, Carlos Andres Velez Altamirano, Heber Cruz Zurian, Seyed Reza Atefi, Erik Billing,
Fernando Seoane Martinez, and Paul Lukowicz.

Others

'Trainwear: A real-time assisted training feedback system with fabric wearable sensors.'
In *Pervasive Computing and Communications (PerCom), 2017 IEEE International Conference on*,
pp. 85-87. IEEE, 2017.
Best Demo Award
Zhou, Bo, Gernot Bahle, Lorenzo Fürg, Monit Shah Singh, Heber Zurian Cruz, and Paul Lukowicz

'The carpet knows: Identifying people in a smart environment from a single step.'
In *Pervasive Computing and Communications Workshops (PerCom Workshops), 2017 IEEE International Conference on*,
pp. 527-532. IEEE, 2017.
Zhou, Bo, Monit Shah Singh, Sugandha Doda, Muhammet Yildirim, Jingyuan Cheng, and Paul Lukowicz.

'Textile Pressure Force Mapping.'
In *Smart Textiles*, pp. 31-47. Springer, Cham, 2017.
Book Chapter
Zhou, Bo and Paul Lukowicz.

'Textile building blocks: Toward simple, modularized, and standardized smart textile.'
In *Smart Textiles*, pp. 303-331. Springer, Cham, 2017.
Book Chapter
Cheng, Jingyuan, **Bo Zhou**, Paul Lukowicz, Fernando Seoane, Matija Varga, Andreas Mehmann, Peter Chabrecek et al.

References

[1] E. Kim, S. Helal, and D. Cook, “Human activity recognition and pattern discovery,” *IEEE Pervasive Computing*, vol. 9, no. 1, 2010.

[2] A. Bulling, U. Blanke, and B. Schiele, “A tutorial on human activity recognition using body-worn inertial sensors,” *ACM Computing Surveys (CSUR)*, vol. 46, no. 3, p. 33, 2014.

[3] A. Avci, S. Bosch, M. Marin-Perianu, R. Marin-Perianu, and P. Havinga, “Activity recognition using inertial sensing for healthcare, wellbeing and sports applications: A survey,” in *Architecture of computing systems (ARCS), 2010 23rd international conference on*. VDE, 2010, pp. 1–10.

[4] I. A. Essa, “Ubiquitous sensing for smart and aware environments,” *IEEE personal communications*, vol. 7, no. 5, pp. 47–49, 2000.

[5] C. Ramos, J. C. Augusto, and D. Shapiro, “Ambient intelligence—the next step for artificial intelligence,” *IEEE Intelligent Systems*, vol. 23, no. 2, pp. 15–18, 2008.

[6] X. Yu, “Approaches and principles of fall detection for elderly and patient,” in *e-health Networking, Applications and Services, 2008. HealthCom 2008. 10th International Conference on*. IEEE, 2008, pp. 42–47.

[7] L. Chen, J. Hoey, C. D. Nugent, D. J. Cook, and Z. Yu, “Sensor-based activity recognition,” *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 6, pp. 790–808, 2012.

[8] R. Bodor, B. Jackson, and N. Papanikolopoulos, “Vision-based human tracking and activity recognition,” in *Proc. of the 11th Mediterranean Conf. on Control and Automation*, vol. 1. Citeseer, 2003.

[9] J. K. Aggarwal and L. Xia, “Human activity recognition from 3d data: A review,” *Pattern Recognition Letters*, vol. 48, pp. 70–80, 2014.

[10] R. Bahroun, O. Michel, F. Frassati, M. Carmona, and J. Lacoume, “New algorithm for footstep localization using seismic sensors in an indoor environment,” *Journal of Sound and Vibration*, vol. 333, no. 3, pp. 1046–1066, 2014.

[11] S. Patel, M. Reynolds, and G. Abowd, “Detecting human movement by differential air pressure sensing in hvac system ductwork: An exploration in infrastructure mediated sensing,” *Pervasive Computing*, pp. 1–18, 2008.

[12] O. D. Lara and M. A. Labrador, “A survey on human activity recognition using wearable sensors.” *IEEE Communications Surveys and Tutorials*, vol. 15, no. 3, pp. 1192–1209, 2013.

[13] J. Pansiot, D. Stoyanov, D. McIlwraith, B. P. Lo, and G.-Z. Yang, “Ambient and wearable sensor fusion for activity recognition in healthcare monitoring systems,” in *4th international workshop on wearable and implantable body sensor networks (BSN 2007)*. Springer, 2007, pp. 208–212.

[14] X. Hong, C. Nugent, M. Mulvenna, S. McClean, B. Scotney, and S. Devlin, “Evidential fusion of sensor data for activity recognition in smart homes,” *Pervasive and Mobile Computing*, vol. 5, no. 3, pp. 236–252, 2009.

[15] M. Zhang and A. A. Sawchuk, “A feature selection-based framework for human activity recognition using wearable multimodal sensors,” in *Proceedings of the 6th International Conference on Body Area Networks*. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2011, pp. 92–98.

[16] D. Bannach, O. Amft, and P. Lukowicz, “Rapid prototyping of activity recognition applications,” *IEEE Pervasive Computing*, vol. 7, no. 2, 2008.

[17] U. Maurer, A. Smailagic, D. P. Siewiorek, and M. Deisher, “Activity recognition and monitoring using multiple sensors on different body positions,” in *Wearable and Implantable Body Sensor Networks, 2006. BSN 2006. International Workshop on*. IEEE, 2006, pp. 4–pp.

[18] C.-C. Yang and Y.-L. Hsu, “A review of accelerometry-based wearable motion detectors for physical activity monitoring,” *Sensors*, vol. 10, no. 8, pp. 7772–7788, 2010.

[19] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, “Activity recognition using cell phone accelerometers,” *ACM SigKDD Explorations Newsletter*, vol. 12, no. 2, pp. 74–82, 2011.

[20] G. Bieber, M. Haescher, and M. Vahl, “Sensor requirements for activity recognition on smart watches,” in *Proceedings of the 6th International Conference on PErvasive Technologies Related to Assistive Environments*. ACM, 2013, p. 67.

[21] J. K. Aggarwal and Q. Cai, “Human motion analysis: A review,” in *Nonrigid and Articulated Motion Workshop, 1997. Proceedings., IEEE*. IEEE, 1997, pp. 90–102.

[22] M. Casdagli, S. Eubank, J. D. Farmer, and J. Gibson, “State space reconstruction in the presence of noise,” *Physica D: Nonlinear Phenomena*, vol. 51, no. 1-3, pp. 52–98, 1991.

[23] D. Kugiumtzis, “State space reconstruction parameters in the analysis of chaotic time series—the role of the time window length,” *Physica D: Nonlinear Phenomena*, vol. 95, no. 1, pp. 13–28, 1996.

[24] D. J. Patterson, D. Fox, H. Kautz, and M. Philipose, “Fine-grained activity recognition by aggregating abstract object usage,” in *Wearable Computers, 2005. Proceedings. Ninth IEEE International Symposium on*. IEEE, 2005, pp. 44–51.

[25] J. A. Ward, P. Lukowicz, G. Troster, and T. E. Starner, “Activity recognition of assembly tasks using body-worn microphones and accelerometers,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 28, no. 10, pp. 1553–1567, 2006.

[26] K. Yatani and K. N. Truong, “Bodyscope: a wearable acoustic sensor for activity recognition,” in *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*. ACM, 2012, pp. 341–350.

[27] K. Kunze and P. Lukowicz, “Sensor placement variations in wearable activity recognition,” *IEEE Pervasive Computing*, vol. 13, no. 4, pp. 32–41, 2014.

[28] A. Dementyev and J. A. Paradiso, “Wristflex: low-power gesture input with wrist-worn pressure sensors,” in *Proceedings of the 27th annual ACM symposium on User interface software and technology*. ACM, 2014, pp. 161–166.

[29] J. Cheng, A. Okoso, K. Kunze, N. Henze, A. Schmidt, P. Lukowicz, and K. Kise, “On the tip of my tongue: a non-invasive pressure-based tongue interface,” in *Proceedings of the 5th Augmented Human International Conference*. ACM, 2014, p. 12.

[30] S. Mota and R. W. Picard, “Automated posture analysis for detecting learner’s interest level,” in *2003 Conference on Computer Vision and Pattern Recognition Workshop*, vol. 5. IEEE, 2003, pp. 49–49.

[31] C. Hsia, K. Liou, A. Aung, V. Foo, W. Huang, and J. Biswas, “Analysis and comparison of sleeping posture classification methods using pressure sensitive bed system,” in *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, 2009, pp. 6131–6134.

[32] “Tekscan - pressure mapping, force measurement & tactile sensors.” [Online]. Available: https://www.tekscan.com

[33] L. Ada, “Force sensitive resistor (fsr)- thin sensor that detects physical pressure.” [Online]. Available: https://learn.adafruit.com/force-sensitive-resistor-fsr?view=all

[34] M. Kreil, G. Ogris, and P. Lukowicz, “Muscle activity evaluation using force sensitive resistors,” in *Medical Devices and Biosensors, 2008. ISSS-MDBS 2008. 5th International Summer School and Symposium on*. IEEE, 2008, pp. 107–110.

[35] G. Ogris, M. Kreil, and P. Lukowicz, “Using fsr based muscule activity monitoring to recognize manipulative arm gestures,” in *Wearable Computers, 2007 11th IEEE International Symposium on*. IEEE, 2007, pp. 45–48.

[36] B. Zhou, M. S. Singh, M. Yildirim, I. Prifti, H. C. Zurian, Y. M. Yuncosa, and P. Lukowicz, “Smart blanket: A real-time user posture sensing approach for ergonomic designs,” in *International Conference on Applied Human Factors and Ergonomics*. Springer, 2017, pp. 193–204.

[37] M. Sundholm, J. Cheng, B. Zhou, A. Sethi, and P. Lukowicz, “Smart-mat: Recognizing and counting gym exercises with low-cost resistive pressure sensing matrix,” in *Proceedings of the 2014 ACM international joint conference on pervasive and ubiquitous computing*. ACM, 2014, pp. 373–382.

[38] S. Schneegass and A. Voit, "Gesturesleeve: using touch sensitive fabrics for gestural input on the forearm for controlling smartwatches," in *Proceedings of the 2016 ACM International Symposium on Wearable Computers*. ACM, 2016, pp. 108–115.

[39] "Tekscan - i-scan system guide." [Online]. Available: https://www.tekscan.com/products-solutions/systems/i-scan-system?tab=software

[40] J. Cheng, O. Amft, and P. Lukowicz, "Active capacitive sensing: Exploring a new wearable sensing modality for activity recognition," in *International conference on pervasive computing*. Springer, 2010, pp. 319–336.

[41] M. Haescher, D. J. Matthies, G. Bieber, and B. Urban, "Capwalk: A capacitive recognition of walking-based activities as a wearable assistive technology," in *Proceedings of the 8th ACM International Conference on PErvasive Technologies Related to Assistive Environments*. ACM, 2015, p. 35.

[42] J. Meyer, P. Lukowicz, and G. Troster, "Textile pressure sensor for muscle activity and motion detection," in *Wearable Computers, 2006 10th IEEE International Symposium on*. IEEE, 2006, pp. 69–72.

[43] J. Meyer, B. Arnrich, J. Schumm, and G. Troster, "Design and modeling of a textile pressure sensor for sitting posture classification," *IEEE Sensors Journal*, vol. 10, no. 8, pp. 1391–1398, 2010.

[44] B. Zhou, *Resistive Pressure Force Sensor Matrix for Wearable and Ubiquitous Computing*. Master's Thesis, 2013.

[45] J. Aneli, G. Zaikov, and L. Khananashvili, "Effects of mechanical deformations on the structurization and electric conductivity of electric conducting polymer composites," *Journal of Applied Polymer Science*, vol. 74, no. 3, pp. 601–621, 1999.

[46] R. Gangopadhyay and A. De, "Conducting polymer nanocomposites: a brief overview," *Chemistry of materials*, vol. 12, no. 3, pp. 608–622, 2000.

[47] M. King, A. Baragwanath, M. Rosamond, D. Wood, and A. Gallant, "Porous pdms force sensitive resistors," *Procedia Chemistry*, vol. 1, no. 1, pp. 568–571, 2009.

[48] H. Mei, C. Zhang, R. Wang, J. Feng, and T. Zhang, "Impedance characteristics of surface pressure-sensitive carbon black/silicone rubber composites," *Sensors and Actuators A: Physical*, vol. 233, pp. 118 – 124, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0924424715300315

[49] M. Knite, V. Teteris, A. Kiploka, and J. Kaupuzs, "Polyisoprene-carbon black nanocomposites as tensile strain and pressure sensor materials," *Sensors and Actuators A: Physical*, vol. 110, no. 1, pp. 142–149, 2004.

[50] S. Schneegass and O. Amft, *Smart Textiles: Fundamentals, Design, and Interaction*. Springer, 2017.

[51] H. Takao, K. Sawada, and M. Ishida, "Silicon smart tactile image sensor with pneumatically swollen single diaphragm structure," in *17th IEEE International Conference on Micro Electro Mechanical Systems. Maastricht MEMS 2004 Technical Digest*. IEEE, 2004, pp. 846–849.

[52] S. AG, *Antistatic Fabrics SEFAR® CARBOTEX*. [Online]. Available: http://techlist.sefar.com/cms/newtechlistpdf.nsf/vwWebPDFs/carbotex_EN.pdf/$FILE/carbotex_EN.pdf

[53] D. Giovanelli and E. Farella, "Force sensing resistor and evaluation of technology for wearable body pressure sensing," *Journal of Sensors*, vol. 2016, 2016.

[54] P. Parzer, A. Sharma, A. Vogl, J. Steimle, A. Olwal, and M. Haller, "Smartsleeve: Real-time sensing of surface and deformation gestures on flexible, interactive textiles, using a hybrid gesture detection pipeline," in *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*. ACM, 2017, pp. 565–577.

[55] A. Libin and J. Cohen-Mansfield, "Therapeutic robocat for nursing home residents with dementia: preliminary inquiry," *American Journal of Alzheimer's Disease & Other Dementias®*, vol. 19, no. 2, pp. 111–116, 2004.

[56] B. Zhou, M. Sundholm, J. Cheng, H. Cruz, and P. Lukowicz, "Measuring muscle activities during gym exercises with textile pressure mapping sensors," *Pervasive and Mobile Computing*, vol. 38, pp. 331–345, 2017.

[57] J. Cheng, B. Zhou, M. Sundholm, and P. Lukowicz, "Smart chair: What can simple pressure sensors under the chairs legs tell us about user activity," in *UBICOMM13: The Seventh International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies*, 2013, pp. 81–84.

[58] M. S. Singh, V. Pondenkandath, B. Zhou, P. Lukowicz, and M. Liwickit, "Transforming sensor data to the image domain for deep learning—an application to footstep detection," in *Neural Networks (IJCNN), 2017 International Joint Conference on*. IEEE, 2017, pp. 2665–2672.

[59] S. Jalaliniya, J. Smith, M. Sousa, L. Büthe, and T. Pederson, "Touch-less interaction with medical images using hand & foot gestures," in *Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication*. ACM, 2013, pp. 1265–1274.

[60] B. Zhou, M. S. Singh, S. Doda, M. Yildirim, J. Cheng, and P. Lukowicz, "The carpet knows: Identifying people in a smart environment from a single step," in *Pervasive Computing and Communications Workshops (PerCom Workshops), 2017 IEEE International Conference on*. IEEE, 2017, pp. 527–532.

[61] B. Zhou, M. Sundholm, J. Cheng, H. Cruz, and P. Lukowicz, "Never skip leg day: A novel wearable approach to monitoring gym leg exercises," in *Pervasive Computing and Communications (PerCom), 2016 IEEE International Conference on*. IEEE, 2016, pp. 1–9.

[62] B. Zhou, C. A. V. Altamirano, H. C. Zurian, S. R. Atefi, E. Billing, F. S. Martinez, and P. Lukowicz, "Textile pressure mapping sensor for emotional touch detection in human-robot interaction," *Sensors*, vol. 17, no. 11, p. 2585, 2017.

[63] B. Zhou, J. Cheng, M. Sundholm, A. Reiss, W. Huang, O. Amft, and P. Lukowicz, "Smart table surface: A novel approach to pervasive dining monitoring," in *Pervasive Computing and Communications (PerCom), 2015 IEEE International Conference on*. IEEE, 2015, pp. 155–162.

[64] B. Zhou, J. Cheng, P. Lukowicz, A. Reiss, and O. Amft, "Monitoring dietary behavior with a smart dining tray," *IEEE Pervasive Computing*, vol. 14, no. 4, pp. 46–56, 2015.

[65] M. Viskaal-van Dongen, F. J. Kok, and C. de Graaf, "Eating rate of commonly consumed foods promotes food and energy intake," *Appetite*, vol. 56, no. 1, pp. 25–31, 2011.

[66] S. Tanihara, T. Imatoh, M. Miyazaki, A. Babazono, Y. Momose, M. Baba, Y. Uryu, and H. Une, "Retrospective longitudinal study on the relationship between 8-year weight change and current eating speed," *Appetite*, vol. 57, no. 1, pp. 179–183, 2011.

[67] A. M. Andrade, G. W. Greene, and K. J. Melanson, "Eating slowly led to decreases in energy intake within meals in healthy women," *Journal of the American Dietetic Association*, vol. 108, no. 7, pp. 1186–1191, 2008.

[68] S. J. Zyzanski and C. D. Jenkins, "Basic dimensions within the coronary-prone behavior pattern," *Journal of Chronic Diseases*, vol. 22, no. 12, pp. 781–795, 1970.

[69] B.-T. Ji, W.-H. Chow, G. Yang, J. K. Mclaughlin, W. Zheng, X.-O. Shu, F. Jin, R.-N. Gao, Y.-T. Gao, and J. F. Fraumeni Jr, "Dietary habits and stomach cancer in shanghai, china," *International journal of cancer*, vol. 76, no. 5, pp. 659–664, 1998.

[70] O. Amft, "Ambient, on-body, and implantable monitoring technologies to assess dietary behavior," in *Handbook of Behavior, Food and Nutrition*. Springer, 2011, pp. 3507–3526.

[71] O. Amft, H. Junker, and G. Troster, "Detection of eating and drinking arm gestures using inertial body-worn sensors," in *Wearable computers, 2005. proceedings. ninth ieee international symposium on*. IEEE, 2005, pp. 160–163.

[72] O. Amft, M. Stäger, P. Lukowicz, and G. Tröster, "Analysis of chewing sounds for dietary monitoring," in *International Conference on Ubiquitous Computing*. Springer, 2005, pp. 56–72.

[73] E. Sazonov, S. Schuckers, P. Lopez-Meyer, O. Makeyev, N. Sazonova, E. L. Melanson, and M. Neuman, "Non-invasive monitoring of chewing and swallowing for objective quantification of ingestive behavior," *Physiological measurement*, vol. 29, no. 5, p. 525, 2008.

[74] J. Cheng, B. Zhou, K. Kunze, C. C. Rheinländer, S. Wille, N. Wehn, J. Weppner, and P. Lukowicz, "Activity recognition and nutrition monitoring in every day situations with a textile capacitive neckband," in *Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication*. ACM, 2013, pp. 155–158.

[75] T. Ono, K. Hori, Y. Masuda, and T. Hayashi, "Recent advances in sensing oropharyngeal swallowing function in japan," *Sensors*, vol. 10, no. 1, pp. 176–202, 2009.

[76] A. Kadomura, C.-Y. Li, Y.-C. Chen, K. Tsukada, I. Siio, and H.-h. Chu, "Sensing fork: eating behavior detection utensil and mobile persuasive game," in *CHI'13 Extended Abstracts on Human Factors in Computing Systems*. ACM, 2013, pp. 1551–1556.

[77] Y. Yue, W. Jia, J. D. Fernstrom, R. J. Sclabassi, M. H. Fernstrom, N. Yao, and M. Sun, "Food volume estimation using a circular reference in image-based dietary studies," in *Bioengineering Conference, Proceedings of the 2010 IEEE 36th Annual Northeast*. IEEE, 2010, pp. 1–2.

[78] H. Gu and D. Wang, "A content-aware fridge based on rfid in smart home for home-healthcare," in *Advanced Communication Technology, 2009. ICACT 2009. 11th International Conference on*, vol. 2. IEEE, 2009, pp. 987–990.

[79] K.-h. Chang, S.-y. Liu, H.-h. Chu, J. Y.-j. Hsu, C. Chen, T.-y. Lin, C.-y. Chen, and P. Huang, "The diet-aware dining table: Observing dietary behaviors over a tabletop surface," in *International Conference on Pervasive Computing*. Springer, 2006, pp. 366–382.

[80] E. M. Tapia, S. S. Intille, and K. Larson, "Activity recognition in the home using simple and ubiquitous sensors," in *International conference on pervasive computing*. Springer, 2004, pp. 158–175.

[81] T. Atherton and D. Kerbyson, "Using phase to represent radius in the coherent circle hough transform," in *Hough Transforms, IEE Colloquium on*. IET, 1993, pp. 5–1.

[82] M.-K. Hu, "Visual pattern recognition by moment invariants," *IRE transactions on information theory*, vol. 8, no. 2, pp. 179–187, 1962.

[83] J. Han, L. Shao, D. Xu, and J. Shotton, "Enhanced computer vision with microsoft kinect sensor: A review," *IEEE transactions on cybernetics*, vol. 43, no. 5, pp. 1318–1334, 2013.

[84] K. Fukushima, "Neural network model for a mechanism of pattern recognition unaffected by shift in position-neocognitron," *IEICE Technical Report, A*, vol. 62, no. 10, pp. 658–665, 1979.

[85] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[86] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[87] X. Zhang, Z. Li, C. C. Loy, and D. Lin, "Polynet: A pursuit of structural diversity in very deep networks," in *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*. IEEE, 2017, pp. 3900–3908.

[88] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning." in *AAAI*, vol. 4, 2017, p. 12.

[89] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[90] A. Graves, M. Liwicki, H. Bunke, J. Schmidhuber, and S. Fernández, "Unconstrained on-line handwriting recognition with recurrent neural networks," in *Advances in neural information processing systems*, 2008, pp. 577–584.

[91] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 2625–2634.

[92] Y. Du, W. Wang, and L. Wang, "Hierarchical recurrent neural network for skeleton based action recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1110–1118.

[93] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[94] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[95] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[96] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," *arXiv preprint arXiv:1409.1259*, 2014.

[97] G. Kortuem, F. Kawsar, V. Sundramoorthy, D. Fitton *et al.*, "Smart objects as building blocks for the internet of things," *IEEE Internet Computing*, vol. 14, no. 1, pp. 44–51, 2009.

[98] M. Kranz and A. Schmidt, "Prototyping smart objects for ubiquitous computing," in *Workshop on Smart Object Systems, 7th International Conference on Ubiquitous Computing (Ubicomp)*, 2005.

[99] C. Guo, J. E. Young, and E. Sharlin, "Touch and toys: new techniques for interaction with a remote group of robots," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2009, pp. 491–500.

[100] I. Iacono and P. Marti, "Narratives and emotions in seniors affected by dementia: A comparative study using a robot and a toy," in *2016 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*. IEEE, 2016, p. 318.

[101] S. Jeong, C. Breazeal, D. Logan, and P. Weinstock, "Huggable: The impact of embodiment on promoting socio-emotional interactions for young pediatric inpatients," in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 2018, p. 495.

[102] C. Darwin and P. Prodger, *The expression of the emotions in man and animals*. Oxford University Press, USA, 1998.

[103] A. Hedge, "What am i sitting on? user knowledge of their chair controls," in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 60, no. 1. SAGE Publications Sage CA: Los Angeles, CA, 2016, pp. 455–459.

[104] M. Vergara and A. Page, "Relationship between comfort and back posture and mobility in sitting-posture," *Applied Ergonomics*, vol. 33, no. 1, pp. 1–8, 2002.

[105] S. J. Pedersen, C. M. Kitic, M.-L. Bird, C. P. Mainsbridge, and P. D. Cooley, "Is self-reporting workplace activity worthwhile? validity and reliability of occupational sitting and physical activity questionnaire in desk-based workers," *BMC public health*, vol. 16, no. 1, p. 836, 2016.

[106] P. T. Katzmarzyk, T. S. Church, C. L. Craig, and C. Bouchard, "Sitting time and mortality from all causes, cardiovascular disease, and cancer," *Medicine & Science in Sports & Exercise*, vol. 41, no. 5, pp. 998–1005, 2009.

[107] T. Widhe, "Spine: posture, mobility and pain. a longitudinal study from childhood to adolescence," *European Spine Journal*, vol. 10, no. 2, pp. 118–123, 2001.

[108] M. Adams and W. Hutton, "The effect of posture on the lumbar spine," *The Journal of bone and joint surgery. British volume*, vol. 67, no. 4, pp. 625–629, 1985.

[109] P. B. O'sullivan, W. Dankaerts, A. F. Burnett, G. T. Farrell, E. Jefford, C. S. Naylor, and K. J. O'sullivan, "Effect of different upright sitting postures on spinal-pelvic curvature and trunk muscle activation in a pain-free population," *Spine*, vol. 31, no. 19, pp. E707–E712, 2006.

[110] M. Kebaetse, P. McClure, and N. A. Pratt, "Thoracic position effect on shoulder range of motion, strength, and three-dimensional scapular kinematics," *Archives of physical medicine and rehabilitation*, vol. 80, no. 8, pp. 945–950, 1999.

[111] A. P. Claus, J. A. Hides, G. L. Moseley, and P. W. Hodges, "Is 'ideal'sitting posture real?: Measurement of spinal curves in four sitting postures," *Manual Therapy*, vol. 14, no. 4, pp. 404–408, 2009.

[112] S. Gracovetsky, M. Kary, S. Levy, R. S. Ben, I. Pitchen, and J. Helie, "Analysis of spinal and muscular activity during flexion/extension and free lifts." *Spine*, vol. 15, no. 12, pp. 1333–1339, 1990.

[113] H. Tanoue, T. Mitsuhashi, S. Sako, R. Goto, T. Nakai, and R. Inaba, "Effects of a dynamic chair on pelvic mobility, fatigue, and work efficiency during work performed while sitting: a comparison of dynamic sitting and static sitting," *Journal of physical therapy science*, vol. 28, no. 6, pp. 1759–1763, 2016.

[114] V. Cascioli, Z. Liu, A. Heusch, and P. W. McCarthy, "A methodology using in-chair movements as an objective measure of discomfort for the purpose of statistically distinguishing between similar seat surfaces," *Applied ergonomics*, vol. 54, pp. 100–109, 2016.

[115] R. Zemp, M. Fliesser, P.-M. Wippert, W. R. Taylor, and S. Lorenzetti, "Occupational sitting behaviour and its relationship with back pain–a pilot study," *Applied ergonomics*, vol. 56, pp. 84–91, 2016.

[116] J. Cheng, M. Sundholm, M. Hirsch, B. Zhou, S. Palacio, and P. Lukowicz, "Application exploring of ubiquitous pressure sensitive matrix as input resource for home-service robots," in *Robot Intelligence Technology and Applications 3*. Springer, 2015, pp. 359–371.

[117] P. Dario and D. De Rossi, "Tactile sensors and the gripping challenge: Increasing the performance of sensors over a wide range of force is a first step toward robotry that can hold and manipulate objects as humans do," *IEEE spectrum*, vol. 22, no. 8, pp. 46–53, 1985.

[118] D. Yamada, T. Maeno, and Y. Yamada, "Artificial finger skin having ridges and distributed tactile sensors used for grasp force control," in *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, vol. 2. IEEE, 2001, pp. 686–691.

[119] R. D. Howe, "Tactile sensing and control of robotic manipulation," *Advanced Robotics*, vol. 8, no. 3, pp. 245–261, 1993.

[120] B. B. Edin, "Quantitative analysis of static strain sensitivity in human mechanoreceptors from hairy skin," *Journal of neurophysiology*, vol. 67, no. 5, pp. 1105–1113, 1992.

[121] H. G. Wallbott, "Bodily expression of emotion," *European journal of social psychology*, vol. 28, no. 6, pp. 879–896, 1998.

[122] H. Miwa, "Effective emotional expressions with emotion expression humanoid robot we-4rii: Integration of humanoid robot hand rch-1," in *Proc. 2004 IEEE/RSJ Int. Conf. Intelligent Robot and Systems (IROS)*, 2004, pp. 2203–2208.

[123] G. Cannata, M. Maggiali, G. Metta, and G. Sandini, "An embedded artificial skin for humanoid robots," in *2008 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, Aug 2008, pp. 434–438.

[124] M. Maggiali, "Artificial Skin for Humanoid Robots," Ph.D. Thesis, University of Genova, 2008.

[125] P. Mittendorfer, E. Yoshida, and G. Cheng, "Realizing whole-body tactile interactions with a self-organizing, multi-modal artificial skin on a humanoid robot," *Advanced Robotics*, vol. 29, no. 1, pp. 51–67, 2015. [Online]. Available: http://dx.doi.org/10.1080/01691864.2014.952493

[126] R. Lowe, R. Andreasson, B. Alenljung, and E. Billing, "A wearable affective interface for the nao robot: A study of emotion conveyance by touch," *Submitted to IEEE Transactions on Affective Computing*, 2016.

[127] H. Knight, R. Toscano, W. D. Stiehl, A. Chang, Y. Wang, and C. Breazeal, "Real-time social touch gesture recognition for sensate robots," in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*. IEEE, 2009, pp. 3715–3720.

[128] A. Flagg and K. MacLean, "Affective touch gesture recognition for a furry zoomorphic machine," in *Proceedings of the 7th International Conference on Tangible, Embedded and Embodied Interaction*. ACM, 2013, pp. 25–32.

[129] Z. Průša, P. L. Søndergaard, N. Holighaus, C. Wiesmeyr, and P. Balazs, "The Large Time-Frequency Analysis Toolbox 2.0," in *Sound, Music, and Motion*, ser. Lecture Notes in Computer Science, M. Aramaki, O. Derrien, R. Kronland-Martinet, and S. Ystad, Eds. Springer International Publishing, 2014, pp. 419–442. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-12976-1_25

[130] S. Mallat, *A wavelet tour of signal processing*. Academic press, 1999.

[131] I. Daubechies, *Ten lectures on wavelets*. SIAM, 1992.

[132] B. Zhou, J. Cheng, A. Mawandia, Y. He, Z. Huang, M. Sundholm, M. Yildrim, H. Cruz, and P. Lukowicz, "Tpm framework: a comprehensive kit for exploring applications with textile pressure mapping matrix," 2017.

[133] A. Khotanzad and Y. H. Hong, "Invariant image recognition by zernike moments," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 12, no. 5, pp. 489–497, 1990.

[134] G. Ye, Z. Tang, D. Fang, X. Chen, K. I. Kim, B. Taylor, and Z. Wang, "Cracking android pattern lock in five attempts," 2017.

[135] K. Mowery, S. Meiklejohn, and S. Savage, "Heat of the moment: characterizing the efficacy of thermal camera-based attacks," in *Proceedings of the 5th USENIX conference on Offensive technologies*. USENIX Association, 2011.

[136] P. S. Intelligence, "Smartwatch market (2013-2023)," 2018.

[137] W. P. Hanten and S. S. Schulthies, "Exercise effect on electromyographic activity of the vastus medialis oblique and vastus lateralis muscles," *Physical Therapy*, vol. 70, no. 9, pp. 561–565, 1990.

[138] J. P. Ahtiainen, A. Pakarinen, M. Alen, W. J. Kraemer, and K. Häkkinen, "Short vs. long rest period between the sets in hypertrophic resistance training: influence on muscle strength, size, and hormonal adaptations in trained men." *The Journal of Strength & Conditioning Research*, vol. 19, no. 3, pp. 572–582, 2005.

[139] S. A. Plowman and D. L. Smith, *Exercise physiology for health fitness and performance*. Lippincott Williams & Wilkins, 2013.

[140] S. T. McCaw and D. R. Melrose, "Stance width and bar load effects on leg muscle activity during the parallel squat," *Medicine and science in sports and exercise*, 1999.

[141] A. Paoli, G. Marcolin, and N. Petrone, "The effect of stance width on the electromyographical activity of eight superficial thigh muscles during back squat with different bar loads," *The Journal of Strength & Conditioning Research*, vol. 23, no. 1, pp. 246–250, 2009.

[142] J. F. Signorile, K. Kwiatkowski, J. F. Caruso, and B. Robertson, "Effect of foot position on the electromyographical activity of the superficial quadriceps muscles during the parallel squat and knee extension." *The Journal of Strength & Conditioning Research*, vol. 9, no. 3, pp. 182–187, 1995.

[143] P. Lukowicz, F. Hanser, C. Szubski, and W. Schobersberger, "Detecting and interpreting muscle activity with wearable force sensors," *Lecture Notes in Computer Science*, vol. 3968, pp. 101–116, 2006.

[144] A. Reiss and D. Stricker, "Aerobic activity monitoring: towards a long-term approach," *International Journal of Universal Access in the Information Society (UAIS)*, 2013.

[145] H. Sakoe and S. Chiba, "Readings in speech recognition," A. Waibel and K.-F. Lee, Eds. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1990, ch. Dynamic Programming Algorithm Optimization for Spoken Word Recognition, pp. 159–165.

[146] A. Fradkin, T. Zazryn, and J. Smoliga, "Effects of warming-up on physical performance: a systematic review with meta-analysis," in *J Strength Cond Res.*, 2010.

[147] D. High, E. Howley, and F. Don, "The effects of static stretching and warm-up on prevention of delayed-onset muscle soreness," in *Research Quarterly for Exercise and Sport*, 1989, pp. 357–361.

[148] M. Mohr, P. Krustrup, L. Nybo, J. Nielsen, and B. J., "Muscle temperature and sprint performance during soccermatches – beneficial effect of re-warm-up at half-time," in *Scand J Med Sci Sports*, 2004, pp. 156–162.

[149] P. Madeleine, P. Bajaj, K. Søgaard, and L. Arendt-Nielsen, "Mechanomyography and electromyography force relationships during concentric, isometric and eccentric contractions," *Journal of electromyography and kinesiology*, vol. 11, no. 2, pp. 113–121, 2001.

[150] C. Orizio, M. Gobbo, B. Diemont, F. Esposito, and A. Veicsteinas, "The surface mechanomyogram as a tool to describe the influence of fatigue on biceps brachii motor unit activation strategy. historical basis and novel evidence," *European journal of applied physiology*, vol. 90, no. 3-4, pp. 326–336, 2003.

[151] T. W. Beck, T. J. Housh, G. O. Johnson, J. P. Weir, J. T. Cramer, J. W. Coburn, and M. H. Malek, "Mechanomyographic and electromyographic time and frequency domain responses during submaximal to maximal isokinetic muscle actions of the biceps brachii," *European journal of applied physiology*, vol. 92, no. 3, pp. 352–359, 2004.

[152] E. Fujiwara and C. K. Suzuki, "Optical fiber force myography sensor for identification of hand postures," *Journal of Sensors*, vol. 2018, 2018.

[153] D. Esposito, E. Andreozzi, A. Fratini, G. Gargiulo, S. Savino, V. Niola, and P. Bifulco, "A piezoresistive sensor to measure muscle contraction and mechanomyography," *Sensors*, vol. 18, no. 8, p. 2553, 2018.

[154] I.-M. Lin and E. Peper, “Keep cell phones and pdas away from emg sensors and the human body to prevent electromagnetic interference artifacts and cancer,” *Biofeedback*, vol. 37, no. 3, pp. 114–116, 2009.

[155] B. Zhou, G. Bahle, L. Fürg, M. S. Singh, H. Z. Cruz, and P. Lukowicz, “Trainwear: A real-time assisted training feedback system with fabric wearable sensors,” in *Pervasive Computing and Communications Workshops (PerCom Workshops), 2017 IEEE International Conference on*. IEEE, 2017, pp. 85–87.

[156] B. Zhou, H. Koerger, M. Wirth, C. Zwick, C. Martindale, H. Cruz, B. Eskofier, and P. Lukowicz, “Smart soccer shoe: monitoring football interaction with shoe integrated textile pressure sensor matrix,” in *Proceedings of the 2016 ACM International Symposium on Wearable Computers*. ACM, 2016, pp. 64–71.

[157] I. Palacios-Huerta, “Structural changes during a century of the world’s most popular sport,” *Statistical Methods and Applications*, vol. 13, no. 2, pp. 241–258, 2004.

[158] T. Reilly and D. Gilbourne, “Science and football,” *Journal of Sports Sciences*.

[159] N. De Mestre, *The mathematics of projectiles in sport*. Cambridge University Press, 1990, no. 6.

[160] K. Bray and D. Kerwin, “Modelling the flight of a soccer ball in a direct free kick,” *Journal of sports sciences*, vol. 21, no. 2, pp. 75–85, 2003.

[161] R. D. Mehta and J. M. Pallis, “Sports ball aerodynamics: effects of velocity, spin and surface roughness,” *Minerals, Metals and Materials Society/AIME, Materials and Science in Sports(USA),*, pp. 185–197, 2001.

[162] L. Oggiano and L. Sætran, “Aerodynamics of modern soccer balls,” *Procedia Engineering*, vol. 2, no. 2, pp. 2473–2479, 2010.

[163] S. Barber, S. J. Haake, and M. Carré, “Using cfd to understand the effects of seam geometry on soccer ball aerodynamics,” in *The Engineering of Sport 6*. Springer, 2006, pp. 127–132.

[164] S. Hong and T. Asai, “Effect of panel shape of soccer ball on its flight characteristics,” *Scientific reports*, vol. 4, 2014.

[165] D. Schuldhaus, C. Zwick, H. Körger, E. Dorschky, R. Kirk, and B. M. Eskofier, “Inertial sensor-based approach for shot/pass classification during a soccer match.”

[166] J. S. Akins, “Development and evaluation of instrumented soccer equipment to collect ankle joint kinematics in the field,” Ph.D. dissertation, University of Pittsburgh, 2013.

[167] P. Cappa, A. Pacilli, E. Palermo, and S. Rossi, “Mobile motion capturing in sport session based on inertial measurement units.”

[168] A. H. Abdul Razak, A. Zayegh, R. K. Begg, and Y. Wahab, “Foot plantar pressure measurement system: A review,” *Sensors*, vol. 12, no. 7, pp. 9884–9912, 2012.

[169] E. Eils, M. Streyl, S. Linnenbecker, L. Thorwesten, K. Völker, and D. Rosenbaum, “Characteristic plantar pressure distribution patterns during soccer-specific movements,” *The American Journal of Sports Medicine*, vol. 32, no. 1, pp. 140–145, 2004.

[170] Y. Weizman and F. K. Fuss, “Sensor array design and development of smart sensing system for kick force visualization in soccer,” *Procedia Technology*, vol. 20, pp. 138–143, 2015.

[171] K. Witte and H. Körger, “Optimierung der ballschussmaschine robyleg ii und untersuchung des deformationsverhaltens von fußbällen,” 2010.

[172] B. Zhou, J. Cheng, M. Sundholm, and P. Lukowicz, “From smart clothing to smart table cloth: Design and implementation of a large scale, textile pressure matrix sensor,” in *International Conference on Architecture of Computing Systems*. Springer, 2014, pp. 159–170.

[173] P. Horowitz and W. Hill, *The art of electronics*. Cambridge university press Cambridge, 1989, vol. 2.

[174] D. Kidner, M. Dorey, and D. Smith, “What’s the point? interpolation and extrapolation with a regular grid dem,” in *Fourth International Conference on GeoComputation, Fredericksburg, VA, USA*, 1999.

[175] Z. O. Khokhar, Z. G. Xiao, and C. Menon, “Surface emg pattern recognition for real-time control of a wrist exoskeleton,” *Biomedical engineering online*, vol. 9, no. 1, p. 41, 2010.

[176] F. Lotte, “A new feature and associated optimal spatial filter for eeg signal classification: waveform length,” in *Pattern Recognition (ICPR), 2012 21st International Conference on*. IEEE, 2012, pp. 1302–1305.

[177] W. Yang, K. Wang, and W. Zuo, “Neighborhood component feature selection for high-dimensional data.” *JCP*, vol. 7, no. 1, pp. 161–168, 2012.

[178] H. Hotelling, “Analysis of a complex of statistical variables into principal components.” *Journal of educational psychology*, vol. 24, no. 6, p. 417, 1933.

[179] E. Moulines and F. R. Bach, “Non-asymptotic analysis of stochastic approximation algorithms for machine learning,” in *Advances in Neural Information Processing Systems*, 2011, pp. 451–459.

Curriculum Vitae

Bo Zhou

Email: bo.zhou@dfki.de
Web: www.researchgate.net/profile/Bo_Zhou30

My research interests are sensing technology in the scope of human-machine interaction and ubiquitous/wearable computing, especially in the scenarios of sports, well-being, and psychology applications. I have also invested effort in combining those technologies with fashion and ergonomics, since I believe that these are the driving factors for novel technologies to be accepted by the wider audience.

Education

2007-2011	Bachelor degree in Technology and Instrumentation of Measurement and Control,
	Navigation specialty ,Harbin Engineering University, Harbin, China
2011-2013	Erasmus Mundus European Master in Embedded Computing Systems (EMECS)
	2011-2012 System-on-Chip, University of Southampton, Southampton, UK
	2012-2013 Embedded Systems, Technical University of Kaiserslautern, Kaiserslautern, Germany

Experience

since 2013	Full-time researcher at DFKI, Kaiserslautern, Germany
	working mainly on smart textiles and ubiquitous sensing
	2013-2016 EU FP7 project SimpleSkin (simpleskin.org)
	since 2018 BMBF project HeadSense
since 2016	assisting tutor of Prof. Lukowicz in the lecture Embedded Intelligence in TU Kaiserslautern