A wearable unistroke textile touchpad

Bachelor's Thesis
submitted to the
Media Computing Group
Prof. Dr. Jan Borchers
Computer Science Department
RWTH Aachen University

by
Christian Schmidt

Thesis advisor:
Prof. Dr. Jan Borchers

Second examiner:
Prof.Dr.Martina Ziefle

Registration date: 04.03.2016
Submission date: 31.03.2016

Eidesstattliche Versicherung

______________________________ ______________________________

Name, Vorname Matrikelnummer

Ich versichere hiermit an Eides Statt, dass ich die vorliegende Arbeit/Bachelorarbeit/
Masterarbeit* mit dem Titel

____

____

____

selbständig und ohne unzulässige fremde Hilfe erbracht habe. Ich habe keine anderen als die angegebenen Quellen und Hilfsmittel benutzt. Für den Fall, dass die Arbeit zusätzlich auf einem Datenträger eingereicht wird, erkläre ich, dass die schriftliche und die elektronische Form vollständig übereinstimmen. Die Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

______________________________ ______________________________

Ort, Datum Unterschrift

*Nichtzutreffendes bitte streichen

Belehrung:

§ 156 StGB: Falsche Versicherung an Eides Statt

Wer vor einer zur Abnahme einer Versicherung an Eides Statt zuständigen Behörde eine solche Versicherung falsch abgibt oder unter Berufung auf eine solche Versicherung falsch aussagt, wird mit Freiheitsstrafe bis zu drei Jahren oder mit Geldstrafe bestraft.

§ 161 StGB: Fahrlässiger Falscheid; fahrlässige falsche Versicherung an Eides Statt

(1) Wenn eine der in den §§ 154 bis 156 bezeichneten Handlungen aus Fahrlässigkeit begangen worden ist, so tritt Freiheitsstrafe bis zu einem Jahr oder Geldstrafe ein.

(2) Straflosigkeit tritt ein, wenn der Täter die falsche Angabe rechtzeitig berichtigt. Die Vorschriften des § 158 Abs. 2 und 3 gelten entsprechend.

Die vorstehende Belehrung habe ich zur Kenntnis genommen:

______________________________ ______________________________

Ort, Datum Unterschrift

Contents

List of Figures

List of Tables

Abstract

This bachelor thesis describes the development of a wearable 2D textile touchpad able to be embedded into everyday clothing. It is a wearable, bendable, washable, and breathable touch sensor designed for eyes-free interaction. The touchpad is basically composed of 3 textile layer, two perpendicular layers of fabric with conductive stripes, separated by a non conductive material. All materials used for manufacturing are low cost and lightweight. The prototype is based on resistive touch, hence it is limited to sensing touch at a single location. The touch coordinates are continuously sent to a gesture recognizer allowing the prototype to detect unistroke gestures. With these gestures can be used to control several applications in a mobile scenario. The physical limitations are evaluated in an informal user study by testing the performance under various conditions.

Überblick

In dieser Bachelorarbeit wird die Entwicklung eines tragbaren, biegsamen, waschbaren und atmungsaktiven 2D Touchpads beschrieben. Der Sensor besteht aus aus zwei Schichten Stoff, der mit leitfähige Bahnen versehen ist, und Schaumstoff, der die beiden orthogonalen Stoffschichten trennt. Alle benutzte Materialien sind leicht und kostengünstig. Der Prototyp basiert auf der resistiven Methodik eine Berührung zu erkennen. Das heißt, dass der Sensor auf eine einzige Position beschränkt ist. Diese Positionen werden kontinuierlich an ein Programm, dass aus den einzelnen Punkten Gesten erkennt, geschickt. Damit können dann diverse Anwendungen insbesondere im mobilen Bereich gesteuert werden. Außerdem haben haben Benutzer den Prototypen unter verschiedenen Bedingungen getestet, um die physikalischen Einschränkungen zu bestimmen.

Acknowledgements

Thank you!

Conventions

Throughout this thesis we use the following conventions.

Text conventions

Definitions of technical terms or short excursus are set off in coloured boxes.

EXCURSUS:
Excursus are detailed discussions of a particular point in a book, usually in an appendix, or digressions in a written text.

Definition: *Excursus*

Source code and implementation symbols are written in typewriter-style text.

`myClass`

The whole thesis is written in Canadian English.

Download links are set off in coloured boxes.

File: myFile[a]

[a]http://hci.rwth-aachen.de/public/folder/file_number.file

Chapter 1

Introduction

Motivation and well known wearables

Electronics are getting smaller, lighter and powerful every year and we reached a point where they have actually become wearable. Therefore the research in the field of wearable computing increased over the last decade. The ultimate goal is to make life easier and more comfortable by integrating controls and sensors even more in your daily life. Health tracking devices are the leading wearables at the moment. Smartwatches are even capable of various smartphone features such that it is less often necessary to take your smartphone out of your pocket.

Smart clothing

One of the first contributions to the field of wearables were made by Post and Orth [1997]. They emebedded easy to build, washable textile based sensors, buttons, and switches into a jacket. Rantanen et al. [2002] integrated a computer including screen and battery into an arctic suite to provide the wearer with information about the surrounding conditions, their location, and controls for the integrated heating system. Brewster et al. [2003] investigated the opportunities of eyes-free hand gestures on a PDA attached to the waist, supported by audio feedback. They found that gestures were performed more accurate when acoustic feedback is provided.

The focus of this thesis lies on wearable textile input devices for eyes-free interaction integrated into everyday clothing. Devices for eyes-free interaction are primary designed for a mobile context where the visual channel is occupied by the environment. While driving a car changing the radio station, skipping a song, and answering a call including activating the hands-free feature on your mobile phone is one example where a wearable touchpad on your thigh would be helpful. Various approaches were presented aiming to create wearable input devices over the last decades. The most used technique for sensing a touch is capacitive touch sensing. [Holleis et al., 2008] built textile prototypes by sewing conductive thread in fabric creating touch sensing buttons. The buttons are discrete input elements and not continuous. However, their research on conductive resulted in several guidelines applicable for this field in general. We try to refine them further with the knowledge we gathered with this thesis.

Limitation and improvements

Most of the textile touchpads today are based on capacitive touch and rather prone to noise. The number of gestures they are able to distinguish reliable is quite limited not to mention multitouch. To improve their performance the influence of the human body has to be minimized. Another possibility is to use another approach which is not affected by that noise at all. Resistive touch is a technology used in PDAs and early smartphones.

Our contributions

In this thesis we present our novel wearable, bendable, washable, scalable, resistive textile 2D touchpad. We provide a detailed description how to build this sensor with low cost materials. We show that our prototype is able to reliably detect various unistroke gestures and evaluate its robustness under several conditions. The software for operating the sensor is provided here Schmidt [2016].

Chapter 2

Related work

This chapter reviews related research in the area of interactive textile. We divided this review into two parts: interactive textile technology and textile touch pads. In the first part we give an overview of ways to integrate and activate textile in everyday objects. In the seconds part we look at research that investigated different techniques to fabricate textile surfaces that can detect user touches and gestures.

2.1 General overview

2.1.1 Resistive vs. capacitive touch

The two most popular touch input technologies are resistive and capacitive. Both serve the same purpose but the underlying principle differs making them more or less suited for wearable computing.

Characteristics of Capacitive Touch

Capacitive touch uses a non-conducting material with conductive material underneath. The capacitance of the human body changes the electrical field of the sensor which is measurable. The advantages of capacitive touch is the easy support for multi-touch input and high resolution. These touch screens only need a slight touch without force. The main disadvantage is the human body itself, since it generates its own capacitive field which makes it hard to detect intentional touches. This flaw is intensified by the body movement continuously changing the proximity between the sensor and the human body. Therefore complex isolation techniques are required to isolate the sensor and the human body which is unfeasible for fast prototyping.

Characteristics of Resistive Touch

Resistive touch technology uses two separated layers of striped electrodes such that it is arranged to a matrix. The spacing material in ordinary resistive touch screens is either an air filled chamber or a non-conductive material which separates both layers while no external force is applied. Therefore one can operate it with a stylus or with gloves since no conductivity is required. On the one hand this solves the main disadvantage of the capacitive method regarding wearable computing, on the other hand it does not support multi-touch and is still prone to deformation and thus to unintentional contacts.

2.1.2 Textile interaction techniques

Holleis et al. [2008] presented several textile prototypes based on capacitive sensing. They embroidered conductive wires to a phone case, a glove, and an apron resulting in small conductive buttons. Besides that, they used conductive foil for buttons on a helmet as well. Their user study however only evaluated the apron with three different button layouts with different visibility. Based on the results they present guidelines for wearable controls such as locating and identifying controls must be quick and easy.
Speir et al. [2014] built two prototypes, a wristband and a glove. The wristband prototype uses a circular conductive fabric surrounded by resistive linqstat. A conductive finger cap connects both of them and generates a value which is used to determine the location and the movement of the touch. The glove works on the same principle. They evaluated their prototypes as remote controls for an iPod using one- and two-handed interaction and have found that the users have no clear preference.

Ubiquitous drums

A different application is presented by Smus and Gross

[2010]. They used force-sensitive resistors and pull-down resistor circuits to sense percussive touch. They taped the sensor into the inside of a pair of jeans and to the sole of a shoe. They created a program that translates the sensor values to different parts of a drum.

Time Domain Reflectometry

[Wimmer and Baudisch, 2011] created 13 prototypes based on time domain reflectometry (TDR). For this approach only one pair of wires is needed. The change in capacitance caused by conductive objects close to the pair of wire is measured and the location determined. Distance between the wires and their shape have significant influence on reliability. Their prototypes include stretchable, curved, and arbitrary shaped surfaces. They can sense touch at a distance up to 20m but TDR is prone to radio interference of mobile phones.

2.2 Textile touch pads

Pinstripe

Pinstripe is a continous textile input prototype created by Karrer et al. [2010]. It detects pinching and rolling of clothing by connecting conductive thread sewn on it. However, it is an unidimensional input device and not a touch pad. Pinstripe is able to detect Nevertheless, when they introduced the participants to the sensor the users intuitively expected a touch pad. This shows that textile touchpads as an input device are not declined in general.

Grabrics

Grabrics by Hamdan et al. [2016] is a fold-based textile sensor that can detect the axis of a pinch and the displacement and direction of the user's thumb over the fold. It, however, cannot detect complex gesture because of the limited resolution.

GesturePad

Rekimoto [2001] presented GesturePad which is a capacitive touch pad integrated in clothing. They propose slightly different architectures consisting of the upper fabric, receiver, transmitter, and a shield layer to reduce the influence of the human body. However, their work was not further evaluated.

PocketTouch

Another capacitive approach was developed by Saponas et al. [2011]. PocketTouch is an eyes-free, calibrateable capacitive touch pad which can sense the proximity of a fin-

Prototype	Touch technology	Gesture detection
Pinstripe	capacitive	detects size of pinch and roughly movement of pinch in 1D
GesturePad	capacitive	not tested (theoretically able to detect 2D gestures)
Pocket touch	capacitive	not tested (multi-stroke recognizer N$ by Anthony and Wobbrock [2012] implemented
FabriTouch	capacitive	vertical swipes
Grabrics	resistive	detects axis of fold and movement of pinch in 2D

Table 2.1: Current textile touch pad technologies.

ger through a wide range of fabrics. They used a touch sensor of a touch screen and attached it to a base which makes it not bendable. The reliability of PocketTouch was not further evaluated.

FabriTouch

FabriTouch is a flexible, capacitive textile touch pad presented by Heller et al. [2014]. It consists of lining, piezoresistive foil, spacing mesh, conductive fabric, and outer garment together integrated into a pair of trousers. FabriTouch is able to detect vertical swipe gestures on the human thigh. Movement has a negative impact on the performance of the sensor.

Going for Resistive Touch

After taking all characteristics into account we decided to go for the resistive touch technology, because we can drop all considerations of capacitive noise caused by the human body.

Chapter 3

Hardware Prototype and Software Development

In this chapter we will present the hardware prototypes. Furthermore, we describe the disadvantages and improvements of former iterations leading to the final prototype. The software implementation is described afterwards.

3.1 System Design

All prototypes presented here are using pinstripe, a fabric with separated conductive lines. For the first prototype we used the Texas Instruments MSP430G2452 microcontroller. Each row and column of the pinstripe fabric has to be connected to a *digitalRead* pin of the micro-controller. The MSP430 controller has 16 of these pins but only 14 can be used since two pins are used for serial communication. This results in a matrix resolution of 7 by 7 at maximum.

MSP430 for first prototype

RESOLUTION IN PINSTRIPE CONTEXT:
When speaking of a certain resolution of our prototype, we talk about the number of connected rows and columns. Since the pinstripe fabric is of a fixed size (3mm conductive material and 3mm spacing). Therefore the higher the resolution the larger the prototype gets.

Definition: *Resolution in pinstripe context*

We use the TI EK-TM4C1294XL for the advanced prototypes. This board has the ability to connect more than 40 pins for *digitalRead* to operate a 20 by 20 sensor. The board is connected to a Computer via USB for serial communication. Short range wireless communication with Bluetooth is added to the final prototype.

Programming Environment: Energia IDE and Processing to gather, send and process input data

For programming the micro-controller we are using the Energia IDE[1]. It is an easy to use IDE to upload programs to the TI micro-controller. The micro-controller itself is solely responsible for sending the data of the sensor to the computer via serial communication. Meaning that it tests a pin against ground for each other line and column. A *1* is written to the serial-port when it is connected to another line or column and *0* otherwise. This is done for each pin where *numberOfPintripes* is the number of all lines and columns. For each prototype an integer array is declared and can easily be commented and uncommented depending on the prototype. The pin numbers are sorted such that the first pins correspond to the x-axis and the last pins to the negative y-axis. After all pins were tested we send a line-break to determine the end of the current input.

We are using Processing[2], a Java based IDE, to structure the input stream from the microcontroller for further analyses. This includes several programs which either display the raw touch points for debugging purpose or filter and display the sensor data. The changes of software are described along with each hardware iteration.

[1]http://energia.nu
[2]http://processing.org

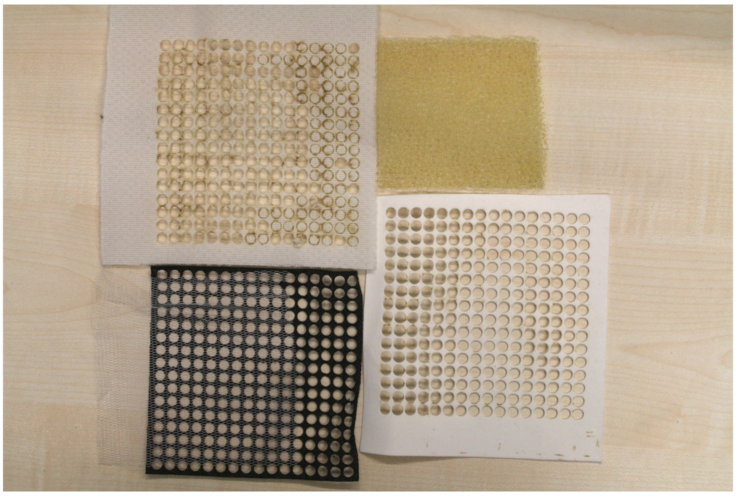

Figure 3.1: Materials used for testing (polyester grid fabric, plastic latticework, jeans with fly screen, and microcellular rubber).

3.2 Early Testing

After deciding to go for the resistive approach, the essential challenge is to find a spacing material with certain characteristics. The material should

- be flexible by means of being wearable
- reliably separate the pinstripe fabric while no touch is intended
- and concede easily when intended force is applied.

We glued both layers of the pinstripe fabric to sheets of paper to eliminate stretching and curling of the fabric. We cut equidistant circular holes to provide space for the pinstripe layers to connect. To display where a touch is present, we created a simple program with Processing. Some of the materials we tested are shown in figure 3.1 but were not well suited as foam which we used for the latest prototypes.

3.3 The Prototype

The prototype uses a 3 mm thick layer of foam, coated with a thin layer of cotton, for spacing. Foam has the properties we need to separate the pinstripe layer while no force is applied and yields rather easily when the user presses on it. Another positive feature is the increased resistance to unintended pressure caused by bending or accidental contact with the sensor.
We use a laser cutter to cut equidistant holes out of the foam to allow the pinstripe layers to connect. This procedure leaves enough foam between the holes to retain the properties we need.

Issues with flexibility and translatory movement

Another problem we have to address is the stretchability and the translatory movement of the fabrics. Each time the user performs a gesture the upper layer moves in the respective direction due to the friction between the operating finger and the surface. This causes the the pinstripe fabric to shift such that the conductive stripes are not aligned to the holes properly, resulting in the prototype to stop working. When we started testing we used needles, clips, and nails to fix the materials to each other. Not only that these methods are not well suited for wearablity, also fixing the materials exclusively at the edges is not sufficient. The flexibility of the pinstripe fabric can causes shifts that we want to eliminate.

Vliesofix for fixating the components of the prototype

To deal with the issues described above we use Vliesofix. Vliesofix is an adhesive on paper which can be ironed on textile. The paper can be removed afterwards and another fabric can be ironed to the corresponding textile. This results in an extensive adhesive area between two fabrics. The application of Vliesofix not only resolves the translatory movement but also the curling of the pinstripe fabric.

We decide two to prototypes with different dimensions shown in figure 3.2. The first prototype has a resolution of 14 by 14 pinstripes and the second prototype has a resolution of 20 by 20. Since the procedure of making the prototypes is similar we only describe its building procedure for the smaller one.

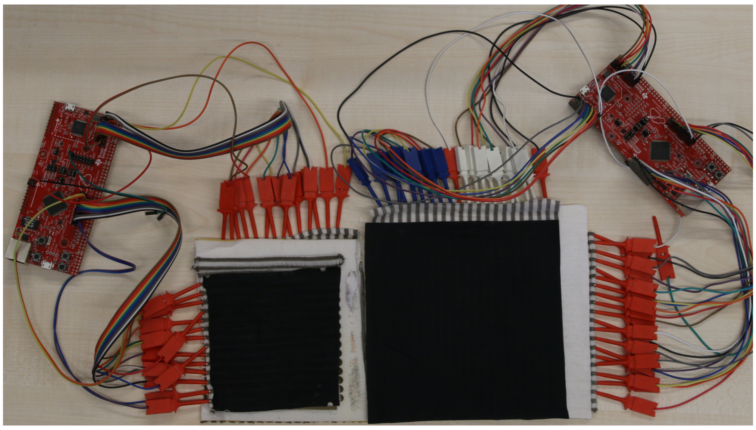

Figure 3.2: The 14 by 14 pin prototype on the left and 20 by 20 pin prototype on the right

We start by cutting out a 137 by 125 cm piece of the 3 mm foam using the laser cutter. Then we cut out two sheet of Vliesofix with the corresponding dimensions and proceed by ironing them on both sides of the foam. This has to be done carefully since applying heat for too long can cause the foam to melt. As a result, the foam loses the desired properties to a certain degree. When removing the iron too soon the Vliesofix might not be adhesive enough.

Assembling the prototype

Once the material is cooled off we can remove the paper of the Vliesofix. We proceed with cutting the holes in the foam with the Vliesofix. Making the laser cutter cut more rows and columns of holes ensures that the resistance of the foam is the same throughout the touch sensitive area. Otherwise more force is needed at the edges than the center.
To prepare the two pinstripe layer to adhere it to the foam we again use Vliesofix first for preventing the fabric from curling and easy stretching. We do so before cutting out the pinstripe to make handling the fabric easier. Note that, while using Vliesofix with the pinstripe fabric, it is even more important to not apply heat for too long. When the Vliesofix becomes liquid it gets soaked into the fabric. In some cases we ascertained that for this reason the conductivity of the pinstripes gets lost in some places. This immediately renders the sensor useless. Apart from that we do not remove the paper of the Vliesofix at this point.

Figure 3.3: The layer of the prototype in correct order with Vliesofix already applied to the lower pinstripe fabric and jeans as potential upper layer.

We proceed with attaching the pinstripe layers to the foam. The stripes of each layer must be perpendicular to one another. Then we iron the untreated side of the one pinstripe fabric to the foam and after cooling off the other pinstripe to the other side. We have to make sure that the conductive stripes and the holes in the foam are aligned properly. Now we can remove the paper from the pinstripe fabric. The arrangement of the materials is shown in figure 3.3.

After that we iron a corresponding piece of polyester on the upper side of the sensor to reduce the friction between the finger and treated pinstripe layer. The last step is to connect the sensor to the micro controller. Furthermore we connect a HC06 Bluetooth module for wireless data stream. The power source can either be a computer or an battery pack which are connected by a micro USB cable.

3.4 The Software

The software is divided into two parts, the code running on the micro controller and the application running on the computer. The code for the micro controller is straight forward. We define a one dimensional array for each prototype in which the pin numbers are stored such that the first half of the array denotes the horizontal x-axis starting from left to right and the second half the vertical y-axis starting from top to bottom. Now we can check each pin of the array respectively if it is connected to ground. This means that the conductive stripe connected to that pin has a connection to another stripe and a *1* is sent via the serial communication and a *0* otherwise. The line break after the *for* loop lets us determine if a complete input set is received.

Receiving the data via serial communication in Processing

In Processing we read the data from the serial communication and store it in a buffer including the time stamp. If at least one *1* per data set is read, we consider it to be a touch. Sometimes the contact of the pinstripe layer is lost while performing a gesture due to insufficient pressure or a undersized locating surface of the operating finger. Therefore we implemented a threshold in which we can read only *0* without the touch phase to end. This threshold is reset anytime a *1* is read. The raw data is logged for potential analyses.

Use a filter algorithm to reduce a set of touch points to one point

Since more than one vertical and horizontal pinstripe can be connected at the same time we apply a filter algorithm. This algorithm takes all x-coordinates and calculates the average and the same for the y-coordinates. The resulting triple composed of the coordinate and the time stamp is added to an *Arraylist* buffer. This is done for all input sets except for all those, which only consist of *0's*. There are two reasons for filtering the input data. The first is the resilience to noise caused by unintended connections or slow separation of the pinstripe layers. The second reason is fact that it is easier for implementing gesture recognition. Apart from this the user intends to press only one point at the sensor.
When a gesture starts we take the coordinates and subtract them from all filtered coordinates of this gesture. Therefore every gesture starts at $(0, 0)$ regardless where it is performed on the sensor. This is done for the graphical representation of the strokes and will be useful later on.

Implementing mark-based gesture recognition

The next step is to actually recognize easy strokes

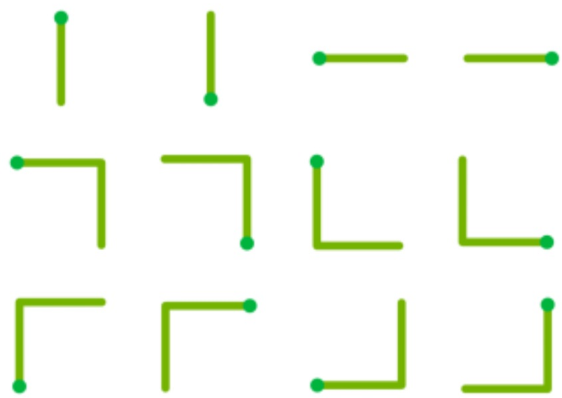

Figure 3.4: Mark-based gestures. Gestures start at the dots. Bragdon et al. [2011]

performed on the prototype. Since our prototypes have a rather low resolution compared to typical touch input devices we focus on rather simple unistroke gestures. These mark-based gestures are shown in figure 3.4. We extend this gestures set by adding 5 gestures. These gestures are a tab and for each swipe gesture we expand it with a swipe in the opposite direction. This leads to a gesture set with 17 simple gestures.
Our gesture recognition works as follows. First of all we do not recognize in real-time. We analyze all filtered points, stored in the buffer, after a gesture is considered finished. Then the algorithm works as follows.

- check for tap
  - return *true* if the size of the buffer is 1
  - return *true* if all coordinates have a *distance* smaller or equal to 1 respectively and the time elapsed is smaller than 200 milliseconds
  - else check for other gestures
- check for swipe
  - assume there is a swipe with the first and the last item of the buffer as terminal points

  - calculate the $length$ of the line
  - return $false$ if the $length$ is smaller than 3
  - for all other points calculate the $distanceToLine$
  - return $false$ if at some index the $distanceToLine$ is greater than 1
  - else determine the direction of the swipe and return $true$

- check for angle
  - check if there is a line between the point at $index - 1$ and last point in the buffer under the exact same conditions applied for swipe
  - return $false$ if at some index the $distanceToLine$ is greater than 1
  - calculate the directions of both lines and return $true$

- end of checking

This algorithm classifies each gesture as a tap, swipe, angle, or no gesture. In combination with the directions we calculate for each swipe we can distinguish between all 16 mark-based gestures. The orientation of a line is mapped to one of the four directions up, left, right, or down. Therefore our prototype is resilient to a certain degree of input error. With less effort we can further extend the gesture set by distinguish more directions.

Using 1$ Unistroke Recognizer for complex gestures

When testing our gesture recognizer with our current prototypes we observe an almost 100% recognition rate. Based on this finding we decide to go beyond simple mark-based gestures and continue with recognizing more complex gestures. Therefore we make use of the 1$ Unistroke Recognizer by Wobbrock et al. [2007]. This is an easy to implement recognizer which does not require any training data. Providing a template for each gesture is sufficient. A template is an array of consecutive pairs of coordinates. We can pass the coordinates in the filtered

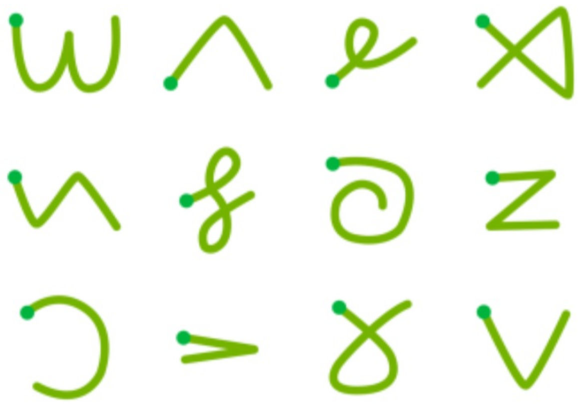

Figure 3.5: Free-form gestures Bragdon et al. [2011]

buffer straight to the 1$ recognizer.
This recognizer is orientation independent. Meaning for the marked-based gestures that, without further analysis of orientation, we can only distinguish between a swipe, an angle to the right, and an angle to the left. However, we can recognize a set of free-form gestures shown in figure 3.5.

Chapter 4

System Evaluation

In this chapter we will evaluate the robustness of our prototype in different extreme conditions. We will take a closer look at the performance of the 14 by 14 prototype. Since the prototype is designed as a wearable, we are interested in its behavior under changing conditions. These conditions are composed of softness, curvature, and friction.

Testing the 14 by 14 prototype

4.1 Physical Limitation Study

The human body is in motion almost all the time and the clothes we are wearing are not fixed to the skin. This *looseness* and the changing subsurface are variables that influence the performance of our prototype. Another variable is the *friction* of the overlaying common everyday fabrics. Depending on the fabric and method of fashioning, it can, more or less likely, happen that the user slips of the touch-sensing area, or experiences an unpleasant feeling in the operating finger. Furthermore the *softness* of the underlying surface may influence the performance of our prototype. The amount of muscles, adipose tissue, and so forth also differs from human to human. This, in the first place, affects the pressure needed by the user. Then there are the different levels of *curvature*. Our prototype has flexible spacing-material to separate the pinstripe layer. After

Independent variables: friction, softness, looseness, and curvature

variable	levels
curvature	3 (flat, 66mm diameter, 53mm diameter)
softness	2 (solid, foam 1000m^3 density)
friction	3 (cotton, jeans, rib knit cotton)

Table 4.1: Variables and their levels.

a certain amount of bend the material starts creasing, causing some permanent contacts. In this study we will test our prototype in conditions which aim to simulate the in field scenarios. To test to which degree of bend the prototype breaks we used different foams with fixed thickness and different density.

4.2 Experiment Setup

The conditions

The conditions and their levels are shown in table 4.1. This leads to 18 combinations in total. For softness we have chosen the solid surface as a baseline. The soft foam with a density of 1000m^3 was considered similar enough to the soft spots on the human body. Curvature has a flat surface as baseline, 66mm diameter curvature, and 53mm diameter as fringe condition. Prior testing has proven that going below 53mm leads to permanent contact due to a kink in the spacing material. Friction is depending on the materials used for the outer layer of the sensor and the clothing, respectively. We have decided to test cotton, jeans and rib knit cotton shown in figure 4.1. They have distinct surface characteristics and behavior when moving across with the finger.

System setup

The users sat in front of a desk on which the conditions were prepared consecutively. The sensor was fixed to the surface below and the overlying fabric was fixated in the corners with pins. Nevertheless, there is still a certain amount of movement due to the flexibility of the overlying fabrics. For the curvature, we used aerosol cans with 53mm diameter and 66mm diameter. We fixated the aerosol cans with stands made with a laser cutter. To achieve the curvature with the foam, we used a book and

Figure 4.1: The materials used in the experiment (cotton, jeans, and rib knit cotton).

wrapped the foam around the cover and hemmed it in a vise. A GoPro camera was placed such that each setup was captured obliquely from above as shown in figure 4.2 and figure 4.3. The observer sat next to the participant ready to make notes and start or stop recording the setup. The user cannot see the output on the screen. Additionally, our program created two log-files for each condition. One logged the filtered data and one the raw sensor data. Both files logged the time stamps of each data point.

Study design and participants

We asked two right-handed participants, one male (24) and one female (22) to test our prototype. One had no experience with wearables. The participants had to perform 8 gestures in each condition with three repetitions. We selected a within-subject design for the evaluation since we only let two users with different experience participate. Thus, each participant had to perform 423 (18 conditions + 8 gestures * 3 repetitions) gestures not including potential repetitions. The set of gestures is shown in figure 4.4. Curvature and softness were counterbalanced. Since each change of a condition takes several minutes we decided to shorten the time for the participant. To do so we tested the upper fabrics consecutively.

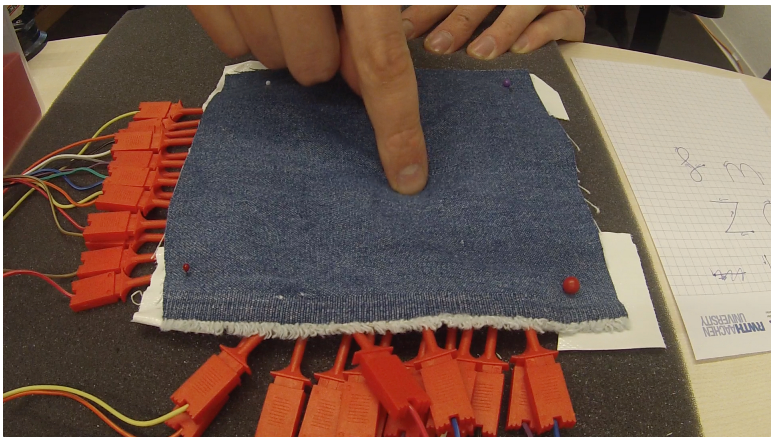

Figure 4.2: Condition: flat, jeans, foam, on the table

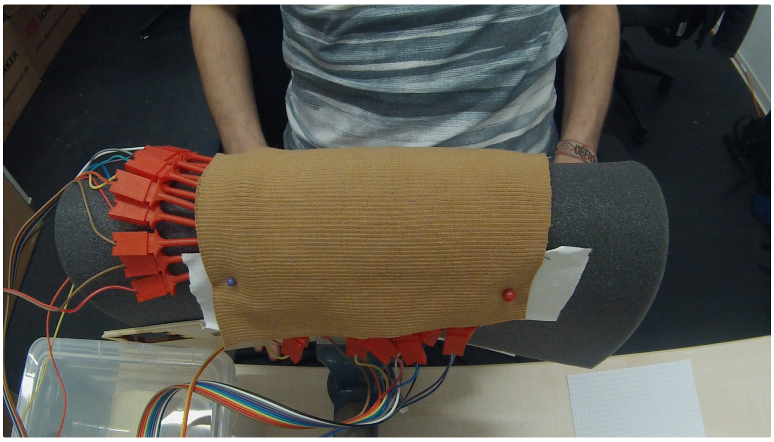

Figure 4.3: Condition: 53mm, rib knit fabric, foam, hemmed in a vise

4.3 Study Procedure

After the user arrived we introduced our prototype. We explained the basic functionality and demonstrated how the output looks like. Then we let the user test the eight gestures and some freestyle strokes. This was done without foam or any additional fabric. We pointed out that a certain amount of pressure is essential for our prototype to

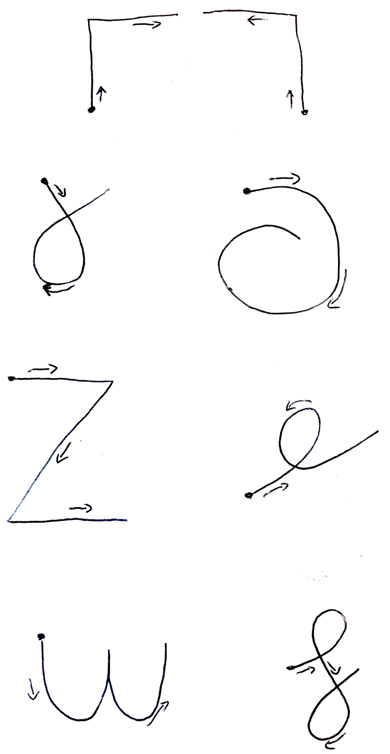

Figure 4.4: Gesture set: right angle, left angle, slope, spiral, z, pigtail, w, doubleslope

recognize the touch. When they felt familiar enough, about 2 minutes of testing, we prepared the first condition.

For each condition we setup a GoPro Hero 3 to capture the prototype and the acting hand of the user. When we were ready to start recording the screen and setup, we told the user to continue. Since the user cannot see the output during the study, we told the user when insufficient pressure was applied or when the touch-sensing area was left. In both cases we most likely recognized one or two wrong gestures. We represent the number of wrong gestures with an *x* in the respective chart.

When one condition is completed we asked the user about their impressions of the fabric, softness, and curvature.

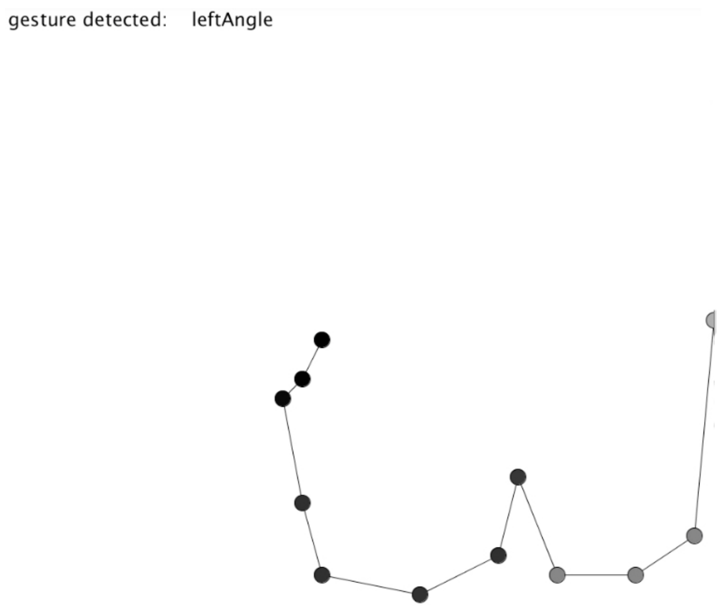

Figure 4.5: The characteristics of w are there, but nonetheless *leftAngle* was detected.

4.4 Observation

Distinguish between hardware and gesture recognition

The results proof the general applicability of our prototype. The overall success rate of performed gesture is shown in figure 4.6. We distinguish between the hardware results by eye, with recognition, and with repetition if the user left the touch sensing area. 84.5% of all gestures generated recognizable data. Meaning that by eye the output of the data matches the current gesture. However only 75.5% of these gestures were recognized correctly using the 1$ recognizer. One example of a false negative is shown in figure 4.5. We made this separation, since we are primarily interested in the capabilities of our hardware prototype.
Additionally, we let the participants repeat those trials, where they left the touch sensing area. This leads to an average success rate of 87.5% and the second user even achieved 91%.
The difference of hardware success rate and recognizer success rate is almost the same for all conditions. Since we are interested in the performance of the hardware we only consider the success rate of the hardware from now on.

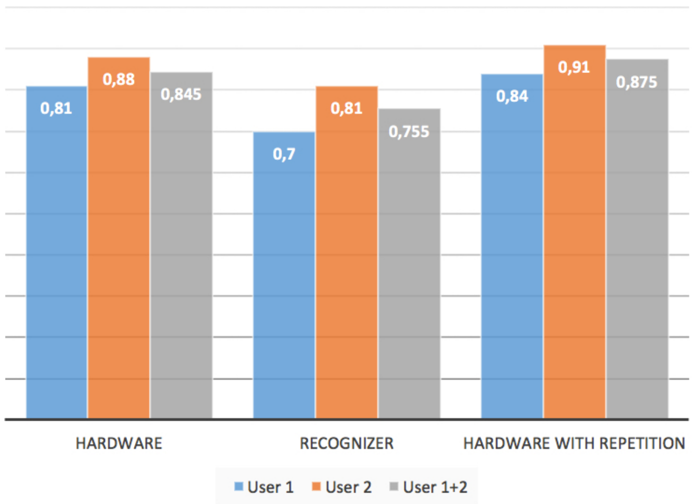

Figure 4.6: Success rate of all performed gestures with different criteria.

When it comes to surface curvature we got the results we expected shown in figure 4.7. The curvature with a 53mm diameter performs worst but still with a success rate of 75.5%. The best curvature is no curvature at all. On the table both users achieved a success rate above 90% with an average of 92%. It is notable that the user with experience obtained a rate of 95% with 66mm curvature where the other user got 79%. Nevertheless, the inexperienced user outperformed the other user on the flat surface.

Flat surface is best and 53mm worst

When we asked the participants which curvature they prefer they agree that the flat surface is most pleasant for touch input and the more curvature, the more likely it happens that they slip off the surface. This leads to unintentional input and thus to more input error. The success rates with different softness are shown in figure 4.8. Our hypothesis that softness has a bad influence on the performance of our prototype was falsified. One user obtained 81% in both cases and the more experienced user performed better on the foam (92%). The participants reported that is was much more pleasant to perform the gestures on the foam due to the distribution of the pressure.

Only slight improvement with foam

Figure 4.7: Success rates on surface curvatures

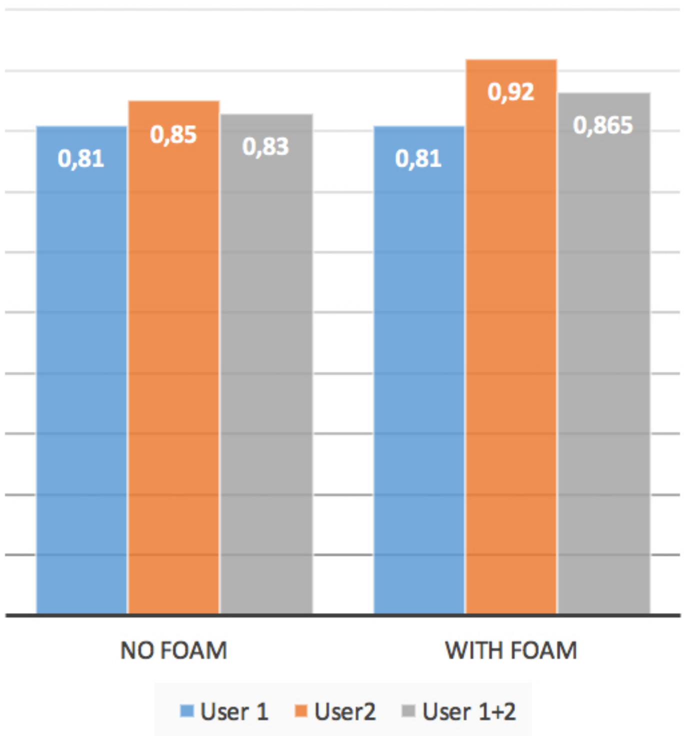

Figure 4.8: Success rates with and without foam

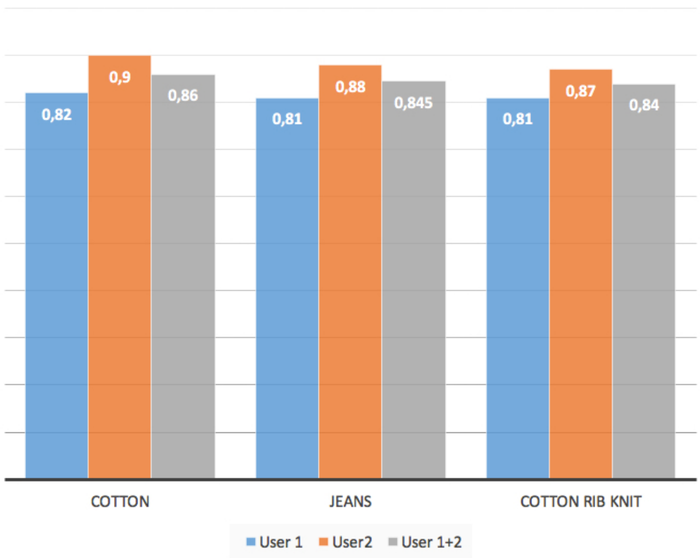

Figure 4.9: Success rates with different materials

The different materials seem to have no influence on the performance of our prototype as shown in figure 4.9. The average success rate is between 84% and 86%. However, the participants reported that the rib knit fabric is extremely annoying due to the immense flexibility. One user said he likes jeans for getting good results but after a while the abrasive surface of the jeans leads to tingle and makes it unpleasant. Both participant prefer cotton and jeans because of their stiffness resulting in less folds. Although the participants reported occasional wrinkles of the rib knit fabric and therefore perceived lose of contact, the sensor still recognized the input as fine as the other fabrics.

The success rates of each gesture are shown in figure 4.10. The most complex gesture was *doubleslope* and was the

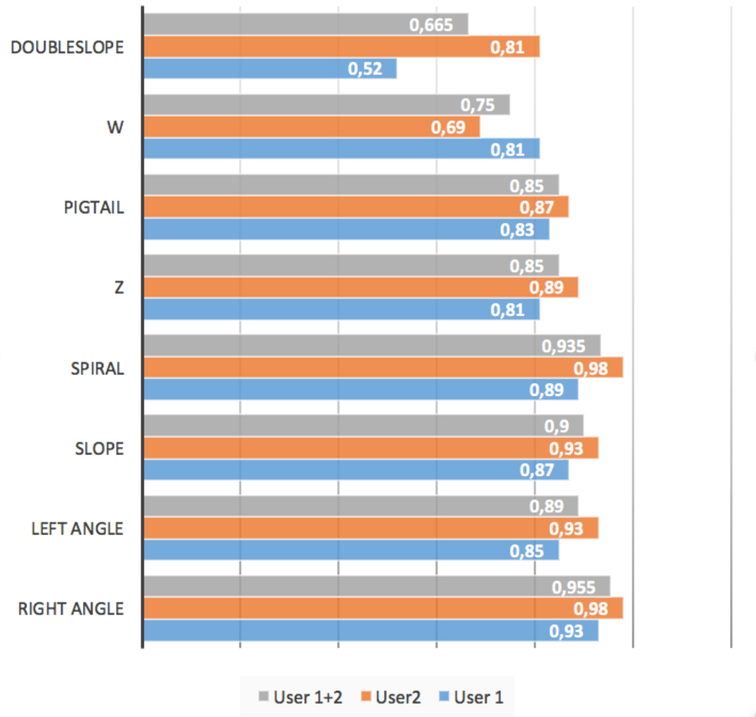

Figure 4.10: Success rates for each gesture

hardest gesture to perform with an average success rate of 66.5%. However the difference between the users is huge (81% and 52%). Applying the required amount of pressure steadily is more difficult when the gesture key characteristics are complex. The *doubleslope* gesture requires more changes of direction than *pigtail* (average 85%).
There is notable difference between *left angle* (89%) and *right angle*, which has the best average success rate of 95.5%. It remains to test if this is ascribed to the dominant hand. Except for the *w* gesture (75%), the rest of the gestures are within 85% and 93.5%.

4.5 Conclusion

The surface curvature has the most meaningful effect on the performance of the prototype next to the characteristics of a gesture. There is a consistent difference between the two participants due to the varying level of experience. This indicates a learning effect which also was the subjective estimation of both participants. Primarily the required pressure to generate a contact is remembered over time.

Chapter 5

Summary and Future Work

5.1 Summary and Contributions

We presented a 2D textile touchpad for eyes free interaction capable of detecting up to eight free-form gestures with 84.5% reliability. The touchpad is composed of textile materials that are flexible, lightweight, breathable, and washable. It is composed of two layers of fabric with conductive yarn sandwiching a space made of 3mm foam. The touchpad is based on the simple principles of resistive touch technology which has some advantages over capacitive technology when it comes to wearable touch sensing, namely neglecting the noise of the human body. Most of them, which use capacitive touch and the gestures, they are able to detect, are rather limited due to that noise. Our prototype only yields a touch if the layers are physically connected. The limitations of our approach are low resolution and the need of a certain amount of force to connect the conductive layers. We evaluated the robustness of the textile touchpad as a wearable sensor by testing gesture recognition rate and noise generated under three extreme conditions: softness, curvature, and friction. We found that, despite the low resolution of 14 by 14, we are able to detect more complex unistroke gestures.

We described how we built our prototype with low cost materials. We explained step by step how to attach the pinstripe fabric layers to the spacing material, such that everyone is able to rebuild it in short time. The necessary code is linked in the 6ppendix. Our prototype is easy scalable and is only limited in the number of pins of the used microcontroller. Although we made our prototypes equilateral, it is simply possible to give it any rectangular size.
Furthermore we explained the software for our sensor to detect simple unistroke mark-based gestures using our own recognizer. The sensor is able to recognize more complex unistroke free-form gestures using the 1$ recognizer. This is the first full textile touchpad being able to do that consistently. We tested the prototype under multiple conditions to evaluate the physical limitations of the sensor. We implemented the recognition with an threshold of 3 milliseconds such that the impact of contact loss is lower. Such a contact loss is caused by too few pressure or slipping of the prototypes touch sensing area due to the curvature. We found that there is a learning effect since we observed better results from the more experienced user.

ToDo: figure of curvature

Beside that we found that the 1$ recognizer is not optimal for sensors with a rather low resolution of 14 by 14. Curvature seems to have the only significant impact on overall performance which makes the thigh best suited additional to the fact that both participants prefer jeans fabric for interacting with the sensor. Although both participants liked operating the sensor, both agree that it gets unpleasant over time and it would be rather suited for occasional use. This is the result of the amount of force needed for a touch and the resulting strain in the operating finger. This varies for different body densities. Effects of abrupt changes in density (e.g. bone $\rightarrow$ muscle) were not yet investigated.

5.2 Future work

Since the results of the hardware evaluation have shown that our technique of building a 2D textile touchpad is promising, the most immediate step would be making the sensor actually wearable by integrating it into everyday clothing. This yields new challenges beside recognizing 2D touch. The wiring of sensor and microcontroller and the power supply should be imperceptible, lightweight, and compact. Furthermore the data processing and gesture recognition should be ported to the microcontroller.

With a longitudinal evaluation of the system by wearing it for a period of 5 days we will be able to evaluate the wearability of the system and get insight in which conditions the noise increase and gesture recognition rate decrease.

Also increasing the resolution without making the surface larger could increase the reliability of gesture recognition. To increase the resolution we will try to decrease the spacing between the conductive lines and the width of the conductive threads. The lack of feedback is another issue for wearable devices. Prattichizzo et al. [2013] investigated this problem by providing haptic feedback. Getting feedback when acting close to the edge of the sensor area could greatly improve the performance.

Finally a number of embedded prototypes would be built with a larger range of fabrics used in today's clothes. Furthermore we will test different spacing techniques to improve wearability and decrease the required pressure.

Chapter 6

APPENDIX

Source code and files of evaluation[a]

[a] http://hci.rwth-aachen.de/texitouch

Bibliography

Lisa Anthony and Jacob O Wobbrock. $n-protractor: a fast and accurate multistroke recognizer. In *Proceedings of Graphics Interface 2012*, pages 117–120. Canadian Information Processing Society, 2012.

Andrew Bragdon, Eugene Nelson, Yang Li, and Ken Hinckley. Experimental analysis of touch-screen gesture designs in mobile environments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 403–412, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0228-9. doi: 10.1145/1978942.1979000. URL `http://doi.acm.org/10.1145/1978942.1979000`.

Stephen Brewster, Joanna Lumsden, Marek Bell, Malcolm Hall, and Stuart Tasker. Multimodal 'eyes-free' interaction techniques for wearable devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '03, pages 473–480, New York, NY, USA, 2003. ACM. ISBN 1-58113-630-7. doi: 10.1145/642611.642694. URL `http://doi.acm.org/10.1145/642611.642694`.

Chat Wacharamanotham Jan Thar Jan Borchers Hamdan et al., Florian Heller. Grabrics: A foldable two-dimensional textile input controller. In *Extended Abstracts of 2016 ACM SIGCHI Conference on Human Factors in Computing*, To appear CHI '16 EA, 2016.

Florian Heller, Stefan Ivanov, Chat Wacharamanotham, and Jan Borchers. Fabritouch: Exploring flexible touch input on textiles. In *Proceedings of the 2014 ACM International Symposium on Wearable Computers*, ISWC '14, pages 59–62, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2969-9. doi: 10.1145/2634317.2634345. URL `http://doi.acm.org/10.1145/2634317.2634345`.

Paul Holleis, Albrecht Schmidt, Susanna Paasovaara, Arto Puikkonen, and Jonna Häkkilä. Evaluating capacitive touch input on clothes. In *Proceedings of the 10th International Conference on Human Computer Interaction with Mobile Devices and Services*, MobileHCI '08, pages 81–90, New York, NY, USA, 2008. ACM. ISBN 978-1-59593-952-4. doi: 10.1145/1409240.1409250. URL `http://doi.acm.org/10.1145/1409240.1409250`.

Thorsten Karrer, Moritz Wittenhagen, Florian Heller, and Jan Borchers. Pinstripe: Eyes-free continuous input anywhere on interactive clothing. In *Adjunct Proceedings of the 23Nd Annual ACM Symposium on User Interface Software and Technology*, UIST '10, pages 429–430, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0462-7. doi: 10.1145/1866218.1866255. URL `http://doi.acm.org/10.1145/1866218.1866255`.

E Rehmi Post and Margaret Orth. Smart fabric, or. In *iswc*, page 167. IEEE, 1997.

D. Prattichizzo, F. Chinello, C. Pacchierotti, and M. Malvezzi. Towards wearability in fingertip haptics: A 3-dof wearable device for cutaneous force feedback. *IEEE Transactions on Haptics*, 6(4):506–516, Oct 2013. ISSN 1939-1412. doi: 10.1109/TOH.2013.53.

J. Rantanen, J. Impiö, T. Karinsalo, M. Malmivaara, A. Reho, M. Tasanen, and J. Vanhala. Smart clothing prototype for the arctic environment. *Personal Ubiquitous Comput.*, 6(1):3–16, January 2002. ISSN 1617-4909. doi: 10.1007/s007790200001. URL `http://dx.doi.org/10.1007/s007790200001`.

J. Rekimoto. Gesturewrist and gesturepad: unobtrusive wearable interaction devices. In *Wearable Computers, 2001. Proceedings. Fifth International Symposium on*, pages 21–27, 2001. doi: 10.1109/ISWC.2001.962092.

T. Scott Saponas, Chris Harrison, and Hrvoje Benko. Pockettouch: Through-fabric capacitive touch input. In *Proceedings of the 24th Annual ACM Symposium on User In-*

terface Software and Technology, UIST '11, pages 303–308, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0716-1. doi: 10.1145/2047196.2047235. URL `http://doi.acm.org/10.1145/2047196.2047235`.

Christian Schmidt. Texipad software, 2016. URL `http://hci.rwth-aachen.de/TexiTouch`.

Boris Smus and Mark D. Gross. Ubiquitous drums: A tangible, wearable musical interface. In *CHI '10 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '10, pages 4009–4014, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-930-5. doi: 10.1145/1753846.1754094. URL `http://doi.acm.org/10.1145/1753846.1754094`.

Jessica Speir, Rufino R. Ansara, Colin Killby, Emily Walpole, and Audrey Girouard. Wearable remote control of a mobile device: Comparing one- and two-handed interaction. In *Proceedings of the 16th International Conference on Human-computer Interaction with Mobile Devices & Services*, MobileHCI '14, pages 489–494, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-3004-6. doi: 10.1145/2628363.2634221. URL `http://doi.acm.org/10.1145/2628363.2634221`.

Raphael Wimmer and Patrick Baudisch. Modular and deformable touch-sensitive surfaces based on time domain reflectometry. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, UIST '11, pages 517–526, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0716-1. doi: 10.1145/2047196.2047264. URL `http://doi.acm.org/10.1145/2047196.2047264`.

Jacob O Wobbrock, Andrew D Wilson, and Yang Li. Gestures without libraries, toolkits or training: a $1 recognizer for user interface prototypes. In *Proceedings of the 20th annual ACM symposium on User interface software and technology*, pages 159–168. ACM, 2007.

Index

Typeset March 30, 2016