3D FROM 2D TOUCH

CHRISTIAN HOLZ

HUMAN-COMPUTER INTERACTION
HASSO PLATTNER INSTITUTE, UNIVERSITY OF POTSDAM
POTSDAM, GERMANY

DISSERTATION
ZUR ERLANGUNG DES GRADES EINES
DOKTORS DER NATURWISSENSCHAFTEN
– DR. RER. NAT. –

JUNE 2013

ABSTRACT

While interaction with computers used to be dominated by mice and keyboards, new types of sensors now allow users to interact through touch, speech, or using their whole body in 3D space. These new interaction modalities are often referred to as "natural user interfaces" or "NUIs." While 2D NUIs have experienced major success on billions of mobile touch devices sold, 3D NUI systems have so far been unable to deliver a mobile form factor, mainly due to their use of cameras. The fact that cameras require a certain *distance* from the capture volume has prevented 3D NUI systems from reaching the *flat* form factor mobile users expect.

In this dissertation, we address this issue by sensing 3D input using flat 2D sensors. The systems we present observe the input from 3D objects as 2D *imprints* upon physical contact. By sampling these imprints at very high resolutions, we obtain the objects' *textures*. In some cases, a texture uniquely identifies a biometric feature, such as the user's fingerprint. In other cases, an imprint stems from the user's clothing, such as when walking on multitouch floors. By analyzing from which *part* of the 3D object the 2D imprint results, we *reconstruct* the object's pose in 3D space.

While our main contribution is a general approach to sensing 3D input on 2D sensors upon physical contact, we also demonstrate three applications of our approach.

(1) We present high-accuracy touch devices that allow users to *reliably* touch targets that are a third of the size of those on current touch devices. We show that different users and 3D finger poses *systematically* affect touch sensing, which current devices perceive as random input noise. We introduce a model for touch that compensates for this systematic effect by deriving the 3D finger pose and the user's identity from each touch imprint. We then investigate this systematic effect in detail and explore how users *conceptually* touch targets. Our findings indicate that users aim by aligning *visual* features of their fingers with the target. We present a visual model for touch input that eliminates virtually all systematic effects on touch accuracy.

(2) From each touch, we identify users biometrically by analyzing their fingerprints. Our prototype Fiberio integrates fingerprint scanning and a display into the *same* flat surface, solving a long-standing problem in human-computer interaction: secure authentication on touch*screens*. Sensing 3D input and authenticating users upon touch allows Fiberio to implement a variety of applications that traditionally require the bulky setups of current 3D NUI systems.

(3) To demonstrate the versatility of 3D reconstruction on larger touch surfaces, we present a high-resolution pressure-sensitive floor that resolves the texture of objects upon touch. Using the same principles as before, our system GravitySpace analyzes all imprints and identifies users based on their shoe soles, detects furniture, and enables accurate touch input using feet. By classifying all imprints, GravitySpace detects the users' body parts that are in contact with the floor and then reconstructs their 3D body poses using inverse kinematics. GravitySpace thus enables a range of applications for future 3D NUI systems based on a flat sensor, such as smart rooms in future homes.

We conclude this dissertation by projecting into the future of mobile devices. Focusing on the mobility aspect of our work, we explore how NUI devices may one day augment users directly in the form of implanted devices.

ZUSAMMENFASSUNG

Die Interaktion mit Computern war in den letzten vierzig Jahren stark von Tastatur und Maus geprägt. Neue Arten von Sensoren ermöglichen Computern nun, Eingaben durch Berührungs-, Sprach- oder 3D-Gestensensoren zu erkennen. Solch neuartige Formen der Interaktion werden häufig unter dem Begriff „natürliche Benutzungsschnittstellen“ bzw. „NUIs“ (englisch *natural user interfaces*) zusammengefasst. 2D-NUIs ist vor allem auf Mobilgeräten ein Durchbruch gelungen; über eine Milliarde solcher Geräte lassen sich durch Berührungseingaben bedienen. 3D-NUIs haben sich jedoch bisher nicht auf mobilen Plattformen durchsetzen können, da sie Nutzereingaben vorrangig mit Kameras aufzeichnen. Da Kameras Bilder jedoch erst ab einem gewissen Abstand auflösen können, eignen sie sich nicht als Sensor in einer *mobilen* Plattform.

In dieser Arbeit lösen wir dieses Problem mit Hilfe von 2D-Sensoren, von deren Eingaben wir 3D-Informationen *rekonstruieren*. Unsere Prototypen zeichnen dabei die 2D-*Abdrücke* der Objekte, die den Sensor berühren, mit hoher Auflösung auf. Aus diesen Abdrücken leiten sie dann die *Textur* der Objekte ab. Anhand der *Stelle* der Objektoberfläche, die den Sensor berührt, rekonstruieren unsere Prototypen schließlich die 3D-Ausrichtung des jeweiligen Objektes.

Neben unserem Hauptbeitrag der 3D-Rekonstruktion stellen wir drei Anwendungen unserer Methode vor.

(1) Wir präsentieren Geräte, die Berührungseingaben dreimal genauer als existierende Geräte messen und damit Nutzern ermöglichen, dreimal kleinere Ziele *zuverlässig* mit dem Finger auszuwählen. Wir zeigen dabei, dass sowohl die Haltung des Fingers als auch der Benutzer selbst einen *systematischen* Einfluss auf die vom Sensor gemessene Position ausübt. Da existierende Geräte weder die Haltung des Fingers noch den Benutzer erkennen, nehmen sie solche Variationen als Eingabeungenauigkeit wahr. Wir stellen ein Modell für Berührungseingabe vor, das diese beiden Faktoren integriert, um damit die gemessenen Eingabepositionen zu präzisieren. Anschließend untersuchen wir, welches *mentale* Modell Nutzer beim Berühren kleiner Ziele mit dem Finger anwenden. Unsere Ergebnisse deuten auf ein *visuelles* Modell hin, demzufolge Benutzer Merkmale auf der Oberfläche ihres Fingers an einem Ziel ausrichten. Bei der Analyse von Berührungseingaben mit diesem Modell verschwinden nahezu alle zuvor von uns beobachteten systematischen Effekte.

(2) Unsere Prototypen identifizieren Nutzer anhand der biometrischen Merkmale von Fingerabdrücken. Unser Prototyp Fiberio integriert dabei einen Fingerabdruckscanner und einen Bildschirm in die *selbe* Oberfläche und löst somit das seit Langem bestehende

Problem der sicheren Authentifizierung auf Berührungs*bildschirmen*. Gemeinsam mit der 3D-Rekonstruktion von Eingaben ermöglicht diese Fähigkeit Fiberio, eine Reihe von Anwendungen zu implementieren, die bisher den sperrigen Aufbau aktueller 3D-NUI-Systeme voraussetzten.

(3) Um die Flexibilität unserer Methode zu zeigen, implementieren wir sie auf einem großen, berührungsempfindlichen Fußboden, der Objekttexturen bei der Eingabe ebenfalls mit hoher Auflösung aufzeichnet. Ähnlich wie zuvor analysiert unser System GravitySpace diese Abdrücke, um Nutzer anhand ihrer Schuhsolen zu identifizieren, Möbelstücke auf dem Boden zu erkennen und Nutzern präzise Eingaben mittels ihrer Schuhe zu ermöglichen. Indem GravitySpace alle Abdrücke klassifiziert, erkennt das System die Körperteile der Benutzer, die sich in Kontakt mit dem Boden befinden. Aus der Anordnung dieser Kontakte schließt GravitySpace dann auf die Körperhaltungen aller Benutzer in 3D. GravitySpace hat daher das Potenzial, Anwendungen für zukünftige 3D-NUI-Systeme auf einer flachen Oberfläche zu implementieren, wie zum Beispiel in zukünftigen intelligenten Wohnungen.

Wie schließen diese Arbeit mit einem Ausblick auf zukünftige interaktive Geräte. Dabei konzentrieren wir uns auf den Mobilitätsaspekt aktueller Entwicklungen und beleuchten, wie zukünftige mobile NUI-Geräte Nutzer in Form implantierter Geräte direkt unterstützen können.

PUBLICATIONS

Some ideas and figures have appeared previously in the following publications:

Christian Holz and Patrick Baudisch. Fiberio: A Touchscreen That Senses Fingerprints. In *Proceedings of UIST 2013*, 10 pages.

Alan Braenzel, Christian Holz, Daniel Hoffmann, Dominik Schmidt, Marius Knaust, Patrick Luehne, Rene Meusel, Stephan Richter, and Patrick Baudisch. Gravityspace: Tracking Users and Their Poses in a Smart Room Using a Pressure-Sensing Floor. In *Proceedings of CHI 2013*, pages 725–734.

Christian Holz, Tovi Grossman, George Fitzmaurice, and Anne Agur. Implanted User Interfaces. In *Proceedings of CHI 2012*, pages 503–512.

Stephan Richter, Christian Holz, and Patrick Baudisch. Bootstrapper: Recognizing Tabletop Users by Their Shoes. In *Proceedings of CHI 2012*, pages 1249–1252.

Sean Gustafson, Christian Holz, and Patrick Baudisch. Imaginary Phone: Learning Imaginary Interfaces by Transferring Spatial Memory from a Familiar Device. In *Proceedings of UIST 2011*, pages 283–292.

Christian Holz and Patrick Baudisch. Understanding Touch. In *Proceedings of CHI 2011*, pages 2501–2510.

Thomas Augsten, Konstantin Kaefer, Rene Meusel, Caroline Fetzer, Dorian Kanitz, Thomas Stoff, Torsten Becker, Christian Holz, and Patrick Baudisch. Multitoe: High-Precision Interaction With Back-Projected Floors Based on High-Resolution Multi-Touch Input. In *Proceedings of UIST 2010*, pages 209–218.

Christian Holz and Patrick Baudisch. The Generalized Perceived Input Point Model and How to Double Touch Accuracy by Extracting Fingerprints. In *Proceedings of CHI 2010*, pages 581–590.

ACKNOWLEDGMENTS

This work would not have been possible without the countless discussions, opinions, and inspiration I have received from a large number of great people throughout the years. Their advice has been invaluable for me and the list below is likely not exhaustive.

First of all, I want to thank my advisor Patrick Baudisch for this experience. I was very lucky to join your lab early on and I have learned so much from you as you created a group of great people and an environment that leaves nothing to be desired as we all pursued exciting research directions. I am thankful for all your support and advice throughout the course of my dissertation, for constantly challenging me to keep asking despite having found answers, for asking the questions that I missed, and for having answers when I ran out of them. Thank you so much!

I am also deeply thankful for the opportunity to work with a few people outside school. Andy Wilson was a wonderful mentor during the summer I was working with him—what a fabulous experience. I had a lot of fun during the many chats about wacky ideas, British humor, and font hinting on the patio. Thank you! Another summer, working with Tovi Grossman and George Fitzmaurice was a rad experience and I am sincerely thankful for their great mentorship and the constant support as we together explored adventurous terrain. Thanks Tovi for the many conversations, your lessons on balance and time management, and all the fun activities (including the Muay Thai events and showing me the best sandwich place ever). Thanks George for the great guidance, your spot-on advice, and the chats about exquisite food in Toronto.

I would like to express my gratitude to Steven Feiner and Alan Borning. It was very helpful to know that I could always turn to you for advice. I would also like to thank Albrecht Schmidt and Ken Perlin for many inspiring discussions throughout the years, for joining my committee, and for their feedback on this dissertation.

I am greatly indebted to all my collaborators for all the cool projects we worked on together and all the input they gave me. Thank you Sean Gustafson for all the feedback throughout the past four years (and for sharing your keen observations), Henning Pohl, Anne Agur, Stephan Richter, Dominik Schmidt, and Christian Steins. Thanks to all the bachelor students who worked on Multitoe and GravitySpace. Both projects were the result of an amazing group effort over the past four years, which allowed me not only to apply my research on a larger scale, but also to learn so much from you and from co-advising these two projects. Thank you Patrick Lühne (also for your thorough feedback on this dissertation), Marius Knaust, Daniel Hoffmann, Martin Fritzsche,

Rene Meusel, Torsten Becker, Alan Bränzel, Caroline Fetzer, Dorian Kanitz, Konstantin Käfer, Thomas Augsten, and Thomas Stoff.

I would like to acknowledge Raimund Mückstein, Andreas Steinbacher, Raphael Wimmer, Frank Li, Christoph Sterz, Toni Mattis, Sven Köhler for all the input and feedback, past and present members of the lab Frederik Rudeck, Anne Roudaut, Christian Loclair, Liwei Chan, Stefanie Müller, Pedro Lopes, Michael Karsch, and Lung-Pan Cheng for many brainstormings, their comments and the good times, as well as Hrvoje Benko, Ken Hinckley, Justin Matejka, Alex Tessier, and Gord Kurtenbach for inspiring conversations, feedback, and fun events during my internships. Thank you very much Azam Khan; your help was instrumental to realize our project "in no time."

I want to thank all the participants of my user studies for their patience. As I kept reassuring you during the trials, repeatedly touching the crosshairs—though arguably a rather "dense" task—did indeed lead to many new insights and you have contributed to something larger (and not just satisfied your craving of chocolate). Thanks!

Finally, I owe much to the companies that generously supported this work. Thank you CrossMatch for providing us with a Guardian Fingerprint Scanner in 2009, which inspired a whole line of research. Thanks also to Loptek for sharing their insights on glass fibers, as well as Incom and Schott for providing us with fiber optic plates.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

Even though screens that sense direct touch input were invented almost fifty years ago [76], the subsequent four decades of personal computing have been characterized by keyboards and mice. The invention of the computer mouse [39] and the introduction of the graphical desktop only a few years later made computers usable for everybody. Even today, the mouse is still the primary means by which people interact with their desktop computers. This is because the entire user interface of such systems is tailored to the use of mouse and keyboard.

While the mouse and keyboard have remained the dominant input controls without major changes in the type of data they capture, computer output has become considerably richer. The resolution of computer screens has increased constantly, powerful graphics cards now render high-fidelity images in real-time, and high-quality audio output further adds to an immersive user experience. The bandwidth of information that computers now emit is immense, which allows users to consume high-quality content in rich detail.

In contrast, mouse and keyboard continue to provide information at a limited bandwidth. The mouse records relative spatial movements in two dimensions and the keyboard senses a combination of button presses. While both input components sense input with high certainty, the input bandwidth they offer is comparably low. This has substantially limited the amount of information computers are able to perceive about the user.

However, recent advances in technology have triggered a change in this asymmetry in human-computer interaction. Computers have started to fuse large amounts of information that they obtain from a wide variety of sensors in order to derive input from the user, thereby considerably extending the depth of data they obtain. Such sensors include high-resolution color cameras, 3D depth cameras, and microphones, all of which provide data about the world around computers at high frame rates. At the same time, computers have become powerful enough to process this information in real-time, enabling them to "see" and "hear" users at the same level of detail as users perceive them.

Following these advances in sensing technology, new interaction modalities have become possible. Commonly referred to as "natural user interfaces" or "NUIs" [166], they encompass modalities that are inspired by users' "natural" (inter)actions, such as touch, gestures, or speech.

1.1 NATURAL USER INTERFACES: COMPUTERS NOW SENSE RICH INPUT

Systems that implement natural user interfaces restore the balance between the bandwidth of input and output. In NUIs, information passes at high bandwidth in both directions and computers *constantly* capture large amounts of input data from a wealth of sensors. While this approach to sensing input is holistic, systems need to *interpret* all this data for meaningful input. This comes at the cost of obtaining less certain input commands compared to the discrete events from keyboard and mouse.

Touch input is an example of a very successful 2D natural user interface. As shown in Figure 1.1a, by using touch, users *directly* interact with content on the screen. This integration of input and output into the same space has been adopted by over a billion devices, often because it allows devices to assume *mobile* form factors. Because touchscreens reduce the overall size of devices by replacing the controls around the display, they are frequently used in small devices, such as phones, music players, and watches. On such devices, touch input is essentially a pointing method for spatial input—much like the mouse is for desktop computers. While multitouch considerably expanded the interaction vocabulary, such as with gesture input, multiple touch contacts are mostly still interpreted in terms of several pointing utilities.

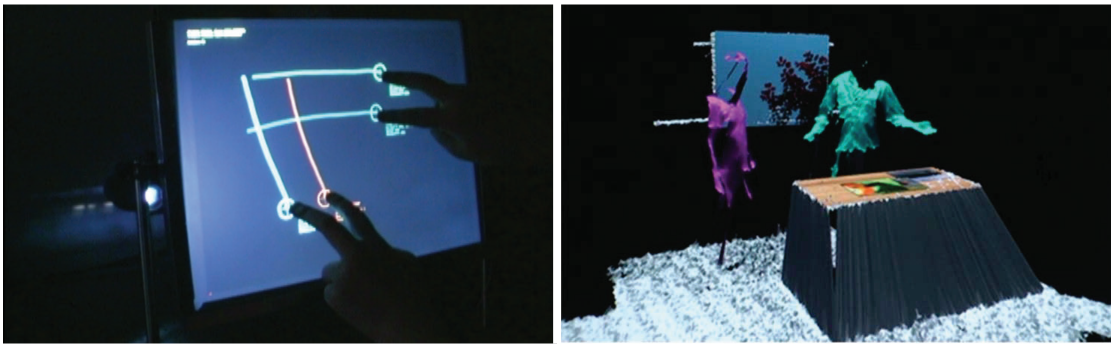


Figure 1.1: Natural user interfaces ("NUIs") are replacing traditional input controls, such as mice and keyboards. Left: Touch input a very successful example of 2D NUIs has enabled users to directly interact with the displayed content. Right: 3D NUIs allow users to interact through gestures or using their whole body. (Left image by Han [58], right image from LightSpace [170])

With the advent of depth cameras, interaction with computers has become even more expressive. Depth cameras sense users, their environment, and their interactions in 3D. Whereas 2D touch provides discrete input events, other information becomes important in 3D scenes, such as the user's shape including the body pose, how they move, and who they are. As shown in Figure 1.1 (right), such systems allow the user to interact with their whole body in the space around computers [170, 67]. While 3D interaction has a long tradition in research [63, 59], it has become mainstream with the advent of commoditized 3D sensors and has entered users' homes, for such uses as playing video games [84].

However, the ability to sense users in 3D is commonly limited to *stationary* installations. Figure 1.2a shows a typical example setup of cameras that are mounted to the ceiling of a room (here LightSpace [170]). This setup affords sensing 3D data from above and ensures a good coverage of the entire space. Fusing the data from all cameras produces a comprehensive 3D snapshot of the scene, including moving users as well as static objects.

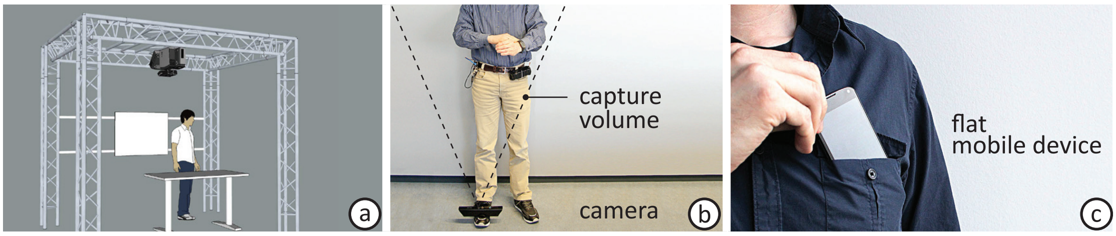


Figure 1.2: (a) Systems that sense users in 3D originated in stationary installations. Mounting cameras to the ceiling ensures a good coverage of the 3D space, including the geometry of users as well as objects inside the room. (b) Since the camera needs to be placed away from the capture volume, this setup requires space, which impedes mobility. (c) To achieve mobility, devices need to adopt *flat* form factors, allowing them to be portable and achieve mass adoption. (Image a from LightSpace [170], b adapted from ShoeSense [12])

In contrast to stationary scenarios, interaction with devices today increasingly involves *mobile* use. This makes the paradigms of natural user interfaces desirable in mobile scenarios as well. Unfortunately, camera-based sensing faces a considerable challenge in mobile scenarios as shown in Figure 1.2b. To capture 3D data using a camera, the camera needs to observe objects of interest from a *distance*. The required distance, however, makes setups involving cameras space-consuming and thus immobile.

The property that has allowed current devices to become mobile and achieve mass adoption in the first place is their form factor: Because mobile devices are *flat*, users can easily carry them around, put them in their pockets, and retrieve them for mobile use as shown in Figure 1.2c.

Therefore, to allow mass available devices to implement 3D natural user interfaces, mobile devices must be enabled to sense the types of data that have traditionally been available only in stationary systems.

In this dissertation, we address this issue by sensing 3D input using a flat 2D sensor, which provides mobile devices with the ability to implement 3D NUIs. While such devices typically sense input in the form of 2D touch contacts, our approach allows devices to *reconstruct* 3D information from the input they observe.

1.2 3D FROM 2D TOUCH: 3D NATURAL USER INTERFACES ON FLAT DEVICES

Our approach to obtaining the type of information required for natural user interfaces is to augment touch sensing, which is the dominant input sensor on current mobile devices. We thereby advance touch input from a 2D pointing modality to a 3D input method, which provides more information upon input. We propose touch devices that capture the 2D *imprints* that all objects leave on the sensor during physical contact. By using touch-surface materials that reveal relevant features and by increasing the resolution of the sensor itself, our devices capture all imprints with 10–100 times higher detail than traditional devices.

By observing touch contacts at that level of detail, our touch devices sense not only the location of touch contacts, but they also resolve the *texture* of the objects that are in contact. Analyzing this texture allows our prototypes to extract additional information from all touches on the surface.

In the case of touch devices that are operated using fingers, the texture of a touch contact represents the user's fingerprint. Since a fingerprint is a biometric feature, we use it to uniquely identify the finger and thus the user who has touched the device. In other cases, touch imprints stem from other objects, such as on multitouch floors where imprints result from users' clothing or the footprint of furniture. Because the texture of such objects is not unique, the imprint we observe offers less discriminative power, but still allows classifying the object.

We finally obtain each object's pose in 3D space. By analyzing from which *part* of the 3D object the 2D imprint results, we reconstruct the 3D orientation of the object that must have caused the particular imprint.

Our approach to sensing 3D input relies on physical contact between 3D objects and 2D touch surfaces. While this limits the extent to which we can sense 3D information above the surface, physical contact happens as part of regular input on touch devices, which enables our approach. On much larger scales, such as in the case of multitouch floors, we use per-pixel pressure sensing that we incorporate into the floor to observe touch contacts. Here, physical contact is facilitated by gravity, which causes all people and objects to leave imprints on the floor.

Our general approach to reconstructing 3D information from 2D touch contacts enables 3D input on flat 2D sensors. We demonstrate three applications of our approach:

1. High-accuracy touch input: We present devices that make touch input highly accurate for users. Compared to existing touch devices, our prototypes infer touch with three times higher accuracy. They achieve this by calibrating touch input on a per-user and a per-finger-pose level. We show that current devices are subject to sensing a *systematic input error*, because they are oblivious to the user's identity and the orientation of the user's finger upon touch. Our devices compensate for this error by reconstructing 3D input from all 2D touch contacts, which allows them to sense touch accurately.

In a series of user studies, we examine this systematic effect in detail. We investigate how participants "conceptually" acquire small targets using touch input. From our results, we derive a new model for touch input that predicts input locations based on *visual* features. This is a departure from current devices, which measure the contact area between the user's finger and the surface to obtain input locations. In a final evaluation, we show that our new model is unaffected by the systematic error that current devices observe and that it requires no user-specific or finger-pose-specific calibration to infer touch accurately. This suggests that our model is a good approximation of how users conceptualize touch input.

2. Biometric user identification on touchscreens: As part of our approach, our touch prototypes implicitly identify users during touch based on the biometric features of their fingerprints. Unlike existing devices, our prototype Fiberio accomplishes this on a touch*screen*, thereby integrating fingerprint scanning and displaying output into the same surface. In addition, Fiberio also reconstructs the user's 3D finger pose from all 2D touch contacts. Knowledge of the user's identity and the 3D pose enables a range of new applications, such as user interface personalization and sensing high-degree-of freedom input—applications that have traditionally only been achieved using the bulky setups of current natural user interaction systems.

3. Reconstructing 3D input on much touch larger surfaces: We generalize our concepts of reconstructing 3D information from 2D touch contacts to larger scales. By implementing these techniques on a large floor that senses touch in the form of per-pixel pressure, we demonstrate how to identify users, detect passive objects, such as furniture, and reconstruct users' 3D poses in a smart room solely based on the touch contacts all objects leave on the floor. To accomplish this, we again analyze the texture of all touch contacts. In the case of the floor, the texture represents users' shoe soles, the structure of their clothing, and the pattern of imprints that passive objects cause when in contact with the floor. Since the user's body leaves multiple contacts on the floor at a time, such as when standing, sitting, or kneeling, we classify all touch contacts and use inverse kinematics to reconstruct users' entire 3D poses. This combination is of particular interest, as it allows us to build smart rooms entirely based on touch sensing.

1.3 CONTRIBUTIONS AND STRUCTURE

The main contribution of this dissertation is a general approach to providing flat devices with the ability to sense 3D input upon touch by reconstructing this information from the 2D contacts. Below, we preview the details of our approach as part of three applications it enables and outline the structure of this dissertation: (1) high-accuracy touch input, (2) biometric user identification on touchscreens, and (3) extending our approach to a room-size touch floor to enable 3D interaction solely based on 2D input.

To put our contributions in the context of related systems, we survey the work from the literature in Chapter 2, which is relevant to the touch devices and concepts we introduce in this dissertation.

1.3.1 *High-Accuracy Touch: a 3D Input Operation Sensed in 2D*

One benefit of our approach on touch devices is that it enables them to sense touch input with very high accuracy. Traditionally, touch input is believed to be inaccurate for very small targets on touchscreens, such as buttons that measure only a few pixels. In Chapter 3, we challenge this belief and systematically investigate the factors that cause devices to observe inaccurate touch input. We find that the current models for detecting input locations and their implementations are oversimplified: They consider touch in 2D and derive input locations from the center of the contact area between the user's finger and the device (Figure 1.3). The user's finger, however, lives in the 3D space around the device. Our results indicate that the input inaccuracy of current devices is the effect of an incorrect mapping between the 3D of the user's finger to the 2D of the screen.

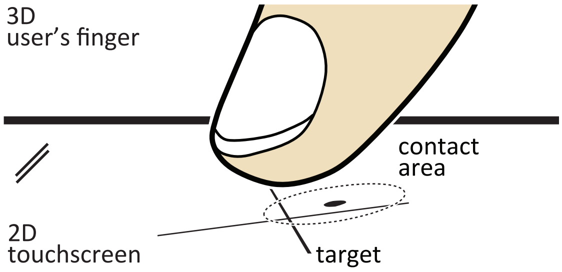


Figure 1.3: Touch is a 3D operation. To acquire the target, the user maps the 2D of the target to the 3D of their finger. To record the touch location, the device maps the 3D of the finger to the 2D space of the screen, typically by sensing the contact area between finger and screen.

As illustrated by Figure 1.4a, we show that deriving input locations from the center of the contact area causes current devices to record touch locations with *systematic error* *offsets*. The spatial arrangement of these error offsets depends on how the user holds the finger in 3D and who the user is. Within each condition, the *spread* of measured input locations is comparably small as indicated by the small cluster ovals in Figure 1.4a. That is, for a particular finger orientation, a touch device observes a comparably small input error, indicating that users can indeed acquire targets with high accuracy using touch. However, current devices lack the notion of 3D touch and do not include the additional degrees of the user's finger orientation into calculating input locations. This causes these devices to observe this systematic effect as a random input error. At the same time, this limits the precision of touch input to the aggregate of all systematically offset clusters (dashed oval in Figure 1.4a).

We address this by introducing the *generalized perceived input point model* to derive touch locations. Our model calibrates touch input on a per-user and per-finger-posture level and compensates for the systematic error offsets current devices observe. This reduces the input error perceived by devices to the spread of the small clusters, as devices can compensate for each of their systematic offsets. In a final evaluation, we demonstrate that our model infers touch-input locations with three times higher accuracy compared to the implementation of existing devices.

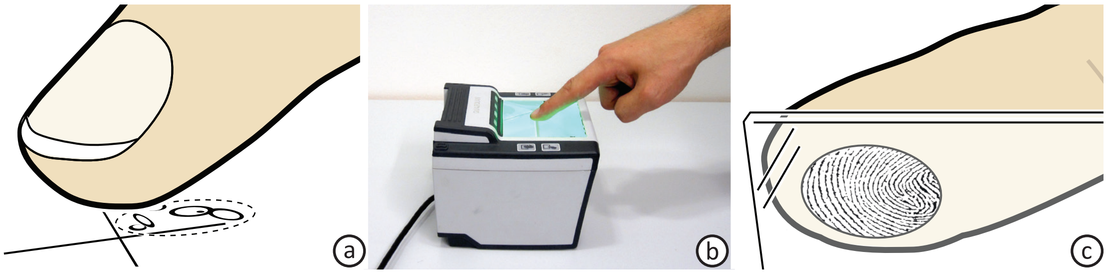


Figure 1.4: (a) Current touch devices infer input locations from the center of the 2D contact area between the user's finger and the input surface. However, this causes them to record locations that are systematically offset from the target depending on the user's finger pose (illustrated by the small ovals). We introduce the *generalized perceived input point model*, which models touch input on a per-user and per-finger-posture level to compensate for these offsets. (b) Our Ridgepad prototype is a high-precision touch-input device based on a fingerprint scanner. (c) Ridgepad implements our new model by identifying the user upon touch based on their fingerprint, and reconstructs the user's 3D finger posture to make 2D touch more accurate.

Based on these insights, we present *Ridgepad* in Chapter 3.4, a high-precision touch-input device (Figure 1.4b). Ridgepad is based on a regular off-the-shelf fingerprint scanner. In addition to recording touch contacts, it also resolves their texture in the form of users' fingerprints. As shown in Figure 1.4c, Ridgepad determines who the user is and how they hold their finger in 3D by comparing the observed fingerprint with pre-recorded images in a fingerprint database, each of which is associated with a known user's name and a 3D finger pose. Depending on the recognized user and

3D pose, Ridgepad applies corrective 2D offsets to make touch input more accurate. In an evaluation, we find that using this approach, Ridgepad improves the accuracy of sensing input by a factor of two, allowing users to reliably touch targets as small as 7.8 mm in diameter. To evaluate the potential of our approach, we compared Ridgepad to a prototype based on sub-millimeter 3D optical tracking, which achieved even 3.3 times higher accuracy, enabling reliable targets to measure only 4.3 mm.

The fact that Ridgepad is able to achieve this substantial increase in touch-input accuracy suggests that the implementation of current touchpads is incomplete. In particular, current devices appear to neglect something that is contained in the user's 3D finger posture, i.e., something that is required to infer accurate touch.

In Chapter 4, we investigate the origins of the effect we observed in Chapter 3 and explore users' *conceptual* understandings of acquiring small targets using touch input. We conduct a series of controlled experiments from which we develop a new model that describes how users touch a target: the *projected center model* (Figure 1.5). In this model, users place certain visual features located on the top of their fingers directly above a target (e. g., the center of the fingernail). This is a departure from the model underlying the design of all current touch devices, which sense input locations as the center of the 2D touch contact, i. e., based on features located at the bottom of the users' fingers. The parallax between top and bottom of users' fingers therefore explains the input error current touch devices observe and Ridgepad's substantial accuracy boost.

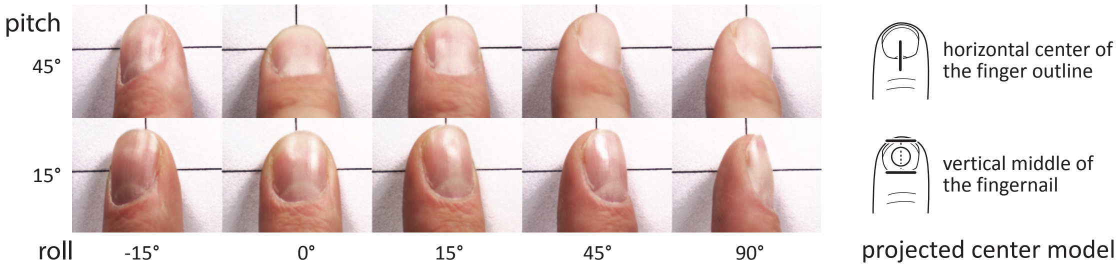

pitch
45°
15°
roll
-15°
0°
15°
45°
90°
horizontal center of the finger outline
vertical middle of the fingernail
projected center model

Figure 1.5: A 3D model of touch input based on visual features. Left: A study participant targeting crosshairs using different finger angles. To acquire the target, our findings suggest that users align *visual* features of their fingers with the target in a hypothesized top-down perspective. Right: We conducted a series of user studies to create and evaluate participants' mental models of touch input. The *projected center model* thereby predicted input locations with the lowest error across all participants and 3D finger postures, suggesting that this is a good description of how users conceptually acquire targets. It says that users target by placing the horizontal center of their finger outline and the vertical center of their fingernail over the target.

Our exploration of users' conceptual models of touch input in Chapter 4 confirms our previous results, which demonstrates that touch needs to be considered in all 3D to serve as an accurate input modality. To realize this finding in practice, touch devices need to sense the top of the user's finger, for example by using an overhead camera. In Chapter 4.10, we demonstrate such an implementation. Our prototype detects touch input on the user's body using a 3D depth camera and implements a visual model to determine input based on features of the user's finger.

However, as mentioned earlier, cameras add volume to the setup, thereby preventing the desirable flat form factor of devices. To achieve this form factor and still interpret touch input in 3D, devices need to reconstruct the 3D finger pose from the observed 2D contacts. This will enable them to sense touch with much higher accuracy while at the same time maintaining their current input modality.

1.3.2 *Biometric Identification and 3D From 2D Reconstruction on a Touchscreen*

As part of making touch input more precise, Ridgepad obtains the user's fingerprint from each touch contact. Based on that, it identifies the user and implicitly reconstructs the user's 3D finger pose.

However, interactive devices offer touch*screens*. Ridgepad is input only and displaying output cannot be added to a fingerprint scanner.

In Chapter 5, we introduce *Fiberio*, an interactive touchscreen that senses fingerprints. Similar to Ridgepad, Fiberio extracts the texture of objects from all touch contacts. At the same time, Fiberio displays images as shown in Figure 1.6. To exploit user identification as a feature, we built Fiberio as a 15.8″×10″ multitouch table—a form factor large enough for use by two to three simultaneous users.

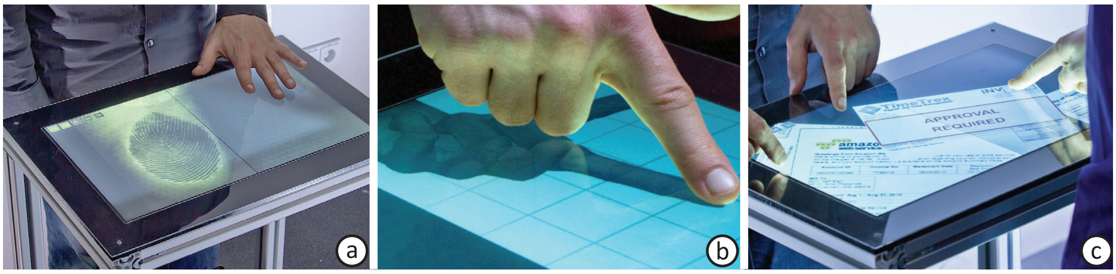


Figure 1.6: Fiberio is a rear-projected tabletop system that captures users' fingerprints during touch. (a) Fiberio is displaying a region of its high-resolution raw input image, revealing the fingerprint of the finger. The key that allows Fiberio to display an image *and* sense fingerprints at the same time is its screen material: a fiber optic plate. (b) Building on Ridgepad's approach, Fiberio reconstructs the user's finger posture in 3D. Here, Fiberio shows a 3D hand that mirrors the 3D finger pose reconstructed from the observed touch image. (c) Fiberio's ability to identify users during touch interaction allows it to support a wide range of applications that require secure authentication (here approving invoices in a bank scenario).

Similar to Ridgepad, Fiberio captures users' fingerprints during each touch. This allows Fiberio to implement user identification and 3D finger pose reconstruction on an interactive touchscreen. Figure 1.6b shows the results of the reconstructed 3D finger pose. Fiberio displays a 3D hand model that mirrors the motion and rotation of the user's finger pose solely based on the observed touch image.

As a side effect, Fiberio addresses a long-standing challenge in human-computer interaction: fine-grained unobtrusive user authentication on touchscreens as shown in Figure 1.6c. Compared to token-based systems, such as access cards, fingerprint-based authentication provides users with more flexible and secure access control [93].

The main contribution of Fiberio is its ability to display images and sense fingerprints in the same location. While Fiberio's setup is similar to a regular diffused illumination table, the key difference is the type of surface material we use: a fiber optic plate that simultaneously provides the properties needed for a projection surface as well as an input surface for scanning fingerprints. The plate consists of 40 million individual fibers and has a resolution of over 4200 dpi. On the one hand, the fiber optic plate diffuses incoming light into all directions, which allows Fiberio to use it as a projection screen. On the other hand, the fiber optic plate provides specular reflection, a property needed to capture high-quality fingerprints. In addition, Fiberio provides the same functionality as traditional diffused illumination tabletop systems in that it senses touch, detects fiducial markers, and estimates users' locations around the table.

1.3.3 *Extending the Same Principles to Room-Size Form Factors*

Having demonstrated 3D reconstruction from 2D touch on touch devices designed for finger input, we generalize our approach to larger touch surfaces in Chapter 6. We present a multitouch floor that senses touch in the form of per-pixel pressure inside a room. Since gravity forces all objects to the ground, our floor captures the imprints caused by users as they walk or sit, as well as those caused by passive objects, such as furniture. Sensing the pressure of imprints at such a resolution thereby reveals the texture of touching objects, such as users' shoe soles, their clothing, or the footprint of passive objects. While this input is very different from the input sensed by traditional touchscreens, we demonstrate that the touch textures extracted from the imprints left on the floor are characteristic and carry enough information to identify users, reconstruct users' 3D postures, and enable high-precision touch input using shoes—similar to how we processed users' fingerprints in previous chapters.

We first present *Multitoe*, a hardware setup that senses multitouch pressure input on a floor. To prototype our approach, we build a table-size installation for foot interaction. Maintaining the same concept, we later present an $8\,\mathrm{m}^2$ floor installation that can accommodate multiple users and objects at the same time as shown in Figure 1.7d.

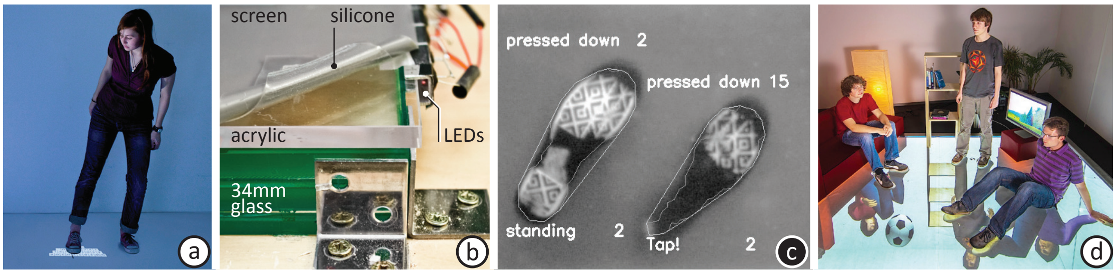


Figure 1.7: We generalize our previous concepts to large multitouch floors. (a) Users interact using their feet on such floors, here to enter text on a touch-sensitive keyboard. (b) Our installation *Multitoe* is a rear-projected floor that detects touch input by sensing per-pixel pressure. (c) The touch image the system obtains from (a). (d) By identifying users based on the shoe soles and classifying all touch contacts, our system *GravitySpace* recognizes people and objects, and reconstructs activity in a whole room solely based on the pressure imprints they leave on the floor. Here, GravitySpace displays the mirrored model of its understanding of the 3D space above the floor, including users and their 3D poses, as well as passive furniture.

We then transfer the concept of direct touch interaction known from mobile devices and tabletop systems to our multitouch floor. We thereby revisit the concepts of user identification and precise touch input that we describe in earlier chapters and explore what they mean for input using feet (Figure 1.7a). Similar to a user's finger, the user's shoe leaves characteristic imprints: the structure of the shoe sole. We use this to identify users and, using the same principles as on Ridgepad, allow them to reliably touch very small targets using their feet.

Building on the concepts we presented in terms of touch devices, our system *GravitySpace* extends 3D reconstruction from 2D touch to a more general reconstruction of the things happening on the floor surface. As shown in Figure 1.7d, we generalize our approach from analyzing the shoes of a standing user to reconstructing the whole 3D pose of each user, such as sitting or kneeling. GravitySpace first classifies touch contacts into body parts and determines to which user they belong. If possible, GravitySpace then infers the location of joints above the floor from the touches that are in contact with the floor using inverse kinematics, such as the user's center of gravity when standing, or the knees when sitting.

This also allows users to interact *above* the floor with virtual objects that are shown on the floor, as GravitySpace estimates the location of a user's foot in the air based on the shifting pressure distribution in the foot that remains on the ground. To extend capturing input to passive objects that rest on the floor, we created specialized furniture that contains vertical bundles of drinking straws. These straws propagate pressure to the floor, allowing GravitySpace to reconstruct users' poses even when they are sitting on the furniture.

1.3.4 *Conclusions and Next Steps: from Flat Devices to Ultra-Mobile Devices*

Chapter 7 concludes the main part of this dissertation. We summarize the contributions of 3D from 2D reconstruction that we make in each chapter and revisit 3D natural user interfaces on flat 2D form factors, both in mobile settings and in room-size installations.

In Chapter 8, we project into the future of interactive devices. While we propose flat devices that allow for "natural" interaction in the main part of this dissertation, in future work we plan on investigating the capabilities and meaning of devices as their form factor continues to shrink. This miniaturization will allow devices to continue to blend into users' environments in the form of ultra-mobile devices. As a first step in this direction, we investigate one manifestation of future ultra-small technology: devices that are implanted under the user's skin. Mainly for medical purposes, millions of people carry implanted devices, such as pacemakers and hearing aids. However, such devices support only limited interaction. We present an early investigation of their capabilities in terms of input, output, wireless communication, and charging.

2 RELATED WORK

The touch devices and processing concepts we introduce in this dissertation are related to materials and setups used in optical touch sensing and algorithms to detect touch-input locations. We also outline related work on touch accuracy and modeling touch input, as well as the importance of finger orientations during touch. Also relevant are systems that identify users and their applications in interactive systems. We particularly examine systems that involve fingerprint scanning, as well as the use of glass fibers to implement such scanners and their use in human-computer interaction. Finally, our work on interactive floors is related to systems that provide foot-based interaction or provide indoor tracking.

2.1 technologies and systems that sense touch input

A wide range of touch sensing technologies have been presented in the related work [140]. While resistive and capacitive solutions to touch sensing are commonly used in commercial products, optical sensing has been explored broadly in the human-computer interaction community.

2.1.1 *Resistive and Capacitive Touchscreens*

For a long time, manufacturers have incorporated resistive touchscreens into devices because of economical costs. Such resistive sensors comprise two layers of transparent conductors on top of the surface of the screen [36]. A thin spacer separates both layers. When touching the screen, the user's finger presses both layers together, causing them to establish contact. To sense the touch location, the device applies a voltage to one conductor in one direction and measures the voltage at the other conductor in the orthogonal direction. The contact established between the two conductor functions as a voltage divider, such that the voltage measured at the second conductor yields one coordinate of the 2D touch position. For the second coordinate, the device applies a voltage to the second conductor and measures the resulting voltage from the first.

This resistive approach to touch sensing is limited to detecting actual contact; it cannot sense objects that hover closely above the surface.

Most mobile touch devices today use capacitive sensing to detect touch input [125, 13], such as smartphones and tablets. Such (projected) capacitive sensors consist of a set of drive lines and orthogonally arranged sense lines, typically made from indium tin oxide. Since this material is transparent, it can be placed on top of the screen without interfering with screen output. When a user touches the surface, the charge of the user's finger affects the capacitance between drive and sensing lines, which impacts the charge measured at the sense lines. Therefore, capacitive sensors observe the *presence* of the user's finger; while the observable electric charge abates rather abruptly as the finger moves away from the surface, capacitive sensors are capable of detecting hovering fingers to a limited extent.

Capacitive touch sensors tend to be more reliable and durable than resistive sensors. Since electric charge passes through glass, the user's finger need not be in direct contact with capacitive sensors. Current devices typically cover the sensor using a protective transparent surface, such as tempered glass. As an example, DiamondTouch is a capacitive multitouch tabletop that is designed for extreme durability [34]. While its surface is opaque and it uses top projection to display output, DiamondTouch incorporates a fiber-glass laminate as the insulating layer above the sensing layer, which protects the sensor and makes the system robust.

2.1.2 *Optical Touchscreens*

Systems based on optical sensing typically use cameras to capture touch input. Although this imposes a certain space requirement, such systems are easy to set up and scale to large dimensions. The latter fact particularly made optical solutions attractive for research on tabletop systems in human-computer interaction.

Diffused Illumination

Systems based on Diffused Illumination implement a simple method for detecting touch input using a light source, a camera, and a diffuser. Examples include Holowall [96] or the Microsoft Surface table [99]. Camera and illuminant are arranged behind the diffusing layer, such that the diffuser scatters all the light from the illuminant towards the objects behind the diffuser. The camera observes the light that is reflected by the user's hands, fingers or other objects. As fingers and objects move away from the diffuser, the diffuser blurs the reflected light more strongly until such objects become unrecognizable in the camera image [16]. If in contact with the diffuser, the blur is weak enough for the camera to resolve fingers and objects, such as fiducial markers.

Despite the blur, diffused illumination allows the camera to resolve objects behind the diffuser to some extent. Systems may leverage this, for example, to sense hovering hands and fingers. On the downside, this fact aggravates detecting the precise moment and location of touch contacts in the camera image; while they appear "less blurry," they do not exhibit a distinct property that determines the part of the finger that is in direct contact with the surface [140].

Diffused illumination is easily integrated with touchscreens. In addition to enabling touch detection, the diffuser simultaneously serves the purpose of producing output in such systems. Often using projectors that are also positioned behind the diffuser (rear-projection), the diffuser scatters the light from the projector and allows output images to emerge on the surface.

Frustrated Total Internal Reflection

Sensing Frustrated Total Internal Reflection (FTIR) to detect touch events allows systems to easily detect the precise moment and area of touch contacts [58]. Such systems also use cameras to detect touches, but inject light laterally into a waveguide, such as a sheet of acrylic. As the light spreads, it is confined within the waveguide, because the optical density of acrylic is higher than the optical density of the air around it. This causes the light to be fully reflected at the top and bottom surface of the waveguide. When a user's finger touches the waveguide, however, the reflections inside the waveguide are frustrated, causing the light to escape the waveguide. This illuminates the contact area of the finger, which the camera observes. The camera therefore produces an image in which only the touch contacts are brightly illuminated.

Because of this property, processing touch input using FTIR is comparably easy. Fingers only light up when they are in direct contact and only the contact area between fingers and the surface is illuminated. Such touch systems therefore detect both the moment as well as the location (i. e., area) of touch input precisely.

When used as part of a touchscreen, the diffuser typically is what allows the projection to image. However, the diffuser at the same time prevents the camera to see past the touch surface. Since all illumination is confined within the waveguide, hovering objects or fiducial markers remain invisible to the camera, because they receive no light from the illuminant.

Overcoming the Diffuser

Due to the nature of the diffuser, the camera in the setups mentioned above resolves fingers or objects only if they are touching or within a small distance to the surface.

To see past the diffuser, touch systems have switched to a range of different surface materials instead.

TouchLight, for example, uses a holoscreen as the touch surface [167]. The holoscreen diffuses light only from a particular direction, while letting light from other directions pass. Arranging projector, camera, and illuminant accordingly allows the projection to image on the surface while the camera can still see through it. SecondLight uses a switchable diffuser to accomplish the same effect [73]. The screen thereby switches between clear and diffuse many times a second and is synchronized with the camera and a shutter in front of the projector. When the screen is diffuse, the shutter is open and allows the light from the projector to pass and image on the screen; when the screen is clear, the shutters block the projector light and the camera captures an image through the now transparent screen.

ThinSight forgoes the diffuser completely and integrates touch sensing into the flat screen of a laptop [65]. ThinSight features an array of infrared light emitters and sensors that is mounted behind the LCD panel of the screen. Similar to diffused illumination, the system senses touch by capturing the reflections from the user's hands and fingers. While the sensing resolution is limited, ThinSight demonstrates how to incorporate optical touch sensing into flat devices.

Top-Down Touch Detection

As an alternative to sensing from behind the surface, some systems have sensed touch input top-down on the surface using a camera. While this approach requires no space behind the touch surface and imposes little requirements onto the surface, it cannot detect touch by observing contact between the user's finger and the surface. Touch-input locations thus need to be determined using features that are visible from above.

The Digital Desk, for example, tracks the location of a user's finger with a camera above the desk and detects touch events using a microphone [164]. The Visual Touchpad also detects touch locations based on visual features and uses two cameras, each mounted at a different oblique angle, to detect touch events on the surface [92]. Agarwal et al. mounted stereo cameras above the user's hands and trained a classifier to recognize fingertips and detect touching fingers as well as fingers hovering above the surface [2]. PlayAnywhere observes fingers from above with a camera mounted at an angle and an illuminant [168]. The system detects touch input on the table when user's fingers approach the shadows they cast. Wilson uses a depth-sensing camera to detect touch locations when the user's finger is close enough to another surface [169]. LucidTouch combines a touchpad mounted to the back of the device with a camera that records the user's fingers [165]. While the touchpad senses touch events, the camera determines input locations and the device displays them along with the fingers to the user on the screen.

2.2 TOUCH-INPUT ACCURACY

Touch devices infer the location of input events depending on their sensing modality; while contact-based technologies observe the contact area, camera-based solutions use visual features to determine input locations.

2.2.1 *Modeling Touch Input*

To operate a user interface, devices need to map touch-input events to screen coordinates. In doing this, devices implement a certain *model* that describes the transformation of the sensed data to a 2D coordinate pair.

Modeling *cursor*-based target acquisition has a long tradition in human-computer interaction. Fitts' Law models the targeting time for one-dimensional targets [43]. Grossman and Balakrishnan model pointing as the *probabilistic* process of acquiring a two-dimensional target under visual control [53]. Both models assume that users can see both the target location as well as the input pointer. This requirement is not fulfilled by touch on targets that are small enough to be occluded by the user's finger, however.

On touch devices, the model that is implemented by a wide range of technologies determines input coordinates from the the observed contact area, which it reduces to a single point, such as the center of gravity [19]. This reduction is necessary to process input locations in the 2D coordinates of the user interface [160]. Examples include touch devices based on capacitive sensing [125] and FTIR [58].

In contrast to processing 2D touch *coordinates*, researchers have proposed considering the entire contact *area* as input. Sliding Widgets, for example, are user interface elements that respond when the contact area touches them [103]. A single touch can thus affect several elements at once. Shapetouch interprets the size of the contact area in terms of the force a touch event represents [26]; while touching an object with the entire palm holds the object in place, a slide with the side of the hand pushes multiple objects aside at the same time.

While touch input works sufficiently well for large user interface targets, it becomes imprecise as an input modality to select very small targets. The common explanation for this inaccuracy is the fat finger problem [157, 142]. This problem comprises two aspects. (1) The soft skin of the fingertip prevents users to use their finger as a precise pointing tool and causes devices to sense input locations anywhere within the contact area between the user's finger and the device. (2) Since users occlude targets during touch with their finger, thereby introducing additional imprecision [15], this prevents the target from providing visual feedback so that users cannot compensate for the randomness.

Researchers have therefore studied the factors of this problem in order to make touch input more precise.

2.2.2 *Touch Precision and the Role of Finger Posture*

Several researches suggested that the angle between the user's finger and the device might impact touch precision. Observing participants that touched targets on interactive tabletops, Forlines et al. noticed that participants' finger pitch changes depending on the location of the target, which at the same time impacted the size of the contact area between the finger and the tabletop surface. As participants touched targets that were farther away from the edge of the table, they held their fingers at a flatter angle, leading to an increased contact area. Since the device used by Forlines et al. inferred touch locations from the centroid of the contact area, this caused the sensed input location to be detected farther from the actual target.

Wang and Ren examined the impact of individual fingers, specific finger postures and gestures on input accuracy and contact area [160]. Their results showed significant differences in how precisely participants were able to touch depending on these factors, which exposed the problem with detecting input from the center of the contact area. Wang and Ren used their results to inform the design of user-interface elements.

2.2.3 *Minimum Target Sizes of User Interface Elements*

As a result of touch inaccuracy, it is commonly understood that a touch target requires a certain minimum size, such that users are able to *reliably* acquire it. While larger targets are one way of addressing this issue, this either limits the number of targets shown at a time or requires larger screens. The latter is especially problematic on mobile devices, where screen space is scarce [15].

In terms of the minimum button sizes for reliable targets, studies in the related work report different values. For example, Hall et al. report a size of 26 mm [56], Wang and Ren find 11.5 mm buttons to be reliable [160], and Vogel and Baudisch obtain 10.5 mm as a reliable button size [157]. All three studies thereby used different touch-sensing technologies.

As a commercial example, Apple advices developers that the "comfortable minimum size of a tappable UI element" is 44×44 pixels in the iOS Human Interface Guidelines [7]. Depending on the pixel density of the device, 44 pixels correspond to 6.85 mm on an iPhone 5 or 8.47 mm on an iPad.

In the light of the results presented in this dissertation, it seems plausible that the disagreement about minimum button sizes is caused by differences in study conditions. Vogel and Baudisch varied finger pitch between the two levels "fingertip" and "nail." In the study conducted by Wang and Ren, it was part of a gesture that combines finger pitch with yaw [160].

Wang and Ren also distinguished individual fingers, while other authors did not. They also recalibrated x/y offsets for every participant, thereby effectively using per-user calibration [Feng Wang, personal communication, 10/03/09]. Finally, there are differences in how users commit a selection, such as by take-off [120, 160] or a button press with the other hand [15].

2.3 IMPROVING TOUCH ACCURACY

Touch devices have improved touch-input accuracy using two complementary approaches. On the one hand, the use of explicit targeting aids has allowed users to be more accurate, albeit at the cost of decreased input speed or less direct touch. On the other hand, some systems aim at improving touch accuracy implicitly, such as by correcting for a bias in sensing input.

2.3.1 *Targeting Aids That Are Part of the User Interface*

A popular approach to circumvent the requirement of a minimum button size is to address the occlusion problem using targeting aids.

Zooming-based techniques reduce occlusion simply by temporarily magnifying screen contents (e. g., [4], TapTap [134], Dual Finger Stretch [19]). While zooming facilitates acquiring small targets with higher precision, this approach cannot fully resolve occlusion issues and in addition reduces users' input speed.

Other targeting aids remove occlusion entirely by separating the user's hand from the pointer. While Offset Cursor shows a selection tool above the user's finger to allow for precise acquisition [120], Shift shows the area under the finger in a callout above the finger [157]. Alternatively, switching touch input to a touchpad on the back of the device [15] removes occlusion and allows users to see where they are touching, as does using a stylus in place of the finger [127].

In contrast to direct touch input, researchers have proposed on-screen elements for precise target selection on touchscreens. Examples include on-screen widgets, such as Cross-Lever and Precision Handle [4] or a mouse cursor operated by both hands (e. g., Dual Finger Midpoint [19]).

On the flipside, targeting aids make touch less direct and may therefore reduce the intuitiveness of input. They also increase targeting time; Offset Cursor, for example, incurs a task time penalty of 250 ms to 1000 ms [120].

2.3.2 *Applying Corrective Adjustments*

To replace targeting aids, researchers have explored options to make touch more precise by applying corrective adjustments to compensate for error offsets. Forlines et al. linked these error offsets to the changes in contact area depending on finger pitch, but did not compensate for this effect [45].

Vogel and Baudisch explained the existence of error offsets as the "perceived input point problem" [157]. When a user tries to acquire a target using their fingertip, the touch device records the contact area, but the center of this area tends to be located "below" the target. Vogel and Baudisch compensate for this effect by applying a constant inverse offset when recognizing touch. Apple's iPhone implements a similar approach and applies a corrective global offset upwards [75].

As we demonstrate in this dissertation, the sensed error offset not just depends on finger pitch, but on the finger's orientation in all three dimensions as well as who the user is. Since detecting these factors allows devices to make touch very accurate, systems that detect the posture of the user's finger are relevant to our work.

2.4 SENSING FINGER ORIENTATIONS AND APPLICATIONS INVOLVING THEM

Different types of touch technologies are able to extract different subsets of the 3D finger posture. The Microsoft Surface table, for example, detects the yaw rotation of the user's finger by analyzing the diffuse reflections of the hovering hand [99]. Capacitive technologies, such as the FingerWorks iGesture pad [42], estimate the yaw orientation of the finger based on the eccentricity of the contact area. AnglePose additionally estimates finger pitch by analyzing the limited range of hover capacitive sensors detect [130].

Some researchers have proposed exploiting finger postures in order to enable additional functionality. Wang and Ren, for example, proposed pie menus that remain free of occlusion by the user's hand by sensing the user's finger orientation [160]. The same authors later also demonstrated hand orientation-aware gesture processing [159]. The proposed algorithm thereby detects finger yaw by observing the changes in contact area over time.

Finger *roll* has been proposed as the basis for a new gesture language (e. g., MicroRolls [135]). The specific implementation of MicroRolls, however, classifies the motion of input samples to detect rolling and dragging; it cannot distinguish between simultaneous finger rolling and finger dragging and has no notion of the difference between the three rotation angles. Therefore, rolling serves primarily as an alternative way of performing a drag gesture. In contrast, the Ridgepad prototype we introduce in this dissertation distinguishes rolling from dragging, even from a single interaction.

In touch systems that detect input using cameras, determining finger orientation becomes comparably easy, because the camera is able to observe the user's entire hand. The Visual Touchpad detects finger yaw using two cameras above the touch surface [92]. LucidTouch detects the user's hands and overlays those parts that are behind the device on the touchscreen [165].

2.5 USER IDENTIFICATION ON TOUCH DEVICES

As we show in this dissertation, identifying the user upon touch not only serves our purpose of making touch input more precise, we also use it to enable devices to offer new functionality for personalized use. While the devices we propose implicitly identify users based on the touch contacts themselves, a number of systems in the related work have explored a wide variety of alternatives to user identification. The majority of touch technologies, however, is ignorant of who touches, such as standard setups using Diffused Illumination [96], FTIR [58], or capacitive sensing [125].

Four general approaches to identifying users on or around touch devices have formed in the related work, listed in the order of increasing reliability: (1) systems that distinguish simultaneously interacting users without obtaining their identity, (2) knowledge-based and appearance-based identification, (3) token-based identification through user-carried objects, and (4) secure user identification.

2.5.1 *Systems That Distinguish Users*

A series of multitouch tabletop systems are able to *distinguish* users that simultaneously interact with the table. For example, DiamondTouch electrically connects users' chairs to the surface of the tabletop [34]. When a user touches the surface, their body closes the circuit between tabletop and chair. This enables DiamondTouch to trace each touch to a chair and distinguish users reliably. While this approach is robust, the number of simultaneous users is limited by the number of seats and DiamondTouch also requires users to remain stationary during interaction.

Other systems have associated touch events with users around the table by tracing their arm to the edge of the table using the reflections of the user's arm above the tabletop [176, 129]. Wang, Cao, et al. estimate the location of a user from the orientation of a touch contact and at the same time use this to distinguish users [159]. Walther-Franks et al. instrument a tabletop with a series of proximity sensors to obtain users' locations around the tabletop [158]. Medusa extends this setup and adds proximity sensors that face up [6]. These sensors additionally capture users' arms, such that Medusa can distinguish simultaneous users and associate touch events with them.

Note that while the previous systems reliably distinguish users, they do not obtain users' identities.

2.5.2 *Knowledge-Based and Appearance-Based User Identification*

A common approach to reliably identify a user from a large set of people on a touch device is authentication using a known secret, such as a PIN code [83]. This solution, however, authenticates users for a whole session and requires an explicit login procedure.

Identifying users based on their appearance has been demonstrated to work well for a limited set of users. Face recognition, for example, achieves high success rates, but requires users to directly face the camera [1].

On touch devices, HandsDown, for example, uses the camera built into a tabletop system and requires users to press their palm against the surface [138]. The system extracts the contour of the hand and obtains the user's identity from the specific dimensions of users' fingers. Bootstrapper is a tabletop system that is equipped with a camera pointed at users' shoes [129]. As users approach the table, Bootstrapper identifies them based on the color pattern of their shoes and links their identity to the touches observed on the table surface. Carpus obtains pictures from the backs of users' hands during interaction from a camera mounted above the tabletop surface [123]. By matching those images against a set of images in its database, Carpus retrieves the users' identities and reliably associates all touch events with them.

To identify users on devices that detect touch using capacitive sensors, devices can be trained to recognize the electrical impedance of the human body. While this approach could be integrated into current smartphones, it has been shown to be limited to two users that remain stationary [61].

2.5.3 *Token-Based Identification Through User-Carried Objects*

To achieve reliable authentication on touch devices, but alleviate the user from typing in a secret, researchers have proposed that users wear or carry identification tokens. Each token is unique, thus ensuring reliable authentication. Examples include the IR Ring, which users wear when interacting with an optical tabletop system [133]. Each ring flashes a unique light pattern, which the camera inside the table observes to identify users. A ring on the user's finger is thereby located close to the observed touch location from the camera's point of view and allows the system to associate touch events with the respective users.

Touching the device by using an object is an alternative solution to user identification. Marquardt et al. proposed gloves with fiducial markers attached to all fingertips and joints [94], which allows optical tabletops to recognize who touches and which of the user's fingers makes contact. PhoneTouch recognizes users with a similar approach, only that users touch the surface of the tabletop using their phones [139]. The system compares the observed touch contacts with accelerometer events sensed by the phones to determine which phone has produced which touch, and maintains a list of associations between phones and users. SurfaceFusion identifies objects that are placed on the tabletop using an RFID sensor inside the table [110]. Of course, all such solutions require users to have an additional object at hand. This renders them less suitable for unencumbered direct touch interaction.

2.5.4 *Secure User Identification*

Researchers have pointed to fingerprint scanning as a solution to user identification on touch devices. Identifying users based on their fingerprints during touch frees them from the requirement of carrying an identification token while still being a reliable authentication mechanism—often considered more reliable than token-based or knowledge-based methods [93]. Fingerprints are a biometric identification feature in widespread use, because they exhibit unique patterns of structural features [8]. (We refer the reader to Maltoni et al. [93] for an exhaustive overview.)

2.5.5 *Applications Involving User Identification*

Interactive touch systems have incorporated user identification for a wide variety of applications. The ability to associate each touch with a particular user has allowed touch systems to personalize interaction [94], log user activity [6], and ensure that only the authorized users can access private objects [138] or perform privileged activities [93].

User identification on shared surfaces has been used for collaborative purposes, such as in educational scenarios [143]. Identifying which student has completed which part of a task that is presented on an interactive tabletop allows teachers to track students' progress [129]. In other use cases, user identification has been used to help children with Asperger syndrome learn social protocol [119].

2.6 FINGERPRINT SCANNING IN INTERACTIVE SYSTEMS

While fingerprint scanners have traditionally been used to identify users, more recently they have also been used as parts of interactive systems.

Sugiura and Koseki envisioned fingerprint scanning to be integrated in future touchscreens and prototyped it using a fingerprint scanner that they placed next to a laptop [150]. Users interacted using the mouse cursor and to simulate a press on a touchscreen, users touched the fingerprint scanner and the system identified them. Their system implemented applications with user interface controls that offer user- as well as finger-specific functionality [150]. Applications of these controls included pasting finger-specific text, starting applications depending on the finger used for touching a button, and copying and pasting content between computers (i. e., pick and drop [126]).

Several patents explain how to control a mouse pointer using a fingerprint scanner. Ferrari and Tartagni's device translates touch input on a fingerprint scanner to movements of the cursor on the screen, allowing users to drag their finger to invoke relative cursor movements [40]. This makes it conceptually similar to a touchpad that is included in current laptop computers. Akizuki's approach translates touches on the scanner into absolute cursor positions [3] and Gust analyzes optical flow in order to extract input motion [54]. A device described by Bjorn and Belongie can distinguish whether users touch the fingerprint scanning surface using their fingertip or a flat finger [20].

To incorporate high-resolution fingerprint sensing into touchscreens, in-cell technology has been hypothesized to one day capture the diminutive structure of fingerprints. In-cell screens place photocells between screen pixels, allowing touchscreens to perceive the light reflected by the structure of objects above the display. Sharp showed an image of a fingerprint captured on a small 2.6″ touchscreen using in-cell technology and VGA input resolution [23]. It is unclear, however, if the quality and resolution of the demonstrated system suffices for processing. Samsung ships a 40″ in-cell touchscreen (Microsoft PixelSense [100]), though with only ∼27 dpi input resolution. This resolution is a factor of 20 too low for high-quality fingerprint scanning. Future in-cell systems may or may not offer the size of current mobile devices and the resolution required for fingerprint scanning, which is considered reliable for user identification at 500 dpi [93].

2.7 APPLICATIONS OF GLASS FIBERS IN FINGERPRINT SCANNERS AND HCI

Since Fiberio uses a fiber optic plate to scan fingerprints on the same surface that the image is projected on, we outline below how related systems use glass fibers to sense fingerprints. We also list applications of glass fibers in human-computer interaction.

2.7.1 *Glass Fibers Used Inside Fingerprint Scanners*

A number of input-only fingerprint scanners have been proposed that exploit the partial reflection of light that occurs inside glass fibers. This Fresnel reflection results from the transition of light from one medium into another [81], such as when light travels from inside glass into air or from glass into human skin—as is the case with fingerprint scanners.

While traditional fingerprint scanners use large glass prisms, allowing them to harvest optimal contrast between the ridges and valleys of the user's fingerprint, scanners based on glass fibers require no camera or lens. Instead, the glass fibers guide the light directly onto the image sensor [97, 109]. However, using glass fibers for fingerprint scanning produces lower contrast between fingerprint ridges and valleys compared to prism-based scanners [97, 93].

Systems in the related work have used various arrangements of glass fibers, light source, and sensor to achieve fingerprint scanning. Examples include setups that consist of slanted glass fibers and an illuminant at the side of the fiber bundle [48, 98] or below the bundle [35]. In both cases, the light from the illuminant is reflected at the top surface inside the fibers and the intensity of the reflection depends on whether or not a fingerprint ridge is in direct contact with the surface. The reflected light is then guided back onto the sensor. The reflections at the top surface are frustrated, however, once a finger is touching the fibers. Other setups operate based on the same concept, but employ straight glass fibers for the touch surface [47]. The apparatus thereby illuminates the user's finger through the space between the fibers and captures the reflected light that is guided onto the sensor using the fibers.

Alternative setups use solid bundles and place them away from the sensor [71]. This, however, enlarges the whole setup and additionally requires a lens to focus all reflections onto the sensor. In this setup, the illuminant emits light at the user's finger through the bundle while the camera is focused onto a small region to capture all reflections.

All previous setups face the challenge of optimally illuminating the user's finger to produce light reflections that are high in contrast. While good illumination is comparably easy to achieve for a small area on the touch surface, such as those that accommodate single fingers, this approach does not scale to larger surfaces, such as tabletops.

Note that all of the previous fingerprint scanners do not produce visual output on the surface to allow for user interaction. They exclusively use glass fibers for scanning, which makes all systems input-only devices.

2.7.2 *Applications of Glass-Fiber Bundles in Human-Computer Interaction*

A number of projects in human-computer interaction have leveraged the property of glass fibers to guide light from one end to the other. Because this property holds true when glass fibers bend up to a certain amount, systems have been able to route light in non-traditional ways compared to standard systems. For example, FiberBoard packs optical sensing into a small form factor by folding the optical paths of its camera system using glass fibers [74]. Lumino fills tangible blocks with glass fibers to allow tabletop systems to detect stacked arrangements of such blocks [16]. Since glass fibers guide the light, stacked blocks transfer light from their top surface to their bottom surface, which works across blocks. Using fiducial markers and a slanted setup of glass fibers, each block projects down the marker of the block stacked on top, such that the camera inside a tabletop system observes the set of all markers juxtaposed on the bottom surface of the lowest block.

Systems have also used glass fibers to create a display. The FUSA^2 system consists of a bunch of fibers and uses a projector to turn the end of the fibers that face the user into a display [106]. The system also senses hovering hands on the fibers' loose ends using a camera and infrared illuminants that are positioned between the fibers.

2.8 INTERACTIVE FLOORS

Because we demonstrate similar outcomes as before on a pressure-sensing smart floor, such as user identification and 3D pose reconstruction, we now list related systems in the domain of foot-based interaction on floors and smart rooms in ubiquitous computing.

2.8.1 *Pressure-Sensing Floors*

A number of (low-resolution) floor prototypes have been presented that are based on pressure sensing. For example, the projection-less magic carpet senses pressure using piezoelectric wires and a pair of Doppler radars [114]. Z-tiles improved on this by introducing a modular system of interlocking tiles [128]. Pressure sensing on floors has also been implemented using force-sensing resistors [156]. FootSee matches foot data from a 1.6 m^2 pressure pad to pre-recorded pose animations of a single user with in a fixed orientation [175].

The UnMousePad implements resistive force sensing and provides a pressure-sensitive touchpad for desktop purposes [132]. Srinivasan et al. built larger-scale installations with a similar approach in the context of floor interaction [145]. They combined the installation with marker-based motion systems as well as audio and video tracking.

Since none of the existing technologies scale to the megapixel range of resolution that we explore in our setup, our prototype uses high-resolution FTIR sensing to detect per-pixel pressure input [58]. The working principle of FTIR does not limit the spatial resolution of pressure sensing; the camera used to record images determines what low-level structure the system can resolve, such as the pattern of users' shoes or texture of their clothing in our case. Systems in the related work have also studied the pressure abilities of large FTIR touchscreens, such as in the context of wall displays [30].

Other instrumented floors that use cameras have mounted them at the ceiling (e. g., iFloor [85]), which is a technique that originated in Ubicomp environments (e. g., EasyLiving [25]). Alternatives have used setups based on front diffused illumination, such as the IGameFloor [52].

Paradiso, Abler, et al. argued that for many types of floor interaction "... fine-grained information delivered by a video camera is unnecessary or potentially inadequate" [114]. An alternative to enabling floor interaction is instrumenting the users' shoes directly. Choi and Ricci created shoes that detect walking direction and speed using buttons mounted under users' soles [27]. The shoes were thereby mainly used for artistic performances and movement training. By adding bending, twisting, orientation, acceleration, and pressure sensors, Paradiso, Hu, and Hsiao gave dance performances extra expressiveness [115]. Kume's Fantastic Phantom Slippers are tracked optically [141], Visell et al. used force sensors to accomplish this [156], and Paradiso and Hu tracked shoes using laser rangefinders [113] and active sonar (Magic Carpet [114]).

2.8.2 *User Identification on Smart Floors*

Interactive floors so far could not use the high resolution our prototypes provide to identify users by their shoe soles. However, footprints have been analyzed as evidence in crime scene investigation [116]. In particular, sole imprints and sole wear have been used to match people either by hand and using semi-automatic techniques based on local feature extractors [117].

Using lower resolution sensing, other prototypes have explored different means to identify users. For example, the screen-less Smart Floor identifies users walking across by observing the forces and timing of the individual phases of walking [111].

2.8.3 *Smart Floors, Rooms, and Multi-Display Environments*

Interactive floors as part of systems in the related work have often been used for natural walk-up-and-use [85], immersion as part of CAVEs [29, 87], gaming [52], and multi-user collaborative applications [85].

The concept of integrating computing into the environment goes back as far as Weiser (Ubiquitous Computing [162]). The concept has been researched in the form of smart components in a room, e. g., in multi-display environments, such as the Stanford iRoom [21] or roomware (iLand [149]). Alternatively, researchers have instrumented the rooms themselves, e. g., using cameras and microphones (e. g., EasyLiving [25]), and made user tracking a key component of their system. The Georgia Tech Aware Home tracks users based on multi-user head tracking and combined audio and video sensing [82]. With the advent of 3D cameras, researchers have started to explore mounting a set of depth cameras inside rooms to sense objects in 3D and allow users to interact in 3D (e. g., LightSpace [170]) or to track users and objects (e. g., interactive projectors [101]).

To enable systems to track otherwise passive objects in a room, such as chairs and shelves, a series of research projects and products have integrated pressure sensing into such objects. Applications include health monitoring, for instance to prevent Decubiti [154], orthopedic use inside shoes [152], and sensing pose while sitting [104, 105].

In contrast to instrumenting all passive objects in a room, we restrict sensing to the floor and design objects so they propagate interaction to the floor, where we can observe it. The pressure-transmitting furniture we present builds on the concept of sensing through an object, which has been explored in the context of tangible objects. Mechanisms include the propagation of light through holes (stackable markers [14]) and glass fibers (Lumino [16]), as well as the propagation of magnetic forces, detected using pressure sensors (Geckos [88]).

3 THE GENERALIZED PERCEIVED INPUT POINT MODEL

This chapter is based on results published in [66].

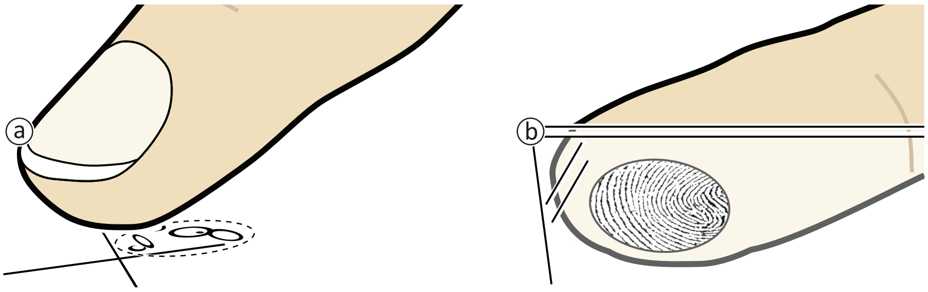


Figure 3.1: (a) A user has repeatedly acquired the crosshairs using finger poses between 90° (vertical) and 15° pitch (almost horizontal). The five ovals contain 65% of the contact points the touchpad recorded for each pitch. The key observation is that the ovals are *offset* with respect to each other, yet small. We find a similar effect across different levels of finger roll and finger yaw, and across users. We conclude that the inaccuracy of touch (dashed oval) is primarily the result of failure to distinguish between different users and finger postures. (b) The ridges of this fingerprint belong to the front region of a fingertip. Our *Ridgepad* prototype uses this observation to reconstruct finger pose and user ID, which allows it to exploit the new model and obtain 1.8 times higher accuracy than capacitive sensing.

It is generally assumed that touch input cannot be accurate because of the *fat finger problem*, i. e., the softness of the fingertip combined with the occlusion of the target by the finger. In this chapter, we show that this is not the case. We base our argument on a new model of touch inaccuracy as shown in Figure 3.1a. Our model is not based on the fat finger problem, but on the *perceived input point problem* [157]. This problem describes input error as an effect of touch screens reporting sensed input locations at an *offset* from the target the user intends to touch. From this problem, we derive a generalized model that represents offsets for individual finger postures and users. We thereby switch from the traditional 2D model of touch to a model that considers touch a phenomenon in 3D space. We report a user study, in which the generalized model explained 67% of the touch inaccuracy that was previously attributed to the fat finger problem.

We then present two devices that exploit the new model in order to improve touch accuracy. Both models consider touch on a per-posture and per-user basis in order to increase accuracy by applying respective offsets. Our *Ridgepad* prototype extracts posture and user ID from the user's fingerprint during each touch interaction (Figure 3.1b). In a user study, it achieved 1.8 times higher accuracy than a simulated capacitive baseline condition. A prototype based on optical tracking achieved 3.3 times higher accuracy. The increase in accuracy can be used to make touch interfaces more reliable, to pack up to $3.3^2 > 10$ times more controls into the same surface, or to bring touch input to very small mobile devices.

3.1 INPUT ACCURACY ON TOUCHSCREEN DEVICES

Acquiring a small target on a touch screen is error prone. We can examine how inaccurate touch is on a given device by letting users acquire a small target repeatedly: the more inaccurate the device, the wider spread the distribution of the sensed contact points (indicated with a dashed outline in Figure 3.1a; the user has targeted the crosshairs). When acquiring a target of finite size, such as a button, wider spread results in an increased risk of missing the target.

The common explanation for the inaccuracy of touch is the *fat finger problem*. That is, the softness of the user's skin causes the touch position to be sensed anywhere within the contact area between the user's fingertip and the device. At the same time, the finger occludes the target. This prevents the target from providing visual feedback so that users cannot compensate for the randomness.

Researchers have therefore argued that users *cannot* reliably acquire targets smaller than a certain size.

3.2 AN ALTERNATIVE MODEL FOR THE INACCURACY OF TOUCH INPUT

While the fat finger problem has received a lot of attention, we argue that it is *not* the true reason for the inaccuracy of touch. We argue that the *perceived input point model* is the primary source of the problem. Vogel and Baudisch mention it in the same paper that discusses the fat finger problem [157]. When a user tries to acquire a target, the center of the contact area tends to be located a couple of millimeters off the target location—typically "below" the target (the black dot in Figure 3.1a). Unlike the fat finger problem, however, the "perceived input point problem" is a *systematic* effect. This allows compensating for the effect by applying an inverse offset when recognizing touch.

We generalize the perceived input point problem in order to reduce touch inaccuracy even further. We hypothesize that the offset between the center of the contact area and the target depends not only on the x/y location of the target, but also on the *wider context* of the touch interaction. The *wider context* in this *generalized perceived input point model* could potentially include a larger number of variables. We examine the following four in this chapter:

1–3. *Angle* between the finger and the touch surface (roll, pitch, and yaw). The related work suggests that pointing might be affected by changes in finger orientation (also called "*yaw*") [159] and finger "steepness" (or "*pitch*") [45]. We also include finger roll. By considering all three finger angles, we implicitly switch from the traditional 2D model of touch to a model that considers touch a phenomenon in 3D space.

4. *User*. Each user might have a different mental model of how to acquire the target. While touch is well understood in the macroscopic world (most people will agree on whether a person is touching the seat or the backrest of a chair), note that there is probably no universally agreed upon interpretation for determining what *exact* location a finger is touching.

To verify these assumptions we conducted a user study.

3.3 STUDY 1: IMPACT OF USER AND 3D FINGER POSE ON TOUCH ACCURACY

The generalized perceived input point model makes the assumption that the offset between the contact point and target depends on the *wider context* of the touch interaction, in particular roll, pitch, yaw, and user ID. The purpose of this study was to verify this assumption.

Our main hypothesis was that a variation of touch context, i. e., a variation of finger posture and/or user ID, would result in distinct clusters of touch positions. Figure 3.2 illustrates this. A participant has repeatedly acquired a target using five different finger postures. Each one results in a distribution, which we illustrate using an oval. If touch inaccuracy is governed primarily by the fat finger problem, we expect to see large ovals, all of which are centered on roughly the same point (Figure 3.2a). If the inaccuracy of touch, however, is primarily explained by the generalized perceived input point model, we expect to see ovals that are visibly *offset* with respect to each other (Figure 3.2b), yet each of them is small in size. That is, the *spread* in measured input locations within each condition is small (i. e., the average distance of all samples in a distribution from its center of gravity).

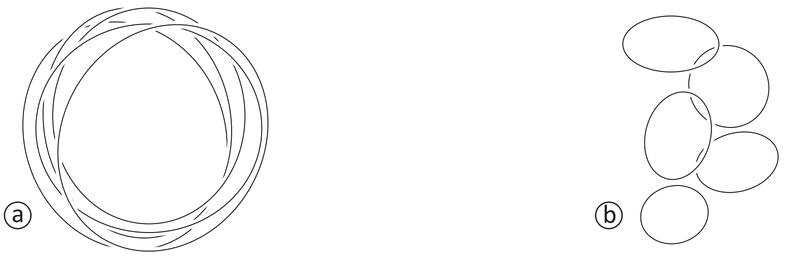


Figure 3.2: Main hypothesis. Expected outcome if touch inaccuracy is caused primarily (a) by the fat finger problem or (b) by the generalized perceived input point model.

3.3.1 *Task*

Figure 3.3 shows a participant during the study. A touchpad showed a single target, which participants acquired repeatedly. (There was no reason to include distracter targets. Distracters have a major effect on *adaptive* input techniques, such as magnetic targets [18], but not on unmodified touch). During each trial, participants first touched the start button on the pad (labeled "okay" in Figure 3.3). Then participants assumed the finger angle for the current condition with their right index finger and acquired the target. Participants committed the touch interaction by pressing a foot switch. This recorded the touch location reported by the touch pad, played a sound, and completed the trial. Participants did not receive any feedback about the location registered by the touchpad.

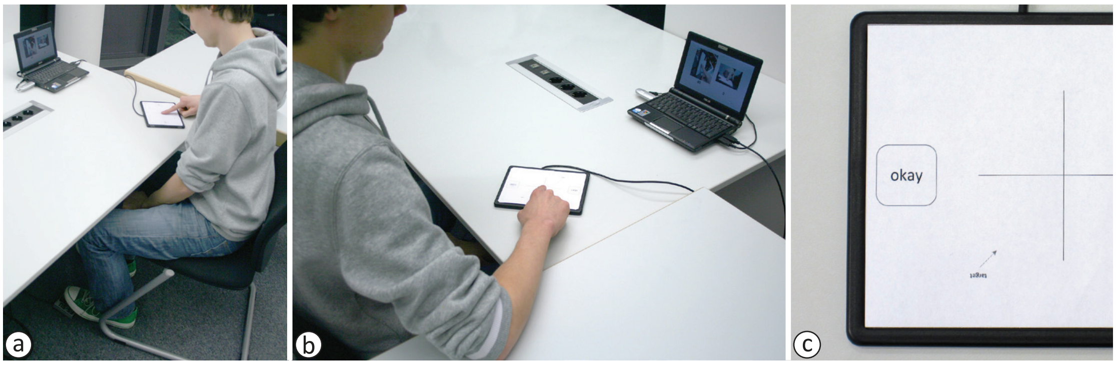


Figure 3.3: (a) A participant acquiring the crosshairs in the touchpad during the study. The laptop displays instructions and the finger orientation for the current trial. Participants committed each trial by pushing the foot switch. (b) Participants rested their elbow on the table to prevent fatigue. (c) The crosshairs mark the target.

We took the following four measures to minimize the impact of other potential factors. First, participants kept their head in a fixed position above the touchpad, as shown in Figure 3.3. This controlled for parallax. Second, the crosshairs marking the target extended beyond participants' fingers, allowing participants to maintain a certain amount of visual control during targeting. Third, the use of a foot switch allowed us to avoid artifacts common with other commit methods, such as inadvertent motion during take-off. And finally, participants were told to use as much time as necessary and that task time would not be recorded. Every participant completed their 600 trials in under 40 minutes.

In the case of an accidental commit, such as activating the foot switch twice, the system discarded the input, notified participants with an acoustic signal, and had them repeat the trial.

3.3.2 *Finger Angles: Roll, Pitch, and Yaw*

Participants acquired the target using their right index finger with five different levels of pitch and five different levels of roll (Figure 3.4). We varied pitch between "close to horizontal" = 15° and "straight down" = 90°. Pitch values beyond that caused the fingernail to touch first, which clashes with many types of capacitive devices. A roll of 0° meant that the nail was horizontal. We varied roll between "rolled slightly left" = -15° and "rolled fully to the right" = 90°.

Varying roll and pitch separately allowed us to keep the number of trials manageable. During the pitch session, participants kept finger roll horizontal (0°), while they used a fixed pitch angle of 15° during the roll session. Combinations of pitch and roll are to be interpreted pitch-first. A pitch/roll of 15°/45° thus means "assume a pitch of 15° and then roll the finger 45° to the right."

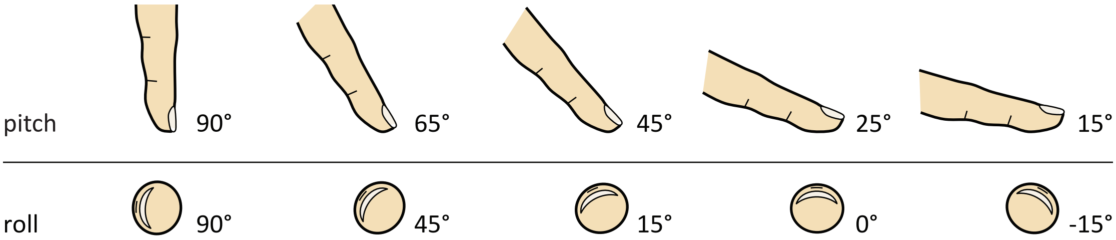


Figure 3.4: Participants acquired the crosshairs by first assuming a combination of these finger pitch and finger roll angles and then touching the target.

We also studied the third angle, i. e., yaw. However, there was no need to vary it, because yaw takes places in the plane of the touchpad. As a result, we can reconstruct all levels of yaw by rotating the touch locations (obtained from a single level of yaw) post-hoc in software around the target. This, however, requires knowledge of the target location. Since the capacitive pad cannot see the target, we approximated its location by testing two levels of touchpad orientation (0° and 180°). We then determined the rotation center as the center of gravity among all touch locations, before we flipped the 180° condition and merged its samples with the 0° condition. To make sure that the 180° condition be identical from the participants' perspective, participants operated a second "okay" button on the opposite of the touchpad (cropped in Figure 3.3c).

3.3.3 *Procedure*

To keep fatigue at a reasonable level, participants performed their trials in two sessions.

In one session, participants performed five variations of pitch {15°, 25°, 45°, 65°, and 90°}. In the other session, participants performed five variations of roll {-15°, 0°, 15°, 45°, and 90°}. Session order was counterbalanced across participants.

Within a session, participants performed a sequence of 150 trials with the touchpad in one orientation and then a second sequence of 150 trials with the touch pad in the opposite orientation. Pad rotation was counterbalanced across participants. For each sequence, participants completed 5 blocks of 5 angles × 6 repetitions each. The order of finger angles was counterbalanced across trial blocks.

Each participant completed all conditions. Each participant completed 5 angles × 2 pad orientations × 2 sessions × 5 blocks × 6 trials per block = 600 trials.

3.3.4 *Apparatus*

The touchpad was a 6.5″×4.9″ capacitive FingerWorks iGesture multi-touchpad [42]. It was connected to an Asus eeePC 900HD. The foot switch was a Boss FS-5U.

3.3.5 *Participants*

We recruited 12 participants (3 female) from our institution. All participants were right-handed and between 17 and 34 years old. Each received a small gratuity as a compensation for their time and we awarded €20 to the most accurate participant.

3.3.6 *Hypotheses*

We had one main and four dependent hypotheses. Our main hypothesis was that a variation of touch context, i. e., a variation of finger posture and/or user ID, would result in significantly different clusters of touch positions. The dependent hypotheses spell this out for the individual variables.

1. *Pitch:* Different levels of pitch result in *distinct* touch location clusters. In other words, we expected to find higher spread across pitch levels than within a given pitch level.
2. *Roll:* analog to pitch.
3. *Yaw:* analog to pitch.
4. *User:* Cluster organization will differ across participants. Different users have different finger shapes and we hypothesized they might also have different mental models of how to map their large fingers to a small target.

3.3.7 *Results*

Figure 3.5 summarizes the complete touch location data obtained from this study. Each column summarizes the recorded locations for one participant; the top chart shows aggregated clusters of touch locations for the different levels of roll, the bottom chart shows the aggregation of the pitch session. All ovals in Figure 3.5 represent confidence ellipsoids that contain 65% of the recognized touch locations per condition. The crosshairs in each chart is the target location. Figure 3.6 shows two examples in additional detail (pitch data of Participants 3 and 4).

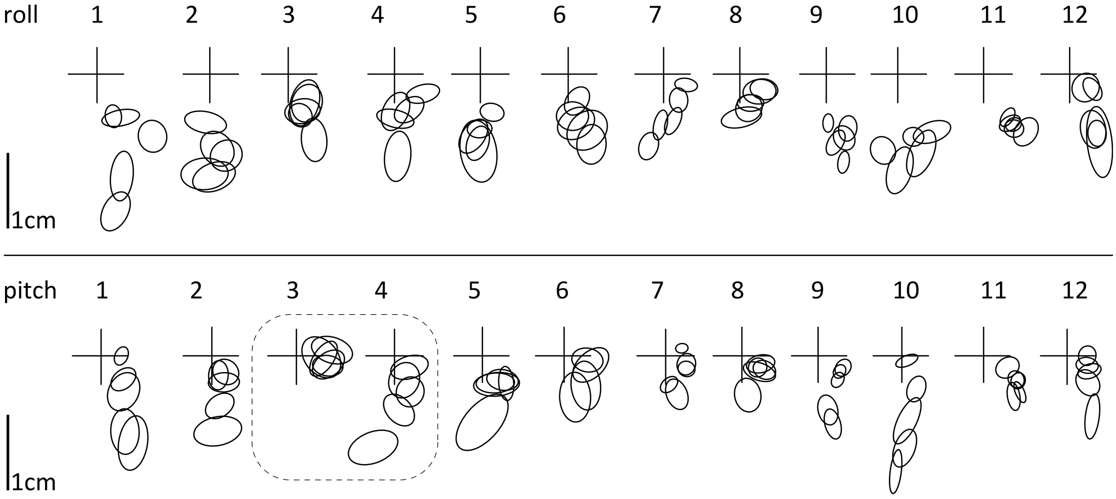


Figure 3.5: Clusters of touch locations for each of the 12 participants (columns 1–12). Crosshairs represent target locations; ovals represent confident ellipsoids. (a) Each of the 5 ovals represents one level of *roll*. (b) Each of the 5 ovals represents one level of *pitch*. All diagrams are to scale. Note how different patterns suggest that each participant had a different interpretation of touch.

Pitch

We analyzed the effect of pitch using a repeated measures one-way ANOVA. To better understand the nature of the differences, we decomposed the differences in recognized touch position into differences *along* the finger axis (y axis in the chart) and *across* the finger axis (x axis in the chart). Changing finger pitch had a significant effect on recognized touch positions ($F_{4,8} = 6.620$, $p = 0.012$) *along* the finger axis. Pair-wise comparisons using Bonferroni-corrected confidence intervals showed that the touch locations of all levels of pitch were significantly different (all $p < 0.05$). We also found a main effect of pitch on touch location *across* the finger axis ($F_{4,8} = 6.972, p = 0.01$). However, pair-wise post-hoc tests showed no significant differences.

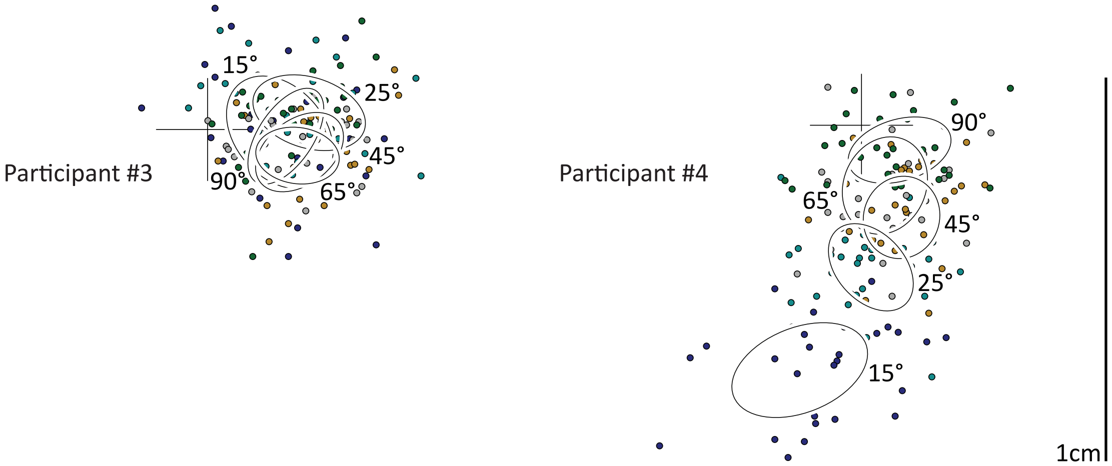


Figure 3.6: Close-up of touch locations organized by pitch of Participants 3 and 4 from Figure 3.5. Even though clusters are much further apart for Participant 4, both are equally "accurate" under the generalized perceived input point model.

Roll

A repeated measures one-way ANOVA found a significant main effect of roll on sensed touch position *along* the finger axis ($F_{4,8} = 4.574, p = 0.032$). Bonferroni corrected pair-wise comparisons showed a significant difference between 90° roll and all other roll levels, as well as 45° vs. -15° and 0° (all $p < 0.05$). An ANOVA on touch location *across* the finger axis did not find an effect ($F_{4,8} = 1.444, p = 0.305$).

Yaw

We ran paired-samples t-tests comparing touch locations *across* and *along* the finger axis in the two yaw conditions. We found both to be significantly different (*across*: $t_{11} = 6.570, p < 0.001$; *along*: $t_{11} = 9.361, p < 0.001$).

Participant

We ran a two-way ANOVA on finger pitch and participant both along and across the finger axis, using participant as a random factor. We found a significant interaction between pitch and participant and significant main effects for both (all $p < 0.001$). For each participant, we ran separate one-way ANOVAs on finger pitch to determine where the effect was particularly evident. We found a significant main effect on touch location *along* the finger axis for all participants and a significant main effect *across* the finger access for all but one participant (all $p < 0.05$).

Similarly, we ran a two-way ANOVA on finger roll and participant. We found significant main effects for participant along and across the finger axis, as well as for finger roll along the finger axis. We further found a significant interaction between finger roll and participant along and across the finger axis (all $p < 0.001$). This indicates that each participant exhibits a different behavior and touch pattern in response to finger roll. We ran one-way ANOVAs on finger roll separately for each participant. We found a significant main effect of finger roll both along and across the finger axis for all participants (all $p < 0.05$), except one whose error rates did not differ significantly across the finger axis.

3.3.8 *Discussion*

Finger angles

As hypothesized, all three angles had an impact on touch location and led to distinct clusters, supporting hypotheses 1–3. As expected based on work by Forlines et al., finger pitch primarily impacted touch location along the finger axis (visible as vertical patterns in Figure 3.5, bottom row). A "flatter finger" caused the touchpad to locate input coordinates farther away from the target towards the user's palm. Somewhat surprisingly, variations in *roll* impacted touch location primarily *along* the finger axis as well, more than across (visible as vertical patterns in Figure 3.5, top row). Finally, as expected, there also was a significant effect of yaw on touch location. This is also obvious in Figure 3.5 where none of the groups of ovals are centered on the target. This emphasizes the fact that global offsets, as applied by Vogel and Baudisch [157] need to consider hand yaw.

Users

Also as hypothesized, there was an effect of user on the touch location. As shown in Figure 3.5, the clusters of recognized touch positions varied across participants, and they did so quite substantially. Figure 3.6 shows a particularly different pair: For Participant 4, touch locations vary drastically with pitch, while pitch has very little impact on the touch locations produced by Participant 3.

Based on this chart, one might think that Participant 3 is simply more *accurate* than Participant 4, i. e., that Participant 3 performed the task with additional care. Whether this is true or not is a matter of perspective. When we look at the size of the individual clusters of the two users, we see that they are roughly comparable. This means that both participants reproduced the target location equally well. What differs between the two participants is their mental model. Participant 3's understanding of touch coincides strongly with the capacitive touchpad model.

So while Participant 3 is more fit than Participant 4 when operating *today's* touch devices, when using an input device based on the generalized perceived input point model, this is not the case anymore. As we explain in the following sections, such a device compensates for roll, pitch, yaw, and user ID. "Accuracy" now means neither the proximity of a cluster to the target (because we can compensate for it), nor the proximity of clusters to each other (again, because we can compensate for it). Instead, accuracy now means size of clusters, as all other factors can be compensated for. Since the cluster sizes for Participants 3 and 4 are comparable, this means both participants will perform equally well under the new model.

Main Hypothesis

Overall, and most importantly, our study supports our main hypothesis: Roll, pitch, yaw, and user ID all lead to *distinct* clusters (i. e., significantly different average centroids). As a matter of fact, these clusters are clearly separated, as discussed earlier when explaining Figure 3.2. Our findings therefore support that the generalized perceived input point model indeed explains a significant part of the inaccuracy of touch, rather than the fat finger problem.

Exploiting the Model With a Device

These observations suggest that a device should be able to obtain improved accuracy by applying compensating offsets for each condition.

The data from our study allows us to make predictions about the performance such a device might achieve. Figure 3.7 shows a summary. Each bar denotes the diameter of a round button that contains 95% of all touches, assuming that we apply compensating offsets for different subsets of factors. Each bar was computed by mapping the centroids of different sizes of clusters to the target center location. For the traditional touchpad condition (left bar), no mapping was applied. For the per yaw condition, the centroid of all touches was moved to the target. For the per yaw and roll/pitch condition, the centroids of each roll cluster and each pitch cluster were moved to the target. For the per all condition, the centroids of each roll cluster and each pitch cluster for each participant were moved to the target.

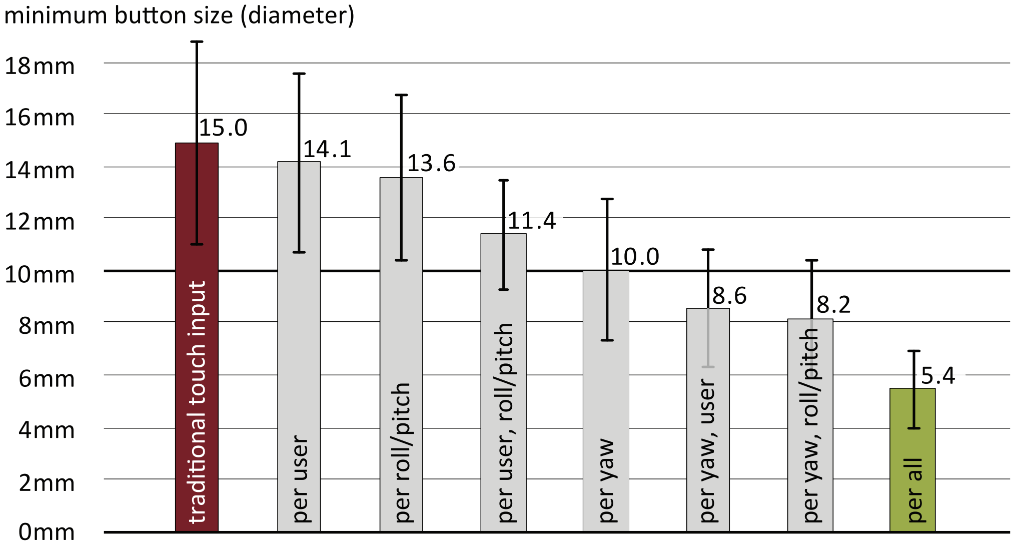


Figure 3.7: Minimum size of a button that contains 95% of all touches on a touch device that knows about different subsets of roll/pitch, yaw, and user ID. Error bars encode standard deviation across all samples.

As illustrated by the chart, each additional piece of information should allow the device to further improve its accuracy up to a factor of 2.75 if all angles and user ID are included. Instead of buttons measuring 15 mm, such a device should allow users to reliably acquire buttons measuring 5.4 mm.

A factor of 2.75 suggests considerable potential. In order to exploit it, however, we need a touch device capable of extracting these four additional variables from a touch interaction. We have created two prototypes of such devices. One of them is *Ridgepad*.

3.4 ridgepad: a high-precision touch input device

Ridgepad is a touch input device that implements the requirements of the generalized perceived input point model based on a regular off-the-shelf fingerprint scanner (a

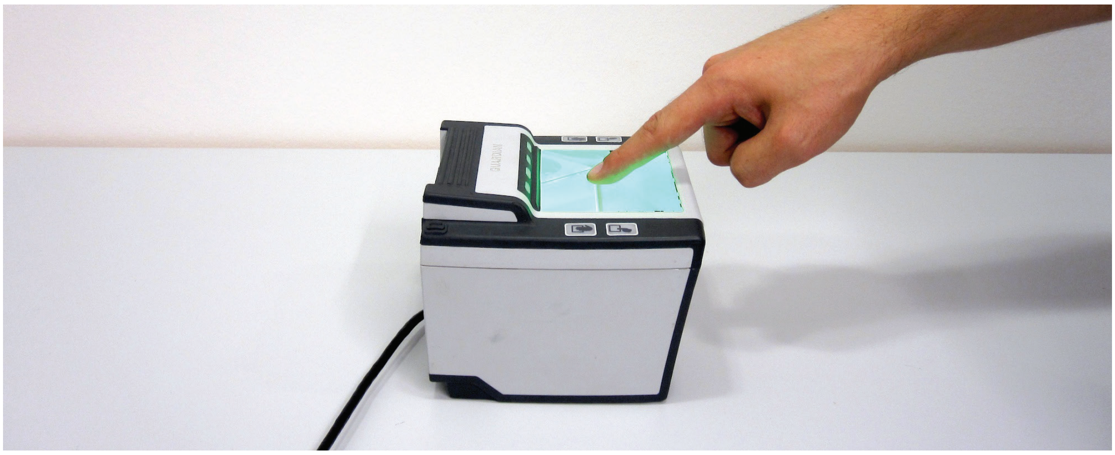


Figure 3.8: Our *Ridgepad* prototype is based on an *L SCAN Guardian* fingerprint scanner.

high-resolution optical *L SCAN Guardian*, see Figure 3.8). Ridgepad senses input only and provides no visual output to the user.

The fingerprint scanner captures the *contact area* between the user's finger and the device. That is, it precisely observes the parts that are in direct contact with the surface.

Traditional touch devices, such as the FingerWorks pad we used in the previous study obtain only the contact area of the finger with the surface. Ridgepad obtains the same contact area, albeit in high resolution from the user's fingerprint. The fingerprint offers two additional types of information. First, it allows Ridgepad to identify the user. Second, it allows Ridgepad to analyze the portion of the user's fingerprint that is located inside the contact area. Based on its analysis of which part of the fingerprint touches the screen (Figure 3.1b), Ridgepad infers all three finger angles, i. e., yaw, pitch, and roll.

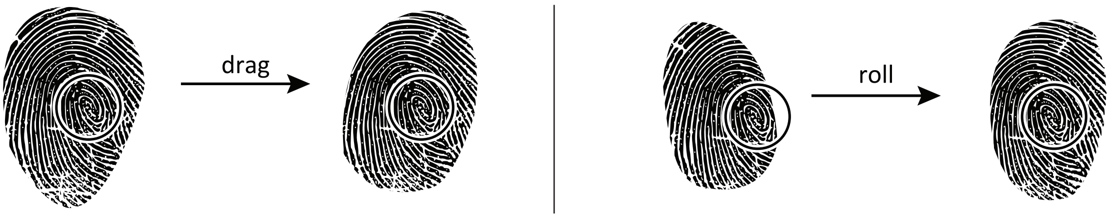


Figure 3.9: (a) When dragging, fingerprint outline and features move in synchrony. (b) When rolling the finger on the surface, fingerprint features remain stationary.

This mechanism allows Ridgepad to extract rolling and dragging from a single interaction. As shown in Figure 3.9a, when dragging, fingerprint outline and features move in synchrony (i. e., features do not change their position relative inside the fingerprint).

When rolling the finger on the surface, however, fingerprint features remain stationary (Figure 3.9b).

3.4.1 *Algorithm: Calibrating Ridgepad for High Input Accuracy*

By default, Ridgepad is only as accurate as a regular touchpad. Similar to such touchpads, it infers touch-input locations from the center of the contact area. Ridgepad achieves improved accuracy through calibration.

During calibration, users repeatedly acquire a single target on the fingerprint scanner. It is not necessary for users to touch under specific roll, pitch, and device rotations; the more postures users cover, however, the more postures will benefit from improved precision.

Every calibration trial produces a pair of a fingerprint image and an associated target position relative to the center of the fingerprint, i. e., an error offset. All such pairs are stored in the user's profile.

The profile is user-specific, but not device-specific. This allows users to calibrate future devices instantly using an existing profile that is associated with their fingerprint.

3.4.2 *Algorithm: Using Ridgepad As a Touch Device*

During actual use, users touch Ridgepad's surface just like any other touch device. Ridgepad computes the center of the contact area as a reference point. It then compares the observed fingerprint with all fingerprints in its database (the search space is reduced to the user's profile as soon as the user has been identified). Ridgepad compares fingerprints using the generic image-matching algorithm SURF by Bay et al. [17]. SURF extracts features from images, such as intersections of lines. It then finds the image transformation that maximizes the number of features that line up.

The number of fingerprints in the profile that have *some* match with an observed fingerprint is typically large. To determine which fingerprints are most likely to represent the pitch and roll position of the observed fingerprint, Ridgepad simply uses the number of features SURF is able to match as a metric. This works because two images are likely to exhibit similar features if and only if similar parts of the finger touched the surface. *All* features typically match only if pitch, roll, and yaw are identical.

Based on this similarity function, Ridgepad looks up the k closest matches in the user's profile (k-nearest neighbor algorithm). Ridgepad then averages the offset values associated with the chosen neighbors (optionally with additional weight for better matches) and finally adds that offset to the center of the current touch location.

3.4.3 *Hardware Implementation*

The Guardian fingerprint scanner in our prototype offers a 3.2″×3.0″ touch area. As common for fingerprint scanners, it works based on frustrated total internal reflection. Unlike FTIR implementations in current tabletop or wall systems, such as [58], the glass surface is illuminated from *below* and its illumination is frustrated by the finger (Figure 5.2 in Section 5.3 explains the optical path in detail). The Guardian provides images at 500 dpi, which allows for high-quality fingerprint recognition.

3.4.4 *Benefits and Limitations*

The generic nature of its algorithm makes Ridgepad particularly robust and flexible. Ridgepad finds matches for a given yaw/pitch/roll/user fingerprint, because it finds other fingerprints that "look" similar; nowhere in the system are they ever labeled with angles. While we designed the algorithm to work with roll, pitch, and yaw, it is independent of any such specifics. It should therefore be straightforward to extend the algorithm to other features, such as finger pressure.

One of the limitations of the current implementation is time complexity. The Guardian fingerprint scanner in our prototype requires a noticeable pause before transmitting a picture. In addition, our non-optimized prototype code sequentially compares fingerprints with all fingerprints in the user's database, which takes 200–300 ms for each comparison. Future versions should be able to achieve real-time performance by extracting features up-front and comparing feature vectors using more suitable data structures.

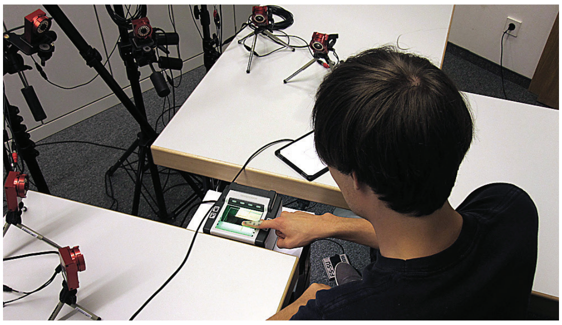
3.5 STUDY 2: TOUCH PRECISION WITH RIDGEPAD

To verify the performance of touch devices based on the generalized perceived input point model, we conducted a second user study. We compared Ridgepad and a device design based on an optical tracker with a traditional baseline condition. Similar to the first study in this chapter, we analyzed the effect of roll, pitch, and yaw. Our main hypothesis was that Ridgepad and the optical tracker would outperform the baseline condition.

3.5.1 *Interfaces*

We tested three interface conditions, all of which were implemented by the hardware setup shown in Figure 3.10 (left). This particular setup allowed us to process every targeting trial with each of the three interfaces simultaneously.

1. The FINGERPRINT interface was implemented using Ridgepad and employed the algorithm described in the previous section.
2. The CONTROL interface simulated a traditional touchpad interface. It received the same input from the fingerprint scanner as the *fingerprint* interface. However, this condition did not use the fingerprint features and instead reduced the fingerprint to a contact point at the center of the contact area.
3. The OPTICAL TRACKER interface was implemented based on a six-degree of freedom optical tracking system (an 8-camera *OptiTrack V100* system). To allow the system to track the participant's fingertip, we attached five 3-mm retro-reflective markers to the participant's fingernail as shown in Figure 3.10 (left). The extreme accuracy of the optical tracker made this interface a "gold standard" condition that allowed us to obtain an upper bound for the performance enabled by the generalized perceived input point model.



roll	-15°	0°	15°	45°	90°
pitch 15°	×	×	×	×	×
25°		×			
45°	×	×	×	×	×
65°		×			
90°		×			

Figure 3.10: Left: The three interfaces: The fingerprint scanner simultaneously implemented the FINGERPRINT interface and the CONTROL interface. The red cameras around implemented the OPTICAL TRACKER interface, which was based on an *OptiTrack VT100* system. Between trials, participants tapped the touch pad. They committed using the foot switch. Right: Angles for finger pitch and finger roll that participants first assumed before acquiring the target on the touchpad.

Similar to the FINGERPRINT interface, the OPTICAL TRACKER interface applied corrections by averaging offsets from $k = 13$ training samples that matched in terms of roll, pitch and yaw. The OPTICAL TRACKER interface obtained these angles directly from observing the location of the markers in 3D space.

Since FINGERPRINT interface and OPTICAL TRACKER interface required per-user calibration, we used 80% of all trials (520 of 650) as training data for the respective calibration procedures. We used the remaining 20% of all trials (130) for the actual analysis.

3.5.2 *Task*

As in our first study, participants acquired a single target repeatedly. The target was drawn onto the surface of the fingerprint scanner. Half of all participants acquired a target marked with crosshairs similar to our first study. The other half of participants acquired a target marked with only a dot. The additional independent variable crosshairs vs. dot allowed us to study the impact of the occlusion problem. As in the first study, participants pressed "okay," acquired the target, and committed using a foot switch. All participants completed all trials of one session in about 30 minutes.

3.5.3 *Procedure*

Participants completed the same roll/pitch combination as in the first user study plus four additional variations of roll across 45 deg of pitch as shown in Figure 3.10 (right, additions are highlighted in bold).

Participants completed the study in two sessions; the second session was identical to the first, except that we rotated the scanner for the same reasons as in our first study. We counterbalanced the order of all conditions within sessions as well as sessions and rotations across participants. Overall, participants completed 2 sessions × 5 blocks × 5 repetitions × 13 angles = 650 trials.

3.5.4 *Participants*

We recruited a fresh set of 12 participants (2 female) from our institution. All participants were right-handed and between 22 and 34 years old. Again, we gave each participant a small gratuity for their time and awarded € 20 to the most precise participant.

3.5.5 *Apparatus*

The fingerprint scanner and the optical tracker were powered using an Intel Core 2 Duo machine running at 3GHz with 3GB of RAM. We reused both the foot switch and the FingerWorks pad from our first study; however, the latter was only used for the "okay" button in this study.

3.5.6 *Hypotheses*

We had two hypotheses:

1. Since optical trackers track angles with extremely high precision, we expected the optical tracker interface to redeem the entire accuracy benefit suggested by the first study, i. e., an improvement of a factor of 2.75 compared to the simulated capacitive control interface.
2. Ridgepad cannot reconstruct angles quite as accurately as an optical tracker. Still we expected the fingerprint interface to improve input precision substantially compared to the simulated control interface.

We were also curious to see how large the improvement of the fingerprint interface would be compared to the control interface.

3.5.7 *Results*

Similar to the analysis of our first study, we compared the spread of recorded input locations (i. e., the mean distance of all points in a condition from their center of gravity). We compared the mean input spread for each participant when using each interface.

We ran a one-way ANOVA on averaged per-participant spread with participant as a random variable and found a significant main effect of interface on spread ($F_{2,9} = 49.457, p < 0.001$). Pair-wise comparisons using Bonferroni-corrected confidence intervals showed statistically significant differences of spread between all interfaces ($p < 0.01$). The control interface caused the largest amount of average spread (2.75 mm), followed by fingerprint-corrected locations (1.24 mm). Locations corrected with the optical tracker interface had the lowest average spread (0.85 mm). This means that the spread of touch input after fingerprint-based correction was on average 2.2 times smaller than when uncorrected. On average, optical-tracker-based corrections brought down spread by a factor of 3.3 compared to the control interface.

Dot Targets vs. Crosshairs Targets

Average spread for each interface was 1.9 mm/1.4 mm/0.9 mm for the crosshairs conditions and 2.2 mm/1.9 mm/1.2 mm for the dot conditions. A two-way ANOVA, however, did not find a significant interaction between target type and interface

($F_{2,9} = 1.44, p = 0.287$). We did not find a main effect of target type on spread either ($F_{1,10} = 3.186, p = 0.105$). The fact that *dot* targets performed successfully as well, however, suggest that the methods we propose also apply to targets that the finger completely occludes during touch.

3.5.8 *Discussion*

This study supports our claim that touch devices can increase accuracy by exploiting the *generalized perceived input point model*.

Figure 3.11 shows another perspective on the results. It shows the minimum target sizes that users can acquire with 95% accuracy for each of the three interfaces. Sizes were computed so as to include 95% of all touches across participants and conditions (spread across all participants, sessions, and roll/pitch conditions plus two standard deviations).

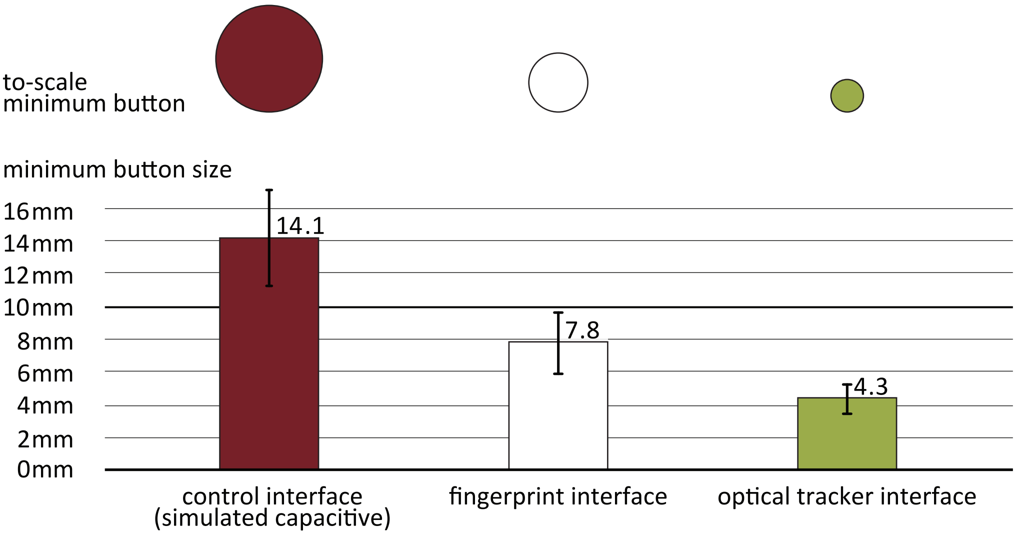


Figure 3.11: Minimum target sizes to achieve a 95% success rate. The circles are to-scale representations of the respective minimum target sizes. Error bars encode standard deviations.

The circles on top of Figure 3.11 illustrate the resulting buttons *to scale*. The FINGERPRINT interface achieves a minimum target size 1.8 times smaller than the CONTROL interface. The OPTICAL TRACKER interface reduces target size by a factor of 3.3. The resulting button would occupy less than 10% of the size of the CONTROL interface button.

3.6 CONCLUSIONS

In this chapter, we made two types of contribution.

On the one hand, we made a technical contribution. Our Ridgepad prototype achieved 1.8 times higher accuracy than simulated capacitive and we demonstrated that the use of high precision 3D tracking can more than triple touch accuracy. This substantial increase in accuracy can be used to make touch interfaces more reliable or to pack up to 10 times more controls into the same touch surface. Future versions of the optical tracking device might achieve a smaller footprint by using cameras placed in the corners of the screen. In Chapter 5, we present our prototype Fiberio, which extends Ridgepad's functionality onto a touch*screen*.

On the other hand, we made a contribution in the theoretical domain, which we tend to think of as at least equally important. We introduced a new model for touch inaccuracy, the *generalized perceived input point model*. We presented a user study, the findings of which are congruent with our new model, while they refute the *fat finger problem*, which was traditionally considered the primary source of touch inaccuracy.

This chapter also contributes a new perspective on touch. Touch has traditionally been considered a 2D phenomenon, most likely because touch screen interaction required only two coordinates, i. e., an x/y coordinate pair. The proposed model, in contrast, establishes touch as a phenomenon of not only the touch surface, but of a wider context of 3D factors. While we primarily investigated the user's finger posture in 3D, this wider context may include additional factors, such as head position, device orientation, parallax, and so on. Tracking these additional factors might allow future devices to realize even larger improvements in touch accuracy. Additional research is required here.

Finally, we learned about users. We found that users are not inaccurate—they are just different. The most likely explanation for this difference is that touch *on a millimeter scale* was never defined in the first place. For targets on this almost microscopic scale, pointing means to "dock" a comparably large, asymmetric object with a tilted surface. Comparing the arrangement of ovals across Figure 3.5 clearly shows that no two participants of our study had the same mental model of how to accomplish this.

4 UNDERSTANDING TOUCH: A NEW PERSPECTIVE

This chapter is based on results published in [68, 55].

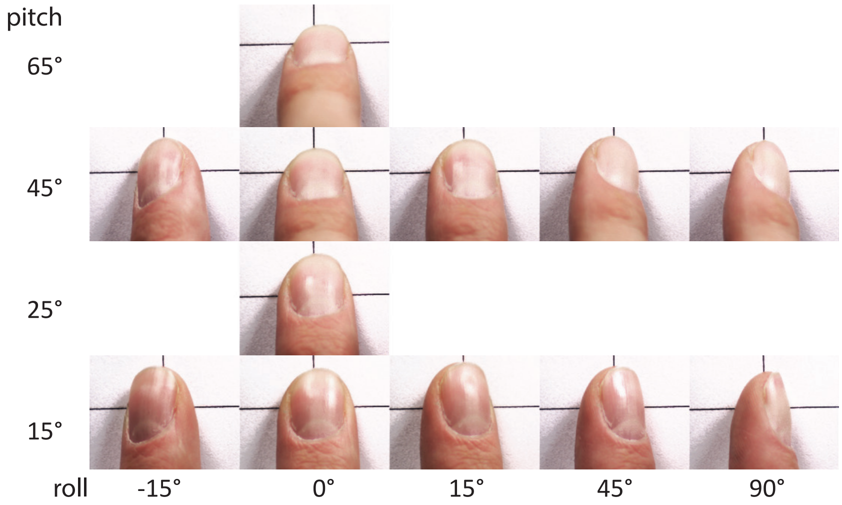


Figure 4.1: A study participant targeting crosshairs using different finger angles. Can you guess how this user is conceptualizing touch, i. e., what geometric relationship between finger and crosshairs the user is trying to maintain independent of how the finger is held? Our findings suggest that users indeed target as suggested by this illustration, i. e., by aligning finger features and outlines in a hypothesized top-down perspective.

Our findings in Chapter 3 indicate that current touch devices are subject to *systematic* error offsets because of their assumption that users acquire targets with the center of the contact area between finger and device. The existence of such systematic error offsets suggests that the assumption underlying the implementation of current devices is most likely wrong. In this chapter, we revisit this assumption.

In a series of three user studies, we find evidence that the features that users align with the target are *visual* features as shown in Figure 4.1. These features are located *on the top* of the user's fingers, not at the bottom, as assumed by traditional devices. We present the *projected center model*, under which error offsets drop to 1.6 mm, compared to 4 mm for the traditional model. This suggests that the new model is indeed a good approximation of how users conceptualize touch input.

The primary contribution of this chapter is to help understand touch—one of the key input technologies in human-computer interaction. At the same time, our findings inform the design of future touch input technology. They explain the inaccuracy of traditional touch devices as an effect of "parallax:" while users target by aligning features on the top of the finger with the target, devices sense using the contact are, which is a feature at the bottom side of the finger. We conclude that certain camera-based sensing technologies can inherently be more accurate than contact area-based sensing.

4.1 HOW DO USERS ACQUIRE SMALL TARGETS USING TOUCH INPUT?

As explained in Chapter 3, current touch technologies typically sense input locations by reducing the contact area between the user's finger and the device to its center. These devices are thereby based on the implicit assumption that the contact area encodes the information about the desired target in the first place, i. e., that *users* somehow use the contact area to encode which target they are trying to refer to, e. g., by touching the target with the center of the contact area.

Our findings in Chapter 3 seem to put this assumption into question. While the CONTACT AREA MODEL is clearly plausible on a macroscopic scale, our findings with very small targets have shown that touch input is subject to systematic error offsets, which cause as much as two thirds of the overall inaccuracy of touch as shown in Figure 3.7. When users target with an almost horizontal finger, for example, the target position measured by a capacitive touch device is off by as much as several millimeters (Figure 3.6)—on small screens devices, this is a substantial effect. In the studies we presented in Chapter 3, the size and direction of the error offset was affected by a range of parameters, including finger posture measured in roll, pitch, and yaw.

While we compensated for this effect using an elaborate scheme of corrective offsets in the previous chapter (user- and finger-posture specific position adjustments), the existence of these systematic offsets raises much deeper questions. In particular, the existence of these offsets seems to indicate that the assumption these devices are built on, i. e., that users target based on finger contact area, *is wrong*.

So if users do not touch targets based on contact area, how *do* they target? How do they decide whether a finger is "on target" or whether it requires corrective moves? We attempt to answer this question in this chapter.

4.2 understanding touch input at microscopic scales

In order to help us specify what we are trying to find out, Figure 4.2 illustrates the general concept of touch input: Users communicate a 2D target location to a touch device. As users acquire a target with their finger, such as the shown crosshairs, they effectively map the 2D target location into the 3D posture of their finger (a position in 3D space, i. e., 3D position and 3D rotation). Due to the lack of a better term, we will refer to this 2D-to-3D mapping as *the user's mental model of touch*, in the traditional sense of a user's mental model of the way an object operates [108], but with the "object" being the user's own finger.

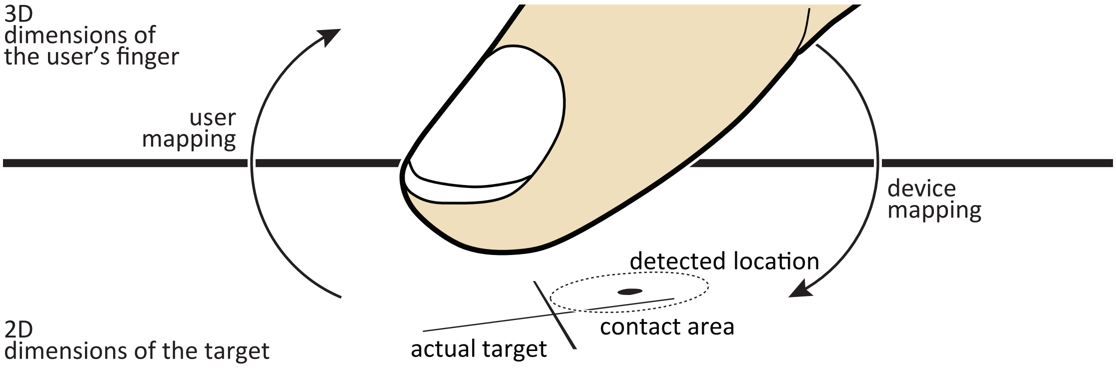


Figure 4.2: When acquiring a target, here marked with 2D crosshairs, users effectively map the 2D position of the target into a 3D position of their finger. Upon touch, current touch devices sense the contact area between the finger and the surface and reduce it to its 2D centroid to detect the input location. Devices thereby map the 3D of the user's finger back into the 2D coordinate system of the touchscreen.

The objective of any touch input device is to *invert this mapping*, i. e., to reconstruct the 2D target location from this 3D finger posture. We will refer to this 3D-to-2D mapping as *the device's conceptual model*. Perfect reconstruction is achieved if and only if the mapping implemented by the touch device is indeed inverse to the 2D-to-3D mapping implemented by the user, i. e., if the device's conceptual model matches the user's mental model.

Current touch devices implement this back-translation as illustrated by Figure 4.2: They observe the 2D contact area between finger and surface, and reduce it to its centroid, from which they infer the 2D input location. As explained earlier, however, our previous findings indicate that this contact area model is not correct, i. e., it *does not* reconstruct the intended input position accurately. Apparently, users do *not* aim by pointing using the center of the contact area.

Note: Throughout this chapter, we will use an apparatus similar to Figure 4.2, i. e., *crosshairs* marking the target. As discussed in Section 3.5, crosshairs performed indistinguishably from a dot target in our previous study, which suggests that the influence of crosshairs onto users' mental model of touch is reasonably small.

4.2.1 *An Example and Preview of Findings*

As a preview of our analysis, take a look at the sequence of images shown in Figure 4.1. They show a user (Participant 7 from User Study 1 in Section 4.5) targeting a pair of crosshairs with different finger angles. Looking across the sequence of images, we may already catch some indication of what mental model of touch this user adheres to. Certain geometric relationships between finger and crosshair seem to remain—independent of what finger posture the experimenter makes this participant assume.

The participant shown in Figure 4.1 is a particularly good representative of the new model of touch we propose, which we call the *projected center model*. This model says that users align certain *visual* features with the target. In the shown example, it is the horizontal center of the finger outline and the vertical center of the fingernail, which the user is aligning with the target.

We chose the specific viewpoint of this image sequence with intent: Even though the user's head was actually located to the bottom left of the picture during these trials, our findings suggest that users imagine this top-down perspective during touch input. Based on this perspective, they decide whether their finger is on target or whether it requires adjustment.

Under the PROJECTED CENTER MODEL, the error offsets of the CONTACT AREA MODEL effectively disappear (they drop to 1.6 mm, compared to 4 mm for traditional, contact area-based sensing), suggesting that the PROJECTED CENTER MODEL matches users' mental model of touch very closely.

The PROJECTED CENTER MODEL also explains why capacitive touch input devices are inaccurate: They implement the CONTACT AREA MODEL and thus infer input locations based on features located at the *bottom* of users' fingers. In contrast, users target based on features located *on the top/along the sides* of their fingers. The inaccuracy of touch on traditional touch devices is therefore an artifact of the parallax between the top and bottom of a finger.

4.2.2 *Approach*

In the remainder of this chapter, we present a series of studies that validate the reasoning outlined above.

Many models in HCI are created by measuring a feature and fitting a function to it. Unfortunately, we do not know yet what feature to measure or even what modality (sense of touch, vision, and so on). This forces us to take a more general approach to the problem [41]: (1) *Guess* the model, (2) compute the consequences, and (3) compare the computation to experiment/observation. If it disagrees with experiment it is wrong.

We apply these three steps as follows: (1) Before we attempt to guess mental models, we narrow down the search space. We conduct a series of interviews and then consider only the subset of models that are based on features mentioned by participants. (2) The consequences we predict are that models that match the user's mental model will feature error offsets approximating zero. (3) We conduct a series of pointing studies. We measure error offsets as the average distance between the sensed input location and the actual target location for the respective user (cf. offset in Chapter 3).

Since our primary goal is to understand touch, we require the remaining error offsets of a good candidate model to be small. Only when the remaining offsets get reasonably close to zero can we argue that the tested model indeed corresponds to the actual mental model of the respective user and thus contributes to an explanation of touch.

4.2.3 *Procedure*

We proceeded in four steps.

Step 1—Interviews: We interview users to learn how they (think they) target using touch input, i. e., what features they try to align with the target.

Step 2—Model creation: Based on participants' input, we create a set of 7×7 candidate models. User input inspires us to focus on models based on a top-down perspective.

Step 3—Filtering models: We conduct two touch pointing studies in which we determine error offset for all candidate models under different variations of finger and head postures. We eliminate models with large offsets, as they indicate a poor representation of participants' mental models. We keep 2×3 candidate models.

Step 4—Final evaluation: We conduct a final pointing study using the combined set of independent variables from the studies in Step 3 (finger and head position) to determine the error offsets and thus the "fit" of the remaining models.

4.2.4 *Contribution*

We make an attempt to understand the *underlying*, not directly observable mechanism of touch input, one of the key technologies in human computer interaction. In particular, we explain why current touch devices are inaccurate by challenging the common assumption that touch input is about the contact area.

Our findings inform the design of better touch input technology. They suggest that devices that observe the outline or "projection" of a finger have the potential to offer better touch precision than devices based on sensing the contact area between the user's finger and the device.

4.3 STEP 1: USER INTERVIEWS ON TARGET ACQUISITION STRATEGIES

The purpose of this study was to learn more about users' mental models by means of an interview. While users are known to have limited ability of rationalizing low-level activities, our goal was to create a selection of *potentially* relevant models and elements that could be used to form a list of *candidate* models. We did not worry about incorrect models at this stage, as we would eliminate these in subsequent steps of our process.

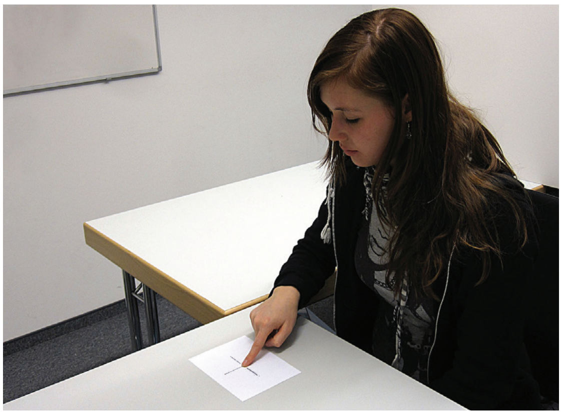


Figure 4.3: Before being interviewed, participants first acquired targets printed on a sheet of paper using various finger orientations. They then reflected on *how* they had acquired the target.

4.3.1 *Task and Procedure*

In order to help participants become aware of their mental models of touch input, they started by repeatedly acquiring a target. As shown in Figure 4.3, the target was marked by crosshairs drawn on a sheet of paper and each participant acquired it 50 times. Participants were instructed to place their finger such that it would "acquire a tiny button located at the center of the crosshairs if the paper were a touch screen."

To encourage participants to investigate their own mental models in more detail, participants acquired the target using the five finger postures shown in Figure 4.4, i. e., combinations of finger pitch and roll. All participants completed this part of the study in 10 minutes or less.

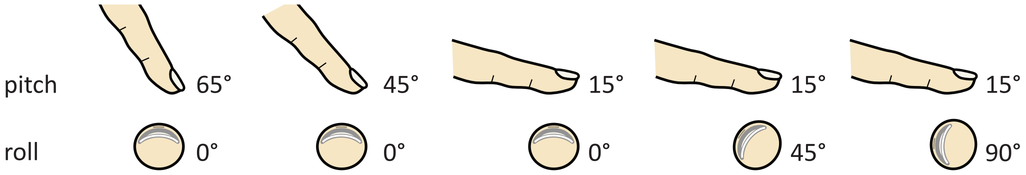


Figure 4.4: Finger postures. Participants assumed these five different combinations of finger pitch and roll, and then acquired the target during each trial.

After they had completed the trials, we interviewed participants about the strategies they had used to acquire the target and based on what criteria they had decided when their fingers were "on target." We were careful not to use any terms that could bias their answer, such as "contact area," "fingertip," "finger nail," and the like. If they, however, did mention "contact area," we asked participants to draw the contact area in four figures showing stylized fingers top-down held at four different angles.

4.3.2 *Participants*

We recruited 30 participants (5 female) from our institution. All participants were between 19 and 29 years old.

4.3.3 *Results*

When asked to verbalize their "targeting procedure," most participants hesitated. Four participants insisted that they could not explain their behavior and "just touched the target intuitively without giving it too much thought."

Six participants stated right away that their experience with mobile touch-screen devices had shaped their input behavior. While two of them understood how such devices determine input coordinates, they all stated to aim based on experience with the device. They all said that the device had "taught" them how to touch small buttons over time.

Contact Area

26 of the 30 participants said that they considered the contact area to be relevant to their targeting strategy. 24 of them stated that they imagined the contact area between their finger and the crosshairs and centered it on the target during the trials.

One said:

> *I could not see the contact area, so I imagined where it should be located. Then I chose the center of it and positioned it on the target. That's all.*

Participants' drawings of contact areas in the four figures supported our assumption that they cannot fully rationalize their behavior. Most drawings largely clashed with reality; while for a finger at a flat angle (15° pitch) the contact area extends fairly far towards the user's palm, participants always drew it too small. Similarly, for a rolled finger (90°), participants drew the contact area too large and mostly centered inside the finger, whereas in fact it is mostly offset horizontally and rather short.

Three participants mentioned a special version of contact area; they claimed to touch the target with the part of the finger that "comes down first." Five other participants explained that they placed their finger such that it applied the maximum amount of pressure to the target.

Visual Feedback

13 participants reported that they positioned their finger using *visual control*. They stated that they mentally connected the crosshairs under their finger and tried to move their finger such that the target was always located under the same position inside their finger.

One said:

> *I can see my fingertip and imagine where the crosshairs intersect. So I can visualize the bottom of my finger and always position it at the same location.*

Nine participants explained that they positioned the finger such that the target was located at a certain distance from the edge of their fingernail. Four other participants said they imagined a virtual 3D point inside their finger, which they repeatedly sought to position directly above the target. Two other participants said that they "projected" a feature in their finger down to the table and then aligned it with the target. This is an interesting observation, because such a projection is a comparably complex 3D operation that requires users to take head parallax into account.

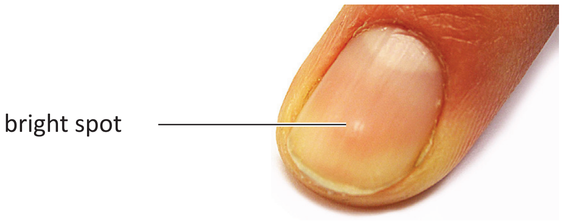


Figure 4.5: One participant targeted by placing a bright spot on his nail over the target.

Two participants described their targeting strategy as "cheating." Both had a visible spot on their fingernail as shown in Figure 4.5. Both said that they used the feature to be more accurate and vertically aligned it with the target whenever finger roll was 0°. For roll different from 0°, they stated that they still aligned the spot on their nail with the horizontal line of the crosshairs.

4.3.4 *Discussion*

While the study allows proposing a wide range of possible models that explain how participants targeted, such as models based on a camera that tracks with the user's head, we decided to make an educated *guess* and limit our search for candidate models to those based on features that users can perceive *visually* and *from directly above*. This seemed plausible given that thirteen participants mentioned visual features and six of them mentioned some sort of vertical projection.

As discussed earlier when stating our 3-step approach, whether or not our intuition was right would have to be determined in the following pointing studies. If the model should perform poorly, our guess would turn out to be wrong and we would have to come back and restart the process with another model (note how this is different from phenomena that lend themselves to direct observation, in which case the interviews themselves would have already answered the question).

4.4 step 2: picking candidate models

We constructed the following two families of candidate models: (1) contact area, which had produced a good fit for one user in our previous study in Chapter 3, Section 3.3, shown as Participant 3 in Figure 3.5 and (2) models based on features of human fingers that are visible from above. We implemented this by tracking users' fingers using a camera placed *directly above* the crosshairs on the touchpad.

Figure 4.6 shows a series of features that we found to be visible from above. We classified them as *horizontal features* if they might help determine the finger's horizontal position and *vertical features* if they might help determine the finger's vertical position; some features, such as the corner of the fingernail are both (e. g., nail left and nail groove). Note the 3d nature of the finger, which causes features, such as *outline center* to refer to the outline of the skin for some levels of finger roll and to the outline of the nail for other levels of finger roll.

In theory, users' mental models might combine any number of features in arbitrarily complex ways. We felt, however, that the effortlessness of touch pointing suggests that only simple models are truly plausible. We therefore only included models that

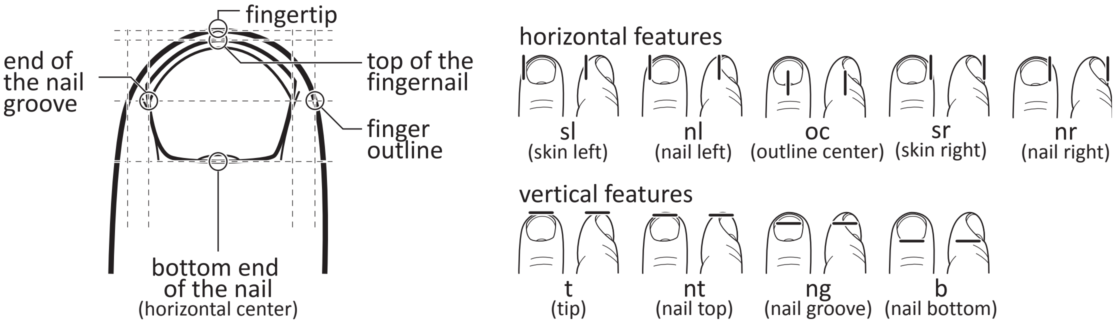


Figure 4.6: We found these *horizontal* and *vertical features* to be detectable on a human finger. We used them to construct candidate models that predict input locations.

use a single feature (such as NR for the mental model of users who aim relative to the right edge of their nail) and models that refer to the center point between two features, such as SL | SR for the mental model of users who aim relative to the center point of the finger's outline, i. e., between "*skin left*" and "*skin right*."

We will refer to terms such as NR or SL | SR as *half models*, because it takes two of them to describe the mental model of a user—one for x and one for y. To avoid the overhead of evaluating the cross product of *horizontal* and *vertical features*, however, we keep these two classes of features and models separate throughout most of this chapter and will not combine them until the final study.

Figure 4.7 lists the horizontal and vertical half models that we created from the respective features shown in Figure 4.6.

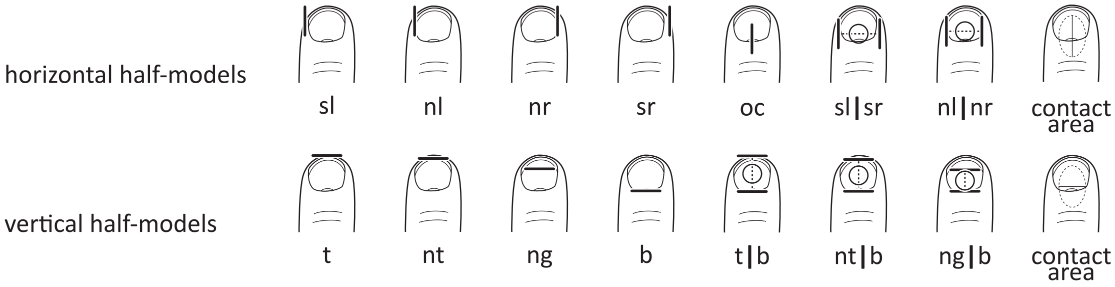


Figure 4.7: (a) 7 horizontal half models based on the features in Figure 4.6, plus the CONTACT AREA MODEL. (b) 7 vertical half models (plus the CONTACT AREA MODEL).

The idea behind a half model, such as "nail right" was not necessarily that participants would place this specific point over the target, but *some* point that is located at an offset from this feature. To include this concept, we complemented all half models with a single user-specific *offset* (unlike our previous approach in Chapter 3, which allowed for one offset *per finger angle*).

4.5 step 3: eliminating models

Next we eliminated those (half) models that did not match the mental models of any users. In order to do so, we conducted a touch pointing study. Using a camera above the target, we recorded participants as they repeatedly acquired a target on a capacitive touch pad. We then tried to "explain" the observed data using each of our half models and eliminated all half models that did not fit any participants.

To keep the overall number of repetitions manageable, we broke this study down into two individual studies. The first study tested twelve combinations of roll and pitch; the second tested only four combinations of roll and pitch, but varied head position in addition.

4.6 step 3a: eliminating models using roll and pitch

4.6.1 *Task*

Participants repeatedly acquired a crosshair target located on a touchpad (Figure 4.8). During each trial, participants first touched a 1″×1″ "start" button located 2″ left of the target. Participants then assumed the finger angle for the current condition with their right index finger and acquired the target. Participants committed the touch interaction by pressing a foot switch. This recorded the touch location reported by the touchpad, triggered the camera to take a picture, played a confirmation sound, and completed the trial. Participants did not receive any feedback about the location registered by the touchpad.

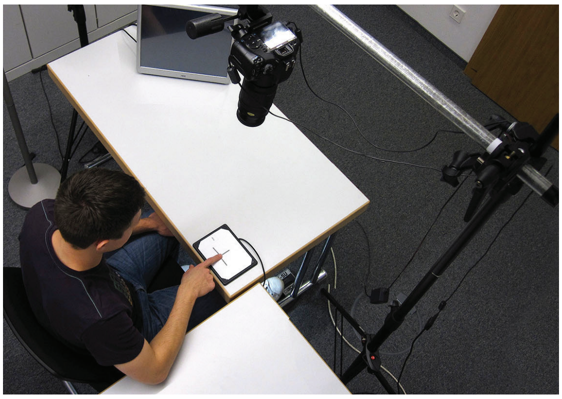


Figure 4.8: Participants acquired a crosshair target located on a touchpad and committed input using the foot switch. The overhead camera recorded participants' fingers upon touch.

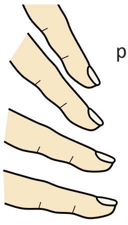
We took the following four measures to minimize the impact of other potential factors. First, participants kept their head in a fixed position above the touchpad, as shown in Figure 4.8. This controlled for parallax. Second, the crosshairs marking the target extended beyond participants' fingers, allowing participants to maintain a certain amount of visual control during targeting. Third, the use of a foot switch to commit input allowed us to avoid artifacts common with other commit methods, such as inadvertent motion during take-off. Fourth, participants rested their elbow on the adjacent table (Figure 4.8) to preclude fatigue. Finally, participants were told to use as much time as necessary and that task time would not be recorded.

Note that there was no need to include distracter targets. Distracters have a major effect on *adaptive* input techniques, but not on unmodified touch.

The purpose of using crosshairs was to reduce noise, thus helping us observe the underlying mental models more clearly. While the use of visible crosshairs may in theory impact participants' targeting behavior, we did not observe any such effect in our studies.

4.6.2 *Independent Variable: Finger Posture*

As shown in Table 4.1, we used the same combinations of finger pitch and finger roll as we did in the previous chapter. However, we dropped the 90° pitch condition, because the camera located directly above the pad could not capture the participant's nail in this condition (Figure 4.8).



roll pitch	-15°	0°	15°	45°	90°
65°		x			
45°	x	x	x	x	x
25°		x			
15°	x	x	x	x	x

Table 4.1: Study conditions. Participants assumed these combinations of finger pitch and finger roll rotations during the study and then acquired the target on the touchpad.

4.6.3 *Procedure*

Participants performed a sequence of 12 angles × 2 repetitions totaling 24 trials for each participant. The order of pitch-roll combinations was counterbalanced across participants. After completing the trials, participants filled out a post-study questionnaire. All participants completed the study in 15 minutes or less.

4.6.4 *Apparatus*

Our study apparatus recorded the contact area of each touch using a capacitive touch pad and captured a picture of the participant's finger using an overhead camera. The capacitive pad was a 6.5″×4.9″ *FingerWorks iGesture* pad; the camera was a *Canon EOS 450D*, capturing participants' fingers at 140 dpi. Participants committed trials using a *Boss FS-5U* foot switch. All components were connected to an Intel Core 2 Duo machine running Windows XP.

4.6.5 *Participants*

We recruited a new set of 30 participants (10 female) from places around our institution. Participants were students from a range of different disciplines and were between 20 and 32 years old. We offered a € 20 incentive for the most accurate participant.

4.6.6 *Data preparation*

We manually annotated the visual features in all pictures that were taken by the camera. During a pilot study, we attached markers to participants' fingers in order to allow for automated tracking. However, participants mentioned that the markers distracted them. Since some participants had started to include them as features into their targeting model, we decided to drop the markers in favor of manual annotation.

4.6.7 *Results*

Vertical half models

Figure 4.9 (left) shows which vertical half models produced the best fit (i. e., the lowest error offsets) for how many participants. We consider a half model to produce the best fit if the systematic offsets produced by this half model for each condition have the smallest distance from the center of mass of all error offsets produced by that model, across all half models.

No single model offers the best fit for all users, suggesting that different users may have *different* mental models. The half model T | B, i. e., the vertical center of the fingernail performs best here—it offered the best fit for 9 participants. It is followed by NG | B, a slightly different version of the vertical center of the fingernail, with another

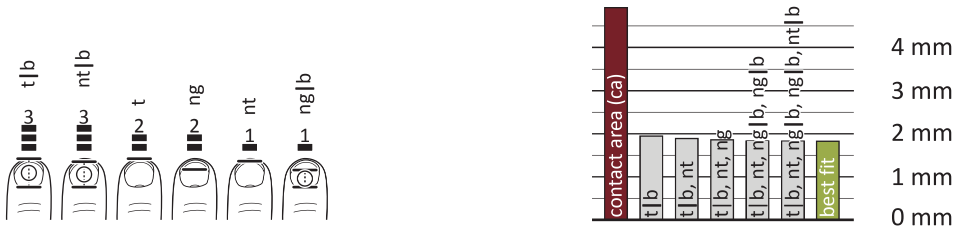


Figure 4.9: Left: Number of participants for which each vertical half model produced the lowest error. Right: Error produced when using only a subset of the half models to analyze participants. Switching from the model CONTACT AREA to T|B reduces error offsets to 44%; using all models listed on the left side reduces the error to 37% compared to the CONTACT AREA MODEL.

6 participants. As expected based on Figure 3.5, the CONTACT AREA MODEL offers the best fit for a very small number of participants, here only 1 out of 30.

Figure 4.9 (right) shows how well different subsets of vertical half models combined fit the data. Bars represent the average vertical error offset if participants' data is processed using only the respective half models. The red bar on the left represents the error offsets produced by the capacitive baseline condition CONTACT AREA.

The green bar on the right in Figure 4.9 (right) represents the error offsets produced if every participant's input were processed using their personal best-fit model from Figure 4.9 (left); the switch to the best fit model reduces the error offsets by about 63%. We will refer to this best-fit case as "*per-participant models*."

Some of the half models listed as best fit model are similar. As a result it may not be necessary to maintain all of them. The gray bars in the middle of Figure 4.9 (right) show how error rate increases if we drop some of the half models. We see that dropping all but three models (T|B, NT and NG|B) incurs a penalty of only 5% compared to using all vertical half models. Dropping all models but T|B incurs a penalty of 18.5% over the best-fit case. The CONTACT AREA MODEL alone, however, leads to large error offsets (averaging 3.75 mm across participants).

Horizontal half models

Figure 4.10 (left) shows the corresponding data for the horizontal half models. The half model OC (center of the finger outline) produced the lowest error for over a third of the participants (11 of 30). The CONTACT AREA MODEL offered the best fit for 5 of the 30 participants.

The benefit of using per-participant models is only a factor of 1.4 (Figure 4.10 right) and, thus, by far not as large as in the vertical case . This is a result that we expected based on the results in the previous chapter, because of the smaller horizontal extent

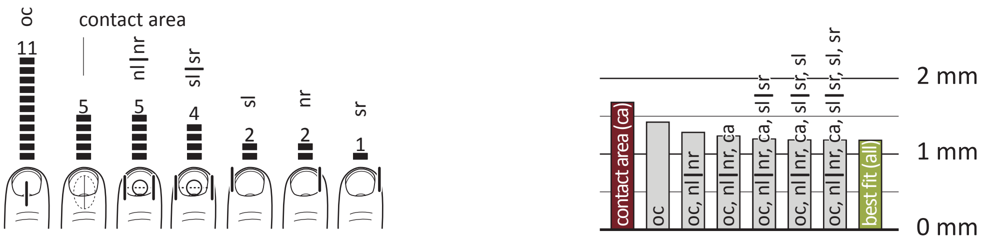


Figure 4.10: Left: Number of participants for which each horizontal half model produced the lowest-error offsets. Right: Using per-participant half models reduces the error offsets on average to 84% compared to using merely the contact area model.

of clusters in Figure 3.5. Horizontal error has always been smaller, thus there is less potential for improvement.

4.6.8 *Discussion*

These findings suggest that the horizontal half model nl (Figure 4.10 left) and the vertical half model ng (Figure 4.9 left) can be eliminated, as they did not produce a best fit for any participant. However, we will postpone the decision until we have seen results of the next study.

4.7 step 3b: eliminating models: head parallax

In contrast to the study in Step 3a, we added head parallax as an independent variable in this study. Again, the purpose of this study was to eliminate candidate half models.

4.7.1 *Task*

The task was the same as in the previous study, except that in addition to assuming a specific finger posture, participants also assumed one out of four predefined head positions shown in Figure 4.11.

4.7.2 *Procedure*

Each participant completed a sequence of 2 pitch angles (15° and 45°, see Table 4.1) × 2 roll angles (0° and 45°) × 4 head positions (Figure 4.11) × 2 repetitions = 32 trials, in four blocks, one for each head position. Finger angles as well as head positions were

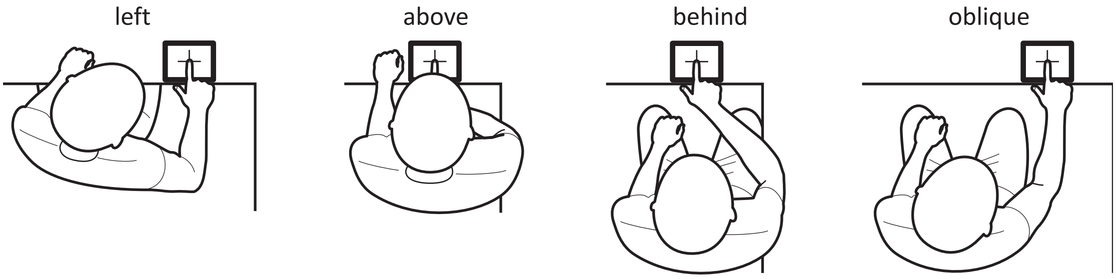


Figure 4.11: Participants acquired the target while assuming one of these four different head positions.

counterbalanced across participants. Participants filled out a post-study questionnaire. All participants completed the study in about 20 minutes.

4.7.3 *Participants*

We recruited a fresh set of 12 participants (5 female). All participants were between 19 and 24 years old. Similar to the previous study, we encouraged participants to be accurate throughout all conditions. Again, we offered a €20 incentive for the most accurate participant.

4.7.4 *Apparatus*

The apparatus was the same as in the Study 3a (Figure 4.8).

4.7.5 *Results*

Vertical half models

As in the previous study, the half models describing the vertical center of the fingernail together (T | B and NT | B) produced the best fit for half of all participants (6 of 12, Figure 4.12 left). The CONTACT AREA MODEL, in contrast, never produced the lowest-error offsets for any participant in this study.

Again, we can reduce the number of half models without sacrificing too much precision; keeping only the vertical model T | B incurs a penalty of 6.5% over using all vertical half models. It still reduces the error of the CONTACT AREA MODEL by a factor of 2.5.

As shown in Figure 4.12 (right), the use of per-participant best-fit half models reduced the error offsets to 1.8 mm, from 5 mm of the CONTACT AREA MODEL to 40% of that value.

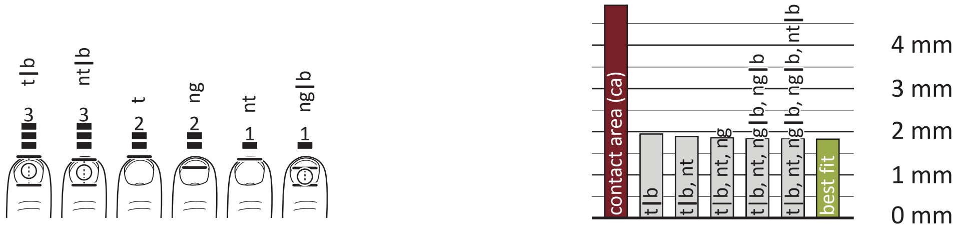


Figure 4.12: Left: Number of participants for which each vertical half model produced the lowest error. Right: The mean error of input drops to 39% when using T|B instead of CONTACT AREA.

The CONTACT AREA MODEL incurs even bigger error offsets than in the previous study (5 mm compared to 3.75 mm). This suggests that the CONTACT AREA MODEL is sensitive to changes in head parallax, but more data is required to know for sure.

Horizontal Half models

As in the previous study, the horizontal half model OC produced the best fit for the largest number of participants (7 of 12, see Figure 4.13 left). Compared to the previous study, the CONTACT AREA HALF MODEL produced the best fit for a similar fraction of all participants (here 1 of 12, compared to 5 of 30).

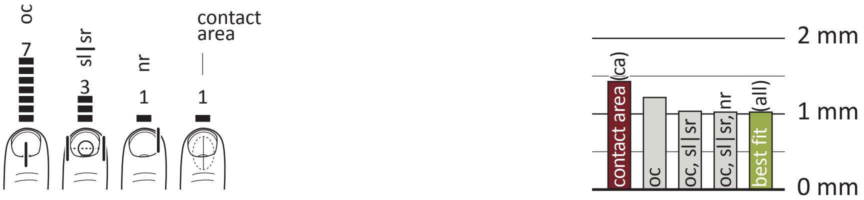


Figure 4.13: Left: Number of times each of the horizontal models produced the lowest-error offsets per participant. Right: The average error produced by the horizontal best-fit models sinks to 72% compared to CONTACT AREA. The three horizontal half models OC, SL|SR, and T|B alone account for this improvement.

As in the previous study, the use of per-participant half models produced only a moderate reduction of error offsets over the CONTACT AREA MODEL (about 15%).

4.7.6 *Discussion*

Figure 4.14 lists the remaining candidate half models. We obtained this list by eliminating all models that did not produce at least one best fit in either one of the two studies. In addition, we eliminated all models whose addition would have decreased the error only marginally—the benefits of including NR, NG | B, and NT | B in the last study, for example, were less than 1%. However, we did maintain the two halves of the CONTACT AREA MODEL as well as the model T (absolute distance from the fingertip, which also only reduced error by less than 1%): both have been implemented in products and related work, so we wanted to see how they perform in the final study.

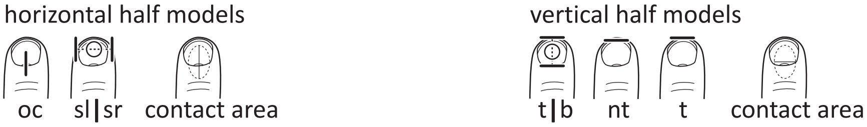


Figure 4.14: The remaining candidate models after elimination.

If we take a closer look at the three remaining horizontal half models, we notice that all of them are versions of the center of the finger outline, only sampled at different locations, i. e., at the bottom of the nail (OC), at the location of the nail grooves (SL | SR), and at the horizontal center of the CONTACT AREA.

The three remaining vertical half models describe the target location either in relation to the fingernail (T | B) or as an offset from the top of the fingertip (T and NT).

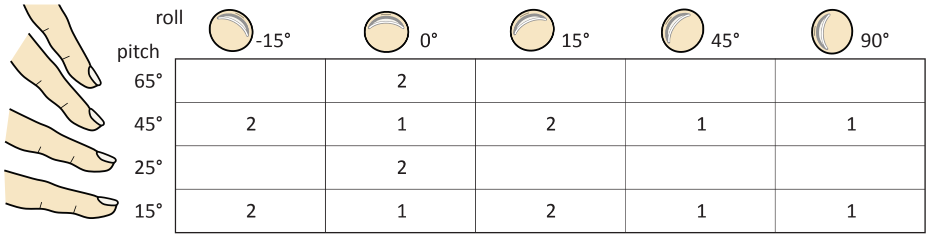
4.8 STEP 4: EVALUATING THE REMAINING MODELS

The purpose of this final study was to evaluate the remaining half models. We could not re-use the data from the previous studies, because we had already used this data for learning and eliminating the very same half models. More importantly, though, participants had performed only few trials per condition; thus the data did not allow us to distinguish error offsets from random noise (i. e., spread by the fat finger problem).

In this final study, we addressed this by increasing the number of repetitions to four trials per condition and extending the study to two blocks. Aggregating these eight trials substantially reduced fat finger noise and thus revealed the systematic error offsets we were looking for more clearly. These offsets provided us with a more reliable estimate sense of how far a change in mental model could reduce offsets and thus how closely the respective models were actually matching participants' mental models of touch.

4.8.1 *Task*

The task was the same as in the previous study (Section 4.7); we included all 12 levels of finger pitch and roll from Study 3a and all head positions. To keep the number of repetitions per participant manageable, we subdivided the roll/pitch variables between subjects, as shown in Table 4.2.



pitch \ roll	-15°	0°	15°	45°	90°
65°		2			
45°	2	1	2	1	1
25°		2			
15°	2	1	2	1	1

Table 4.2: To keep the number of trials per participant manageable, we ran roll/pitch between subjects. The table shows the assignment of conditions to the two groups.

4.8.2 *Study design*

Each participant completed 6 combinations of finger angles (Table 4.2) × 4 head positions (Figure 4.11) × 2 blocks × 4 repetitions = 192 trials. All participants completed the study in 40 minutes or less. Participants filled out a questionnaire afterwards.

4.8.3 *Apparatus*

The apparatus was the same as in Study 3b (Figure 4.8).

4.8.4 *Participants*

We recruited a fresh set of 12 participants (6 female) from places around our institution. All participants were between 21 and 32 years old. As in the previous studies, we encouraged participants to be accurate and offered a € 20 incentive for the most precise participant.

4.8.5 *Results*

The final study did not produce anything unexpected as the performance of the participating half models was similar to the previous two studies. As shown in Figure 4.15 left, both oc and sl | sr produced the best fit for 5 of the 12 participants; t and t | b produced the best fit for 5 participants each. While the use of all three horizontal half models yields only 15% less error in offsets (per-participant models), vertically, per-participant models reduce the error offsets substantially by 60%. t | b alone reduces error offsets by a factor of 2.5 compared to contact area (Figure 4.15 right).

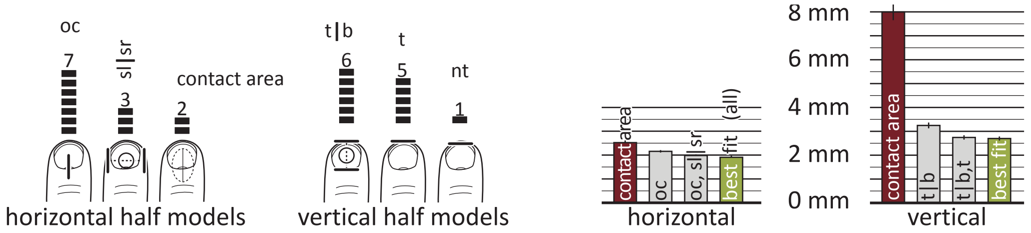


Figure 4.15: Left: Number of times each of the horizontal and vertical models produced the lowest error, respectively. Right: Using the best-fit horizontal half models instead of contact area reduces the error by 15%. Vertical per-participant models reduce the error produced by contact area by 60%.

4.8.6 *Merging Models*

Finally, we rejoined half models into full models. Table 4.3 shows which half models went together well: The combination oc & Tb produced the best overall fit for 4 of the 12 participants, followed by oc & t, sl | sr & t | b, and contact area & t | b for another 2 participants each.

vertical half models \ horizontal half models	oc	sl | sr	contact area
t | b	4	2	2
t	2	1	
nt	1		
contact area			

Table 4.3: Number of times a combination of half models together produced the best fit and the lowest overall input error for a participant.

Figure 4.16 shows error offsets for the eight full models from Table 4.3. A one-way ANOVA with participant as a random variable found a significant main effect of MODEL on error offsets ($F_7 = 38.662, p < 0.001$). Post-hoc t-tests using Bonferroni correction found that all six other models and the per-participant best-fit aggregate produced significantly lower error offsets than CONTACT AREA (all $p < 0.003$).

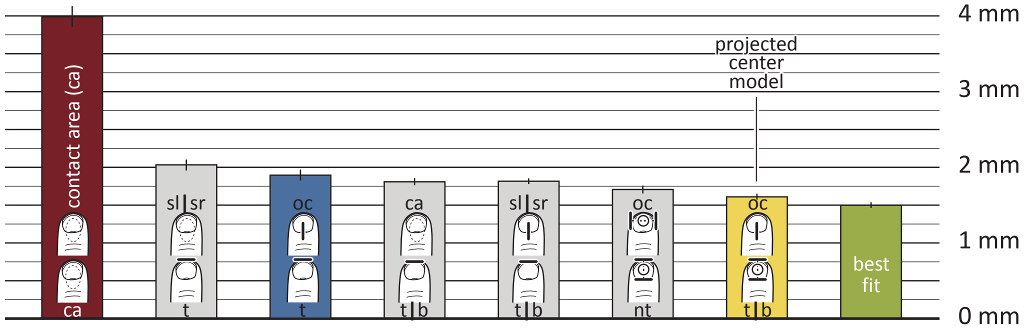


Figure 4.16: Size of the remaining offset for the combined full models from Table 4.3 compared to the CONTACT AREA MODEL (± std. error of the mean).

The best individual model was OC & T|B. It says that participants target by placing the horizontal center of their finger outline and the vertical center of fingernail over the target. In the beginning of this chapter, we already referred to this model using the name *projected center model* and the images shown in Figure 4.1 are best explained using this model.

In summary, the PROJECTED CENTER MODEL performed best out of all the models tested. Under the PROJECTED CENTER MODEL, the large systematic offsets of 4 mm observed by the CONTACT AREA MODEL shrink down to 1.6 mm, an improvement by a factor of 2.5. At the same time, the remaining offsets are close enough to zero to suggest that this model approximates participants' mental model indeed well.

4.9 APPLYING THE RESULTS TO MAKE MORE ACCURATE TOUCH DEVICES

In order to incorporate the findings from the previous studies into a touch device that senses input with high precision, we analyze which models they need to implement to redeem the effects we observed.

4.9.1 *Adding Error Spread to Infer Minimum Button Sizes*

So far, we have discussed models based on systematic offsets, as it is a good metric for testing the quality of mental models. To answer questions about device performance, we add the other error variable, i. e., spread, back in. The resulting *minimum button sizes* for 95% reliable touch input are shown in Figure 4.17. These values specify how accurately a device based on the respective model can be expected to perform.

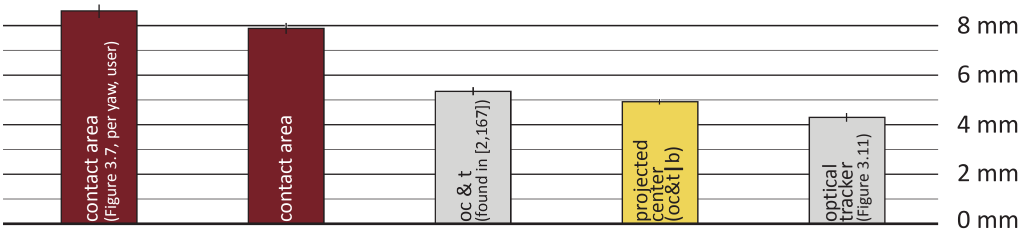


Figure 4.17: Results from this study in terms of minimum button sizes. We juxtapose our results to the results of our studies that we reported in Chapter 3.5.

The chart shows that devices based on the PROJECTED CENTER MODEL allow users to acquire targets of 4.9 mm with 95% accuracy, compared to 7.9 mm target size for the corresponding CONTACT AREA MODEL (8.6 mm in Figure 3.5, corrected for yaw to match conditions in this study). In terms of target surface, this difference amounts to a factor of 2.6. This means that a device implementing the PROJECTED CENTER MODEL could pack 2.6 times more targets into the same screen space or, alternatively, that a device could be reduced to less than half its size and still allow users to operate it reliably.

For reference, in Figure 4.17 we also included the 4.3 mm minimum target size that we previously achieved by attaching retro-reflective markers to the user's finger and tracking it using an optical tracking system as described in Section 3.5. Based on 600 repetitions of training data, it removes all known offsets, so that this model can be considered a current lower bound for touch accuracy. In comparison, the 4.9 mm minimum button size of the *calibration-free* PROJECTED CENTER MODEL gets surprisingly close.

4.10 HIGH-PRECISION TOUCH INPUT USING CAMERA-BASED TOUCH DETECTION

In order to implement the PROJECTED CENTER MODEL, a device needs to be able to reliably locate a user's fingernail, which is technically challenging.

Figure 4.17 points out an alternative. At a minimum button size of 5.35 mm, the model OC & T does not quite reach the 4.9 mm of the PROJECTED CENTER MODEL. However, it is comparably easy to manufacture, as this approach only requires locating finger outlines in a camera positioned above the target, namely the outlines of the sides and the top of the finger.

Touch detection using an overhead camera has already been explored in a number of research prototypes, such as CSlate [2] or LucidTouch [165]. Our findings suggest that this track of engineering, combined with sufficiently accurate cameras, could be a more promising approach to high-precision touch sensing than the currently more widely available devices based on sensing the contact area.

We present one such prototype that infers touch events using an overhead camera. Although designed for a different purpose than input precision, the *Imaginary Phone* implements the findings presented in this chapter by inferring input on the user's own body using the visual model oc & t.

Imaginary Phone: Shortcut Touch Interaction Using a Wearable Camera

The Imaginary Phone is a device that senses touch input on the user's body using an overhead camera. The device thereby implements a visual model to derive input locations.

The purpose of Imaginary Phone is not input precision but shortcut interaction with mobile devices users carry in their pockets as shown in Figure 4.18, here an Apple iPhone. Users interact by mimicking the use of the physical phone using touch input on their own empty hand. The Imaginary Phone tracks all touch interaction and forwards touch events wirelessly to the physical mobile phone, where they invoke the corresponding actions. The physical device supplies feedback to operations via the built-in speaker or a wireless headset worn by the user.

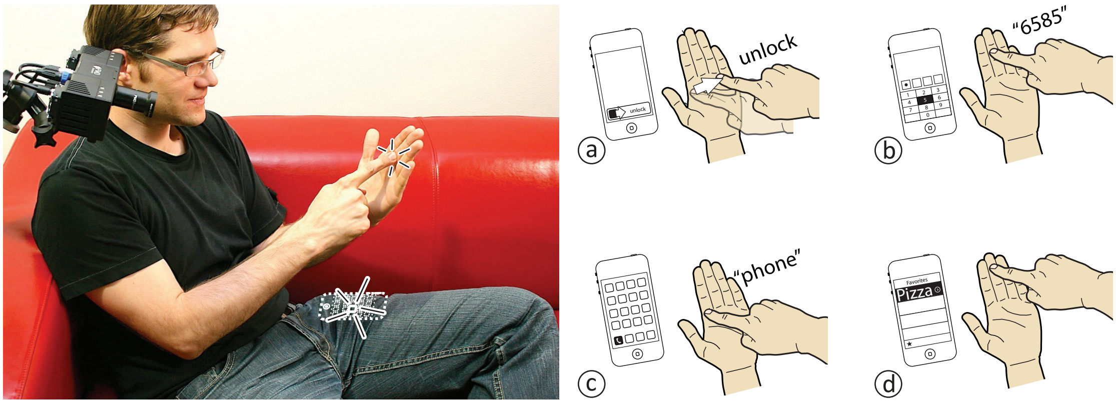


Figure 4.18: The Imaginary Phone detects touch input on the body using an overhead camera and implementing a *visual* model for touch detection. Left: This user operates his iPhone in his pocket through touch input on his palm, which can be used in place of the physical phone. The depth camera tracks all touch interaction on the body and sends input events to the physical device where it triggers the corresponding function. Right: Example scenario of making a call with the Imaginary Phone: (a) unlock with a swipe, (b) enter your pin, (c) select the 'phone' function and (d) select the first entry from the speed dial list.

Our prototype allows the user to perform everyday tasks, such as picking up a phone call or launching the timer app and setting an alarm. Imaginary Phone thereby serves as a shortcut that frees users from the necessity of retrieving the actual physical device, for example in situations where the physical phone is out of reach.

4.10.1 *Sensing Hardware*

Imaginary Phone uses a depth-sensing camera to observe the user's hands during interaction as shown in Figure 4.18 (left). The camera currently looks over the user's shoulder, but future versions will be wearable and attached to the user's shirt. We use a time-of-flight depth camera to implement the Imaginary Phone, which allows it to work indoors as well as outdoors and thus support mobile use. The camera is a PMD[vision] CamCube that provides frames at 40 Hz with 200 × 200 px resolution as shown in Figure 4.19a.

4.10.2 *Algorithm*

In order to extract the two hands from the input image, we pre-process the raw depth image as shown in Figure 4.19a. We first find the closest pixels in the depth image (b), remove all pixels with relative depth values of more than 30 cm, and smooth all remaining values. To determine the number of visible hands, we create a depth histogram of the masked image (c) and calculate the number of strong peaks (indicated by green squares in Figure 4.19c). Based on the two distributions in the histogram, we classify pixels in the depth image (d) to obtain the masks for the pointing hand (e) and the reference hand's palm (f).

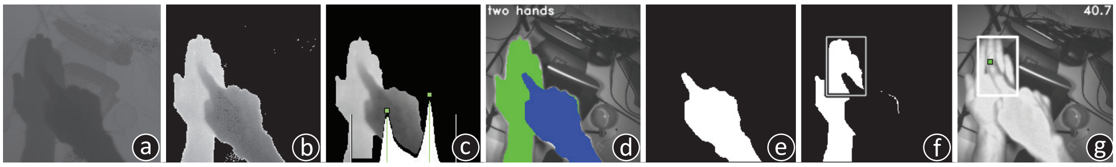


Figure 4.19: (a) In processing the raw depth image, our system (b) thresholds the image and (c) calculates a depth histogram to (d) segment the image into two masks: (e) pointing hand and (f) reference hand. From that we calculate (g) the final touch position and reference frame. This location is then forwarded to the physical device.

To determine if and where the user is touching the palm, we pick a location inside the pointing hand (Figure 4.19e) and fill using a small tolerance value, eventually "walking down" the finger towards the reference hand (f). If the fill does not reach a depth value that belongs to the reference hand while staying within the tolerance value, we infer no touch. If it does, we infer that the finger is touching.

To determine the location of the touch event, we look for the tip of the user's finger as described by the model OC & T. Due to the limited resolution of the depth camera, however, the previous flood-fill operation does not precisely yield the location of the fingertip. We therefore derive that location on the user's finger that is sufficiently different from the depth values of the reference hand. In real-world coordinates, this amounts to a difference in depth of about 2 cm. Since the location on the finger that is 2 cm closer to the camera is "farther up" the user's finger, we determine the final touch location by adding a small vector in the direction of the finger (green square in Figure 4.19g). During our tests, this procedure obtained the position of the user's fingertip with sufficient accuracy to target home-screen buttons reliably.

To obtain a frame of reference for all touch-input events, we use a bounding box around the the palm's fingers excluding the thumb. We first calculate the width from the top 3 cm of the hand to exclude the thumb (the depth values allow translating this into pixels measurements) and draw a frame around those values. We then set the height of this reference frame to match an aspect ratio of 1.5, which corresponds to the aspect ratio of an iPhone 4. The final frame is shown in Figure 4.19f and g. As this reference frame is subject to noise if the pointing hand is present, we update the reference frame only if one hand is visible and, upon sensing both hands, adapt it by tracking the reference hand.

As the computed raw locations are subject to strong noise, we use hysteresis to maintain touch states (touch/no touch) and smooth input coordinates, which enables smooth dragging or even free-form drawing. This also prevents processing inadvertent input, such as a hand waving by the camera. Our system supports all of the same single-touch interactions that are possible on the phone: swiping, scrolling, tapping, dragging, drawing, and the like.

After determining the touch position on the palm, our prototype relays touch input to an iPhone. A custom-written input daemon on the iPhone receives the smoothed events via TUIO over WiFi and injects them into the event stream of the iPhone. The VoiceOver accessibility mode built into Apple iOS 4.0 and greater provides auditory confirmation of actions. The built-in unlock gesture on the iPhone, designed to prevent inadvertent touch input, additionally helps our system to disregard spurious input through gestures that happen naturally when not using the system.

In summary, the Imaginary Phone implements touch interaction on the body using an overhead camera. Our system thereby implements a visual model to infer touch locations and forwards them to the actual device to invoke commands.

4.11 CONCLUSIONS

In this chapter, we conducted an exploration of users' mental models of touch. The fact that under the proposed PROJECTED CENTER MODEL the error offsets found by our work in Chapter 3 essentially disappear suggests that this model is likely to closely match how users proceed while acquiring a target on a touch device. Our findings suggest that systems that track fingers using cameras from above have the potential for substantially better pointing accuracy than CONTACT-AREA-based sensing as currently implemented.

In order to translate the results from this chapter into prototypical implementations for high-accuracy touch sensing, we have two options. Either we switch to an overhead perspective and implement touch recognition based on visual features using a camera, such as the Imaginary Phone in the previous section in conjunction with a high-resolution camera; or we maintain contact-based touch recognition as implemented on all of today's mobile devices, reconstruct the user's finger in 3D from the 2D touch contact, and use that to make touch accurate.

The devices we presented in previous chapters accomplished 3D reconstruction, yet they were input-only. In the next chapter, we present a fully interactive touch*screen* that reconstructs 3D information from all touch contacts while providing output on the same surface.

5 FIBERIO: A TOUCHSCREEN THAT SENSES FINGERPRINTS

This chapter is based on results published in [70, 129].

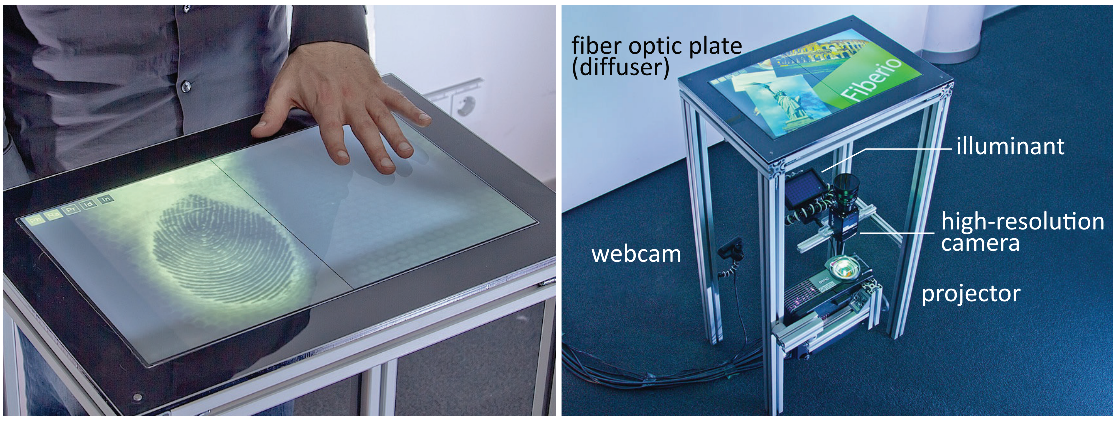


Figure 5.1: Fiberio is a rear-projected tabletop system that captures users fingerprints during touch. Left: Fiberio is displaying a region of its high-resolution raw input image, revealing the fingerprint of the finger. The key that allows Fiberio to display an image *and* sense fingerprints at the same time is its screen material: a fiber optic plate. Right: This fiber optic plate replaces the diffuser in Fiberio's configuration, which is otherwise a standard diffuse illumination setup.

The results we presented in the previous chapter further support our new perspective on touch input: To infer input locations accurately, the user's finger has to be considered in 3D.

In Chapter 3, we described how touch devices may reconstruct 3D information from the 2D contacts they observe. While our prototype Ridgepad achieved this by resolving touch contacts at a fingerprint level, the device itself was input-only and could not support the tasks of today's touch*screen* devices.

In this chapter, we present Fiberio, a rear-projected multitouch table that observes users' fingerprints during interaction (Figure 5.1). Fiberio extracts users' fingerprints to biometrically identify them and reconstruct 3D information *during* each touch interaction. Both, input and output thereby appear on the same surface, allowing Fiberio to be a fully interactive touchscreen.

Fiberio accomplishes projection and fingerprint sensing on the same surface using a new type of screen material: a large fiber optic plate as shown in Figure 5.1. The plate *diffuses* light on transmission, thereby allowing it to act as projection surface. At the same time, the plate *reflects* light specularly, which produces the contrast required for fingerprint sensing.

As a side effect, Fiberio offers all the functionality known from traditional diffused illumination systems. More importantly, Fiberio is the first interactive tabletop system that authenticates users during touch interaction—unobtrusively and securely using the biometric features of fingerprints, which eliminates the need for users to carry identification tokens.

5.1 FINGERPRINT SCANNING AND PROJECTING IMAGES ON THE SAME SURFACE

The challenge of scanning fingerprints on a surface that simultaneously functions as a display for projected images boils down to two contradicting requirements with respect to the screen material. On the one hand, the screen has to reveal fingerprints, i. e., produce contrast between the ridges and valleys of the fingerprint. Known solutions require a *specular* screen surface to accomplish optical fingerprint scanning. On the other hand, to be used as a display, the screen has to allow the rear-projection to produce a visible image, which requires the screen material to be *diffuse*. Unfortunately, specular and diffuse are contradictory requirements for such a surface.

These contradictory requirements eliminate a number of candidate technologies that appear suitable at first glance. Tabletops based on frustrated total internal reflection [58], for example, cannot generate the contrast between fingerprint valleys and ridges and thus do not afford scanning users' fingerprints with sufficient quality.

In this chapter, we demonstrate how to resolve this contradiction. Our prototype Fiberio is a multitouch table that recognizes fingerprints during touch interaction. As shown in Figure 5.1, Fiberio scans the user's fingerprint upon touch, identifies the user during interaction and projects output onto the tabletop surface. Building on the concepts in Chapter 3, Fiberio also reconstructs the user's 3D finger pose from the 2D fingerprint it observes.

5.2 Fiberio's solution: a large fiber optic plate as the surface

Figure 5.1 (right) shows Fiberio's hardware configuration, which is essentially a diffused illumination setup [96]: a 19″ screen (*diffuser*), a projector that rear-projects onto the screen, an infrared illuminant that illuminates the screen from behind, and cameras that observe touch input on the screen.

What distinguishes Fiberio from a regular diffused illumination setup is the nature of the diffuser: At first glance, Fiberio's diffuser appears like a sheet of frosted glass, but it is a 3 mm thick, 4233 dpi fiber optic plate. Its 40 million optical fibers run perpendicular to the surface and transmit light between the top and the bottom of the screen. Such plates, typically marketed for shielding CCD sensors from X-ray radiation in medical applications, are being produced in large numbers today and we repurpose them without modification in our prototype.

In Fiberio, the fiber optic plate resolves the aforementioned contradiction. As we describe in detail in Section 5.4, the fiber optic plate (1) diffuses light on transmission. This causes the light coming from the projector located below the screen to scatter, allowing users to see the image on the surface from all locations around the table. (2) With the correct illumination setup, the fiber optic plate creates a specific type of specular reflection: *frustrated Fresnel reflection*, which is different from the type of reflection used in FTIR-based tabletop systems. This setup causes the infrared light that illuminates the plate from below to produce a visible contrast between fingerprint ridges and valleys, which allows the high-resolution infrared camera below the table to capture fingerprints (Figure 5.1 right). Because of the fiber optic plate, Fiberio is capable of *simultaneously* displaying images and capturing fingerprints.

5.3 Background: optical fingerprint sensing

In order to record fingerprints, a camera needs to produce sufficient *contrast* between a fingerprint's ridges and valleys. Existing diffused illumination systems do not produce this contrast, because the skin of the user's finger diffusely reflects light and because the system's diffuser further blurs those reflections, thereby discarding all the structural details [96].

5.3.1 *Prism-Based Fingerprint Scanning Yields Excellent Contrast*

As shown in Figure 5.2, prism-based fingerprint scanners achieve excellent contrast by shining light through a light diffuser and into a large solid glass prism [173]. (a) Since the light hits the top surface at an oblique angle, any blank part of the surface reflects the light directly into the camera, causing such areas to appear bright. (b) Whenever human skin touches the surface (i. e., the ridges of the fingerprint make contact), the light reflection is *frustrated*. That is, the light exits the prism and enters the finger, where the skin diffuses the light. Thus, little to no light reaches the camera, causing fingerprint ridges to appear dark in the image. The fact that valley locations *reflect* light whereas ridges *absorb* it produces a stark contrast, allowing such devices to capture fingerprints that are high in quality and high in contrast.

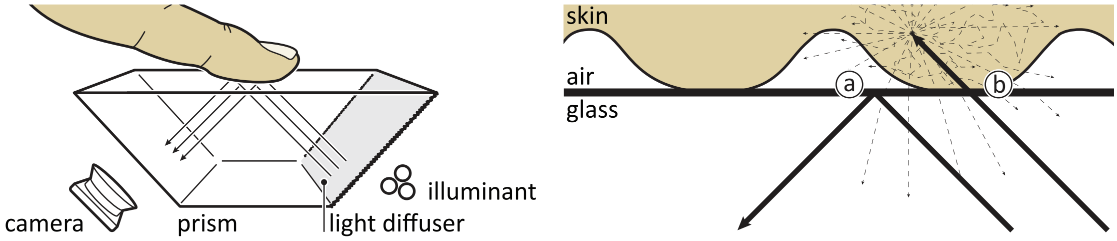


Figure 5.2: Left: Optical fingerprint scanners produce crisp contrast between fingerprint ridges and valleys using a prism, strong illumination that enters the prism and a camera on the opposite side to capture *reflections*. Right: Rays from the illuminant are totally reflected at the prism surface, (a) causing such locations to appear *bright* as the camera sees directly into the light source. (b) Light escapes the prism (i. e., the reflection is frustrated) where fingerprint ridges touch the surface, causing ridges to appear *dark*.

Unfortunately, prism-based fingerprint scanning cannot be integrated into touch*screens*, because the prism construction does not allow these devices to produce visual output. The reason is that, as discussed above, the prism-based design requires a *specular* surface; projection, however, can only image on a *diffuse* surface.

5.3.2 *Touchscreens Based on FTIR Cannot Sense Fingerprints*

As mentioned in Section 5.1, touchscreens based on frustrated total internal reflection [58] cannot be enabled to capture fingerprints. The primary reason is that such systems employ compliant surfaces to act as diffusers and at the same time facilitate sensing touch input. Their structure is coarse, however, which dampens touch input to the extent that fingerprint ridges cannot leave distinct impressions on the waveguide. Such surfaces thus blur all the details required for fingerprint sensing.

Even if we eliminate the light-diffusing property of the surface (e. g., by using a switchable diffuser [73]), the design of FTIR will still not produce the contrast required for fingerprint scanning. Figure 5.3 illustrates such a device without a compliant surface. When the finger touches the surface, the light escapes the waveguide and enters the ridges of the fingerprint. (b) The finger's skin, however, *diffuses the light* at

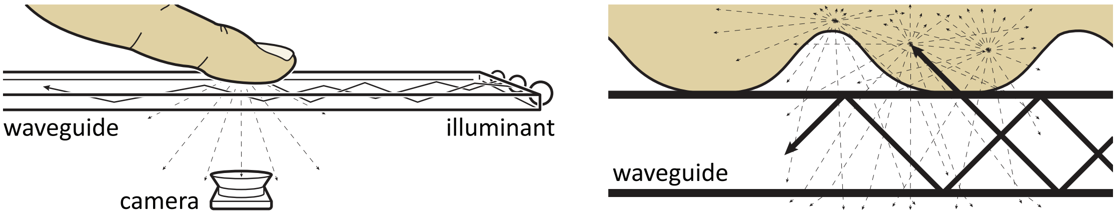


Figure 5.3: Left: Han's FTIR setup [58] does not afford high-contrast fingerprint scanning, even if we eliminate the compliant surface. Right: When a finger touches the surface (i. e., the waveguide), fingerprint ridges frustrate the internal reflection, thus light escapes and enters the fingerprint ridges, which diffuse the light and consequently illuminate adjacent valleys. Therefore, FTIR setups illuminate the *entire* finger upon touch. Since the camera observes the entire finger and thus sees a finger that is illuminated as a whole, such system cannot resolve fingerprints with high contrast.

a depth of 1 mm, which causes the light to spill over into adjacent valleys [48, 163]. Unfortunately, the camera below the waveguide captures this diffused light for ridges and valleys alike. The finger thus appears illuminated as a whole with very little contrast between ridges and valleys.

5.4 working principle and optical path

As explained above, the key innovation behind Fiberio is that the fiber optic plate allows the screen to serve as a *diffuse* surface for projection and simultaneously act as a *reflective* surface for fingerprint scanning. We now describe the details of the optical path that enables this.

5.4.1 *Diffuse Transmission*

The diffusion of projected images inside Fiberio's fiber optic plate is the result of two independent effects: (1) ring diffusion and (2) microstructural effects inside fibers.

Ring Diffusion

As shown in Figure 5.4, light rays shone onto a fiber optic bundle with relatively large-diameter fibers (1 mm) form a cone on exit. This cone manifests itself as a *ring* on a projection surface, here a table surface 5 cm below the bottom surface of the fiber optic bundle.

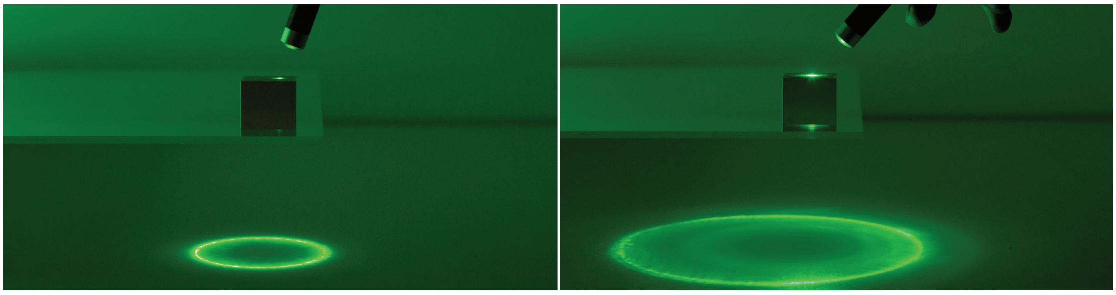


Figure 5.4: Pointing a laser at a fiber optic plate causes it to diffuse incident light into a ring. The angle of incident light thereby determines the radius of the ring, here (left) small and (right) large. The shown fiber optic bundle contains large-diameter fibers (1 mm) and rests 5 cm above a table surface

Figure 5.5 explains this effect. (a) Looking at the fiber from the side, we see that the exit angle along this axis is always identical to the angle on entry. (b) Looking into a glass fiber from one end, a light ray hits the surface of the fiber. (c) The ray enters the fiber and on its way down, the ray describes the shape of a star polygon. (d) We inject a second ray, parallel to the first, but at a small offset. We see how the slight offset causes the star polygon of the second ray to be made from more obtuse angles, allowing this ray to travel a greater angular distance and thus exiting in a different direction. (e) With multiple rays varying by how much they "rotate" inside the fiber, but exiting at the same angle with respect to the fiber, rays form a ring.

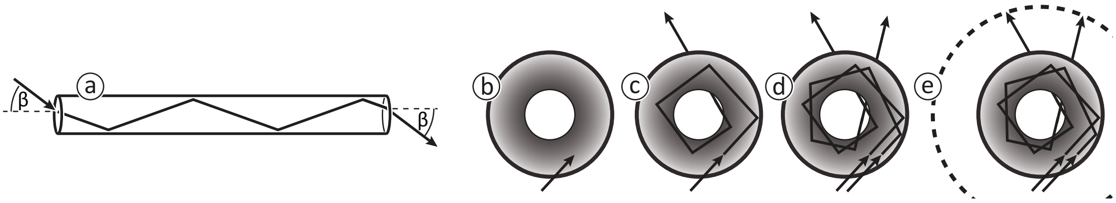


Figure 5.5: Each fiber diffuses parallel incident light into rings. (a) All rays have a constant entry and exit angle with respect to the fiber, which determines the radius of the emerging ring. (b) A fiber viewed top-down. (c) A ray enters the fiber and bounces down the fiber in a rotary pattern. (d) A parallel, but slightly offset ray exits into a different azimuth direction. (e) This variation in azimuth causes incident light to form a ring upon exiting the fiber.

Microstructural Effects Inside Glass Fibers

In contrast to the large-diameter fibers, a fiber optic plate made from very small-diameter fibers produces not only ring diffusion, but also much more diffuse light scattering as shown in Figure 5.6. This is essential for making Fiberio's projected image visible from all sides. At 6 µm, each fiber in our fiber optic plate is only an order of magnitude larger than the wavelength of the light it transports, causing transmitted rays to scatter due to diffraction effects. In addition, the light reflected inside the fibers is subject to the microstructure of each fiber's core and cladding [81], which produces variations in reflection angles inside each fiber. Due to the thinness of such fibers, light is frequently reflected inside the fibers, which causes microstructural effects to manifest themselves in a stronger scattering of light upon exit.

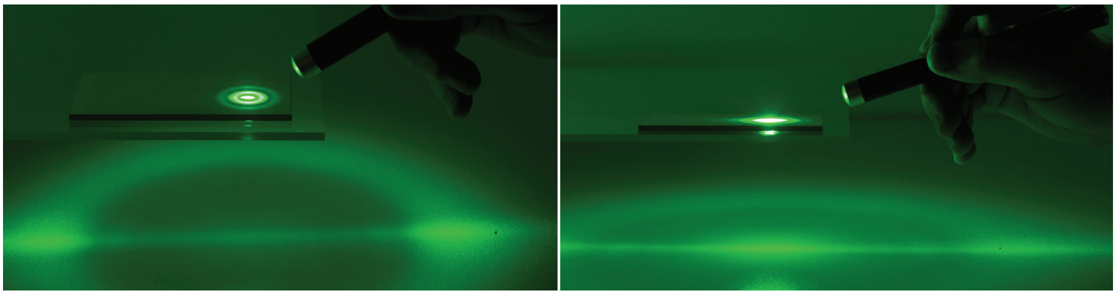


Figure 5.6: Left: A fiber plate with a multitude of very small-diameter fibers (6 µm) blurs ring diffusion, which scatters the incoming light into all directions. This is the basis for good light diffusion, which we need to produce an image on a touchscreen. Right: The fiber plate rests 5 cm above the table. The image on the plate is visible even from extreme angles.

As shown in Figure 5.6, the light diffusion produced by the fiber optic plate exhibits a mild hotspot around the ring. We account for this in Fiberio's setup by mounting the projector at an angle with respect to the fiber optic plate to further increase the amount of light diffusion.

In summary, the fiber optic plate diffuses light while *conducting it along* the fiber; this is different from the traditional way of diffusing light while passing through a diffuse *surface*. Diffusing light while conducting it allows Fiberio to maintain a specular surface, which is key to generating the contrast required for fingerprint capturing.

5.4.2 *Sensing Fingerprints Using Frustrated Reflections*

The specular reflection of light at the *top* surface of the fiber optic plate is what allows Fiberio to capture fingerprints. These reflections occur when the light *exits* fibers.

As illustrated by Figure 5.7a, Fiberio shines infrared light onto the fiber optic plate from below. Some light is reflected at the bottom surface, but most light enters and travels up the fibers (Figure 5.7b). A large portion of the light exits the fibers, but the remaining portion is reflected at the top surface; the reflected light then travels back down inside the fibers and exits at the bottom, where Fiberio's camera observes it. Due to the reflection at the top surface of the plate, locations of fingerprint valleys and areas around the finger appear "brighter" in the resulting image.

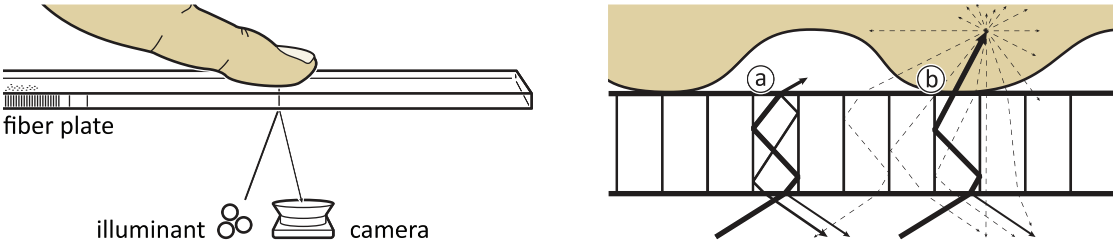


Figure 5.7: Left: In Fiberio's configuration, we positioned the illuminant to shine light onto the fiber optic plate from below. (a) A portion of the incoming light is reflected at the top of the fiber, traveling back down the fiber, where the camera observes it. (b) A fingerprint ridge touching the fiber, in contrast, frustrates the reflection at the top end of the fiber, so that only little light travels back down the fiber, causing this spot to appear dark to the camera.

If, however, a fingerprint ridge makes contact with the top end of the fiber (Figure 5.7b), the reflection at the top surface is frustrated and *almost all* light exits the glass fibers. Only a negligible fraction of light travels back down the fiber, so that this point appears "dark" to the camera.

The contrast between the light reflected at the top surface and the frustrated reflection allows Fiberio to sense fingerprints. Compared to prism-based scanning, this mechanism offers less contrast, because it returns only a small percentage of light. Since we use a camera with very low noise, however, we obtain a good signal-to-noise ratio. Fiberio thus extracts high-quality fingerprints from the captured images with fingerprint edges that appear very sharp.

In the optimal case, all light reflections are frustrated at the top surface when a fingerprint ridge is in contact. However, this requires that the skin of the finger be in direct contact with the fibers. In the case of a very dry finger (or dust on the skin), the frustrations may fail to occur at some locations, causing only a partial fingerprint to appear.

To address this, we created a thin compliant surface by pouring a layer of silicone onto the fiber plate. After having cured, the silicone increased the quality of the fingerprint. At the same time, it reduced the polished impression upon touch, which impeded dragging to a small extent.

In practice, we found no compliant layer to be necessary even for dry fingers, because over time the user's fingers leave small amounts of remnant grease on the surface. This facilitates the process of light coupling into dry skin with-out affecting the quality of projection or fingerprint sensing.

5.4.3 *Placing the Illuminant and Camera for Optimal Contrast*

We explained in the previous sections how the fiber optic plate enables scanning fingerprints at particular locations; to enable fingerprint scanning across the entire large surface of Fiberio, the location of camera and illuminant become important. The illuminant needs to shine light at the entire screen from below while the camera has to be placed so as to capture the reflected light coming back down the fibers.

Figure 5.8 illustrates the challenge. The light that comes back down the fiber is subject to same ring diffusion that we described earlier in the context of projection. To enable the camera to capture reflections across the entire surface, we need to place the camera so that it is in the optical path of the returning light. We explored three solutions.

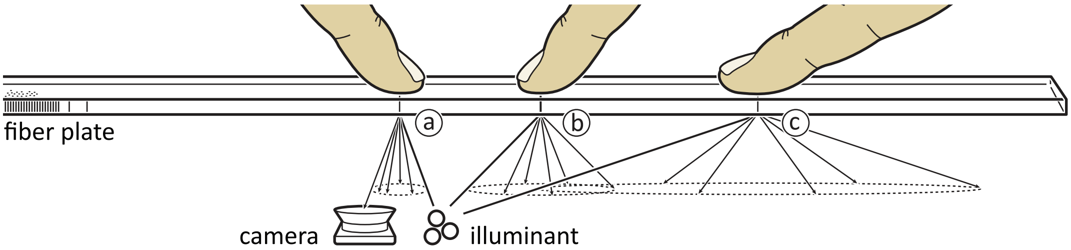


Figure 5.8: The light that comes back down the fiber is subject to same ring diffusion that we described earlier in terms of projection. While the camera will see reflections off the fiber plate's surface at location (a), reflections from locations (b) or (c) will be invisible, because the camera does not sit on the same ring as the illuminant with reference to the respective finger position.

5.4.4 *Solution 1: Shared Location for Camera and Small Illuminant*

Our first solution was to place the illuminant in the same location as the camera—or *around* the camera to approximate a shared location of camera and light source (Figure 5.9 left). This arrangement causes light to ring-diffuse back into the camera for *all* locations on the screen as shown in Figure 5.9 (right). In the shown design, we offset both camera and illuminant from the screen and mounted them at an angle in order to prevent the camera from seeing the direct reflections of the illuminant (i. e., hotspots).

While this design works well on a small prototype, it does not scale to large screens. In this case, the intensity of the reflected light falls off with increasing distance to touch contacts as shown in Figure 5.9b. Eventually, the sensor in the camera will not be sensitive to resolve the contrast between fingerprint ridges and valleys for far-away touches, causing the resulting fingerprints to appear noisy. Since we scaled Fiberio to

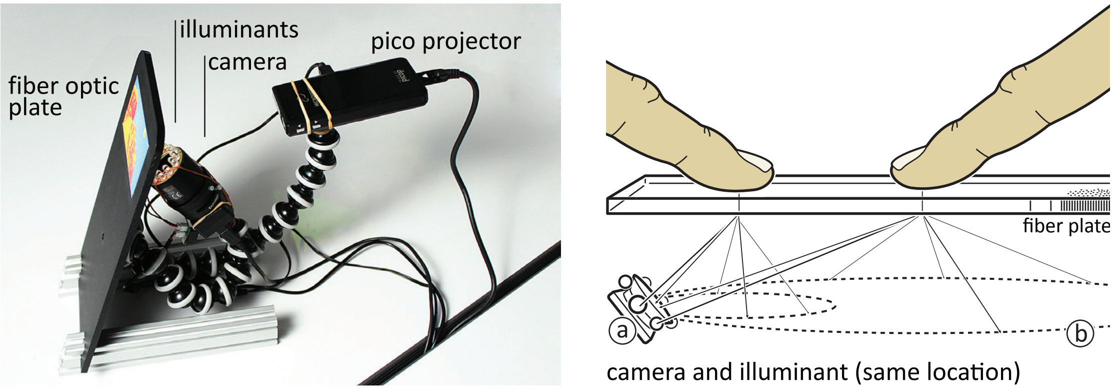


Figure 5.9: Left: This earlier prototype placed the illuminant around the camera to approximate a shared location of camera and light source. Camera and illuminant were tilted with respect to the screen to prevent the camera from seeing the reflection of the illuminant in the fiber optic plate (hotspot). (a) Placing camera and illuminant in the same location causes the returning, ring-diffused light to always hit the camera. (b) The farther away the finger, however, the less intense such reflections appear as they spread along increasingly large rings.

its current 19″ size, we switched to designs that illuminate the screen using a large homogenous illuminant.

5.4.5 *Solution 2: Using a Large Homogenous Illuminant*

Our current Fiberio prototype uses evenly distributed illumination across the entire surface. The illuminant uniformly shoots light at the fiber optic plate from below, creating one evenly illuminated area. Since light intensities are roughly identical across the entire surface, no single hotspot occurs and thus no area of oversaturation or undersaturation in the camera image.

As shown in Figure 5.10, we prototyped two approaches to create a light source that evenly illuminates the fiber plate. In an earlier prototype, we placed a uniform area illuminant below the fiber optic plate (here Acrylite LED [38]) as shown in Figure 5.10a. The main limitation of this solution was that it produced low contrast, as the reflected light from the fingerprint not only competes with light reflected directly off the bottom of the fiber optic plate, but the illumination layer also shines light directly into the camera. We addressed this with yet another iteration on our design.

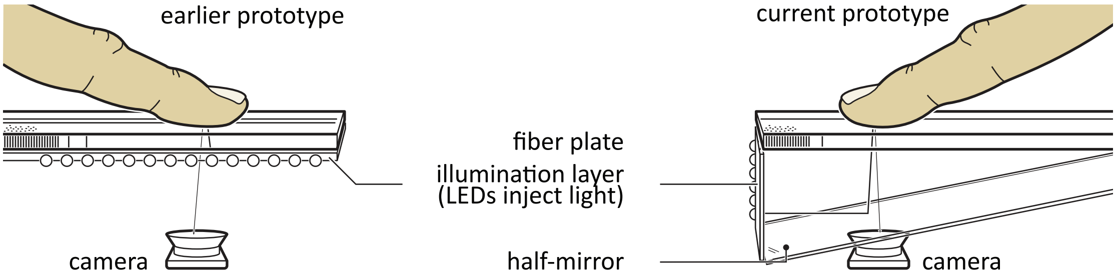


Figure 5.10: Fiberio evenly illuminates the entire surface, creating one even reflection to recognize fingerprints across the whole surface. Left: In an earlier prototype, we used a sheet of Acrylite LED below the fiber optic plate. Right: Our current prototype uses a half-mirror that reflects illuminations from the side.

5.4.6 *Current Solution: Even Illumination Via a Half-Mirror*

Figure 5.10b illustrates the conceptual setup that we use in our current prototype as shown in Figure 5.11. It continues to use Acrylite LED to illuminate a large area. However, we now place the sheet at the side of the table and use a half-silvered mirror to reflect illumination to the fiber optic plate. This prevents the camera from seeing the illuminant layer directly and thus avoids the loss of contrast that characterized our earlier design.

The resulting design works well and since this setup illuminates the screen using a large illuminant, the solution scales well to large screens, even beyond the 19″ of our current Fiberio prototype.

5.5 DETAILS ON HARDWARE SETUP

As shown in Figure 5.1, Fiberio offers a 40 cm×25 cm screen surface (16″×10″, 19″ diagonal). This surface we implement by tiling two 25 cm×20 cm fiber optic plates (Incom B7D59-6), which are polished and feel like a piece of glass.

The BenQ short-throw projector pointed at the screen offers a resolution of 1024×768 pixels. A hot mirror in front of the projector prevents interference with the cameras. Due to its high resolution, the fiber optic plate has no impact on the resolution of the projected image; each fiber measures 6 microns, whereas a projected pixel measures ∼390 microns and thus covers a multitude of fibers. Projected images are visible even from extreme angles (Figure 5.6 right), because of the numerical aperture of the fibers we use (1.0). The refractive index of their core (1.8) and that of the cladding (1.49) allows for maximum acceptance and exit angles of 90°.

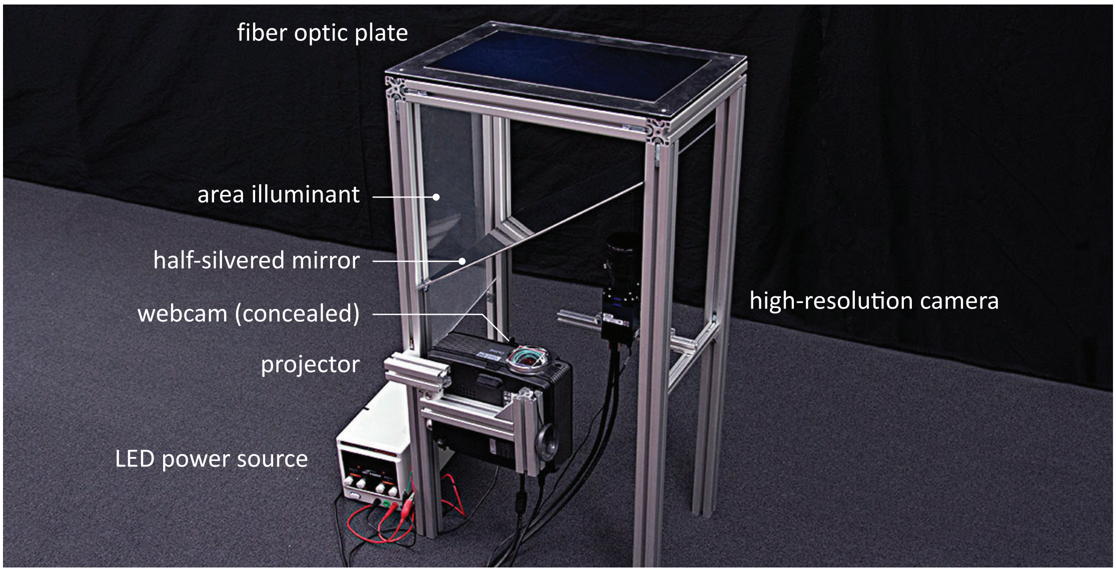


Figure 5.11: Our current prototype illuminates the fiber optic plate evenly, creating optimal reflections for the camera to resolve fingerprints at all locations on the touch surface. We use an area illuminant (Acrylite LED with light injected from the sides) mounted to the side of the table and a half-silvered mirror to reflect illuminations. All sides of the table are covered with black cloth to prevent reflections from the environment in the camera image (here we left out the covers to show the inner components of the system).

A frame made from a 40 mm aluminum profile system holds Fiberio's components in place. Fiberio's height of 38″ is designed to minimize fatigue on the standing workstation.

To achieve fingerprint scanning across Fiberio's entire surface at a resolution needed for reliable scanning (500 dpi [93]), our setup would require a camera resolution of 8000×5000 pixels. In our prototype, we used a high-resolution camera (Teledyne Dalsa Falcon2, 4000×3000 pixels, 60 fps), which observes only a sub region of Fiberio's surface (20 cm×15 cm). We supplemented this approach with a web camera to enable touch interaction across the entire surface (Sony PS3, 640×480 pixels, 75 fps), which we set up in a diffused illumination arrangement. Both cameras and the projector are calibrated to a shared world-coordinate system.

While our prototype setup using a half-mirror achieves the best illumination across Fiberio's whole surface, the switch to scanning prints on only a quarter of the surface allowed us to reduce the footprint of the table. We therefore substituted the half-mirror with one 4 W infrared illuminant.

5.6 IMAGE PROCESSING

Currently, Fiberio locates and tracks all touches based on the low-resolution camera, implementing a typical diffused illumination processing pipeline [96]. When touches enter the region observed by the high-resolution camera, Fiberio locates fingerprints, extracts them along with their features, and matches features against the records stored in its fingerprint database.

Future versions of Fiberio will cover the entire screen either using a 2×2 array of high-resolution cameras or using a single camera and a high-speed pan and tilt mirror.

5.6.1 *Fingerprint Processing Pipeline*

To extract the locations and directions of fingerprint features, i. e., ridge endings and bifurcations (so-called *minutiae*), which allow identifying users, we implemented the algorithms commonly used to process fingerprints [93]. Figure 5.12 illustrates the pipeline we implemented.

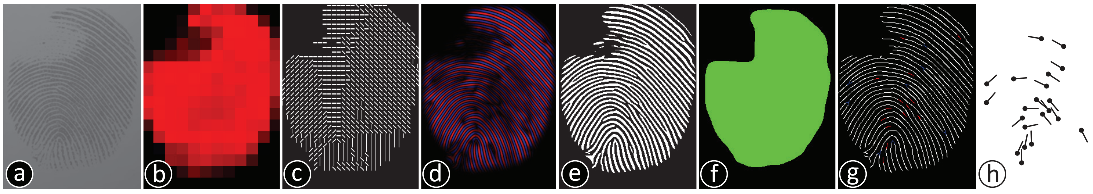


Figure 5.12: Fingerprint processing pipeline (images are cropped). (a) Raw image, (b) areas of high standard deviation, (c) flow field, (d) Gabor filter, (e) binarized fingerprint, (f) mask, (g) skeleton and (h) extracted locations and orientations of all fingerprint features.

The 768×768 pixel raw image shown in Figure 5.12a contains the reflections from the user's finger. Fingerprint ridges appear as dark lines inside a brighter area. Fiberio starts by removing possible luminance gradients by subtracting a low-pass copy from the image. (b) Fiberio locates fingerprints by calculating the standard deviation of brightness values for 16×16-pixel subregions in the image. High brightness deviation indicates the presence of adjacent ridges and valleys. Fiberio uses this to produce a mask—all further processing takes place inside this area.

To improve the contrast of the fingerprint, (c) Fiberio computes the direction of the main gradient across all 8×8-pixel subregions, resulting in the flow field of the fingerprint. We input the flow field into (d) a Gabor filter, which improves the edges in the fingerprint according to their orientation, thereby smoothing noisy and interrupted ridges. (e) Binarizing the result now brings out a sharp contrast between ridges and valleys in the fingerprint.

To extract the locations of all minutiae from the fingerprint, Fiberio obtains (f) a refined mask of the fingerprint and (g) derives the skeleton of the binarized fingerprint. The skeleton reveals the locations and orientations of minutiae; locations in the skeleton that have three neighboring pixels are bifurcations, whereas locations with only one neighbor are ridge endings as shown in Figure 5.12h.

To match two fingerprints based on their minutiae, Fiberio finds the best spatial alignment of both point sets using Bozorth matching [93]. It then computes a matching score based on the number of minutiae that match in terms of location and angle. When Fiberio compares an observed fingerprint to fingerprints in its database, it requires fingerprints to match in at least 10 minutiae locations.

5.6.2 *GPU Acceleration and Resulting Performance*

Fiberio runs touch recognition, fingerprint extraction, matching, and graphics using parallel threads, allowing it to stays responsive to user input at all times. Since processing fingerprints is computationally expensive, we implemented our pipeline in CUDA 4.2 to run on the GPU (NVIDIA GTX 680), which allows our system to run at interactive rates.

Extracting all minutiae from the raw fingerprint image currently takes Fiberio 21 ms per frame. The speed of matching fingerprints currently increases linearly with the number of records in the database (0.55 ms per record).

5.7 RECONSTRUCTING FINGER POSES IN 3D FROM FINGERPRINTS

Being able to extract the fingerprints from the 2D touch image, Fiberio analyzes fingerprints to reconstruct users' 3D finger poses in addition to identifying users. One of the outcomes of this reconstruction is that Fiberio supports high-precision touch input as described in Chapter 3, but this time on a fully interactive touchscreen. While Ridgepad accomplished 3D reconstruction using image matching, Fiberio implements a more elaborate pipeline to determine the 3D finger pose.

Figure 5.13 shows Fiberio while reconstructing the user's 3D finger pose from the fingerprint. On the left, Fiberio shows the user's fingerprint and parts of the hovering finger as observed by the camera. From this image, Fiberio reconstructs the 3D configuration of the user's finger and renders a 3D hand model on the right that matches the user's hand pose. Note that the hand model is *mirrored* at the touch surface and mimics the user's finger 3D rotations as they change in yaw, pitch, and roll.

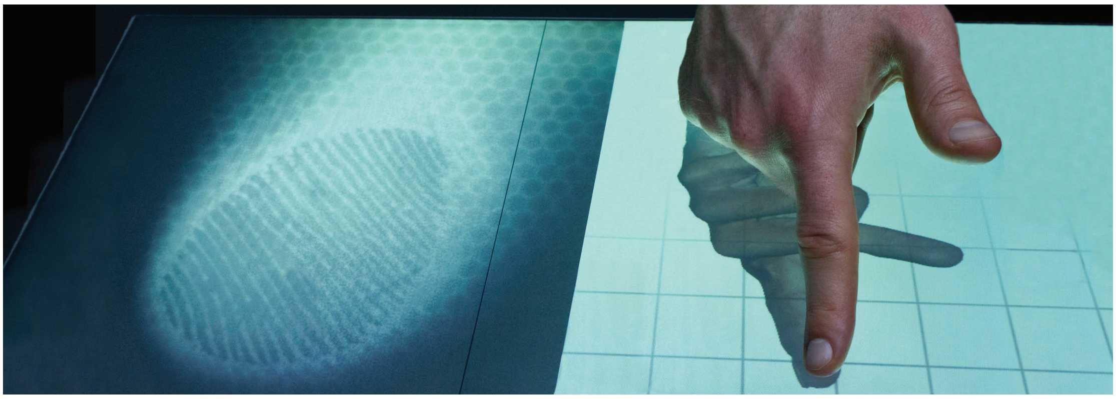


Figure 5.13: Fiberio reconstructs the user's 3D finger orientation from the observed 2D touch image by extracting and analyzing the fingerprint. Fiberio renders its reconstruction in the form of a 3D model that matches the user's pose.

5.7.1 *Algorithm: From the 2D Fingerprint to 3D Rotations: Yaw, Pitch, and Roll*

To reconstruct the 3D rotation angles of the user's finger, Fiberio reuses the features that we extract when identifying users as explained in Section 5.6. This identification step not only identifies the user, but in addition reveals with which finger the user has touched the surface, as fingerprints are finger-specific [93]. We base all following explanations on having identified the user and the finger used for touch input as a first step during the identification phase.

While Fiberio obtains the yaw rotation of the user's finger directly from the image, we predict roll and pitch rotations using pre-recorded training data. Key to extracting all three angles is deriving a contact mask that precisely encompasses the fingerprint in order to distinguish it from the hovering parts of the user's finger in the input image.

Contact Mask: Distinguishing the Contact Area from Hovering Parts

In order to predict the finger rotation based on the fingerprint, we first need to localize the fingerprint and extract it from the remaining part of the finger in the image. The difficulty lies in precisely detecting the outer edges of the fingerprint ridges, because an inexact mask will later cause artifacts outside the contact area to influence the prediction of finger pitch and roll. As shown in Figure 5.14a, the structure of the fiber optic plate becomes apparent outside the contact area and causes spurious features if not masked correctly.

To obtain a mask of the contact area between the finger and the touch surface as shown in Figure 5.14, we apply a threshold to extract the rough region of the finger, which results in the touch contact and part of the hovering finger (b).

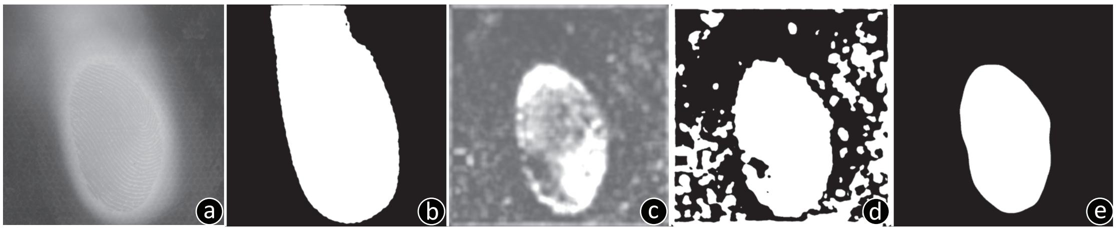


Figure 5.14: Extracting the contact mask from the (a) raw image: (b) Applying a threshold produces the touch contact and parts of the hovering finger. (c) We calculate the magnitude of flow intensities using a sliding window-based approach and (d) threshold the result. Extracting the largest resulting contiguous area, multiplying it with the threshold (b) to remove spurious blobs, blurring and thresholding the result produces (e) the final mask.

To refine the mask, we analyze the covariance of brightness intensities in the raw image, which is high at locations in the image that exhibit closely collocated fingerprint ridges and valleys, i. e., areas that have a certain "direction" (c). To accomplish this, we reuse the magnitude of the flow intensities that we calculated to obtain the directions of the main gradient for the region around each pixel in the image in Section 5.6 (see Figure 5.12c). Performing this on tiles measuring 16×16 pixels thereby produces a mask with high enough precision along the outline of the contact area. As shown in Figure 5.14d, thresholding the magnitudes, which we normalize across the whole fingerprint, at 90% results in an approximation of the contact mask.

Finally, we extract the largest connected component from the thresholded magnitudes, multiply it with the hover threshold (b), blur the resulting blob and threshold it again to obtain the final mask with a smooth outline as shown in Figure 5.14e.

From the resulting mask, we extract the typical properties of a touch contact, such as position, width, and height.

Yaw

Having located the touch contact in the input image, we can now extract the yaw rotation of the user's finger by analyzing the hovering parts of the finger. In Figure 5.14, we previously extracted the (b) hovering part of the user's finger as well as (e) the contact mask. As common in processing touch input in diffused illumination systems [96], we simply relate the center locations of both blobs to obtain the yaw rotation of the user's finger. Figure 5.15a shows the contact mask overlaid in gray onto the hovering finger and the resulting yaw estimate.

In addition to the finger's yaw rotation, we exploit the diffused illumination properties of the fiber optic plate to estimate the user's position around the table. We accomplish this by adaptively thresholding the entire raw touch image we obtain from the web camera at a very low brightness intensity. This reveals the user's hand and parts of their arm in addition to the finger. Starting at the touch position, we trace a path through the user's hand and arm to the edge of the tabletop. If an arm does not extend all the way to a table edge, our algorithm extrapolates linearly from the farthest point inside the visible part of the arm.

Pitch and Roll

To extract the pitch and roll rotation of the finger from the masked part of the fingerprint (Figure 5.15b), we analyze the curvature of the visible part of the user's fingerprint and derive a histogram from all local curvatures. Similar to how we trained Ridgepad, we again use a *k* nearest neighbor approach to determine the rotation angles, this time matching the histograms of local curvatures.

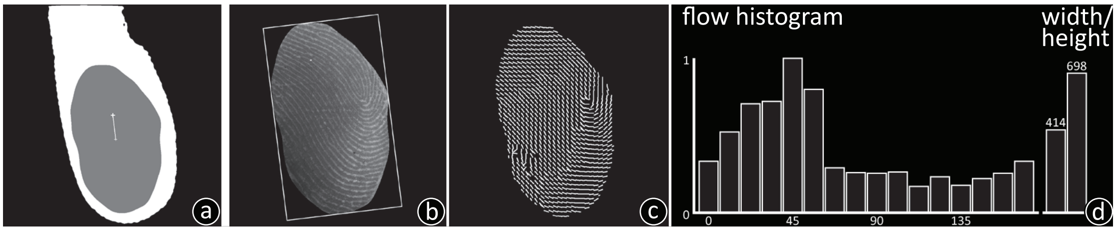


Figure 5.15: (a) The yaw rotation of the finger results from relating the center of gravity of the touch mask (gray, small cross) to the center of the hovering finger (white, large cross). (b) We determine the width and height of the mask in the direction of the yaw rotation. We separately extract the fingerprint from the raw image and compute the (c) flow field using a sliding window-based approach for regions within the mask. (d) We then derive a histogram of flow directions, which we normalize and correct for the yaw rotation we obtained in (a).

We compute the histogram of fingerprint curvatures using the flow field we obtained to improve the contrast of the fingerprint during identification in Section 5.6. As shown in Figure 5.15c, we again tile the fingerprint in regions of 16×16 pixels and obtain the direction of the main gradient within that cell, each time using the gradients in the surrounding eight cells to stabilize our calculations. We then correct this direction for the previously computed yaw orientation and derive a histogram with 16 buckets, each bucket representing a direction (Figure 5.15d). Compensating for yaw rotations allows us to obtain yaw-independent histograms, which is crucial to matching pitch and roll later.

In order to ensure that this approach produces valid curvature histograms, only the flow of tiles that actually contain a part of the user's fingerprint must be added to the histogram. We use the previously computed mask to decide whether or not the flow within a tile should be added to the histogram.

In addition, we discard the flow direction of a tile if the magnitude of flow within the tile is below a threshold, i. e., the flow within that tile is not pronounced enough. This prevents adding spurious flow directions to the histogram, such as those that may occur around the outline of the contact area. Spurious flow directions may also occur inside the fingerprint, such as in regions that are too noisy, causing ridges and valleys not to be discernible.

After deriving the histogram of local curvatures, we normalize all values in the histogram. This allows us to treat fingerprints that result from touches independent of the size of the contact area. Normalizing histograms also facilitates comparisons during the matching process.

Separately from the previous computations, we derive the width and height of the contact area in the direction of the finger yaw, which we obtained before. We thereby consider the height of the contact area in the direction of the user's finger and the width across its direction as shown in Figure 5.15b. This again allows us to obtain yaw-independent values for the dimensions of a touch contact.

Altogether, Fiberio extracts 18 features from each fingerprint to predict pitch and roll rotation of the user's finger. They comprise 16 features derived from the histogram of local curvatures and 2 features obtained from the dimensions of the contact area.

5.7.2 *Training the Predictor*

We collect training data to derive a model for the prediction of pitch and roll angles using a procedure that is similar to Ridgepad's training (Section 3.4.1). To train our system using their fingerprints, users assume various pitch and roll combinations with their finger and touch Fiberio's surface. Fiberio captures the visible part of the user's fingerprint, extracts the 18 features we described in the previous section, and stores them in a database along with the current roll and pitch rotation angles.

Figure 5.16 shows the pitch and roll rotations users assume to touch the surface and train the system. In our tests, users touched the surface 3 times for each combination of pitch and roll rotation, i. e., 5 roll angles × 5 pitch angles × 3 repetitions = 75 times. While we experimented with an optical tracking system to record ground-truth data for the assumed 3D finger poses (similar to our setup in Section 3.5), we noticed no difference in system performance if the assumed finger poses were instead controlled by an experimenter. Including an experimenter in the training procedure also rapidly sped up training procedure and involved less overhead managing the optical tracker.

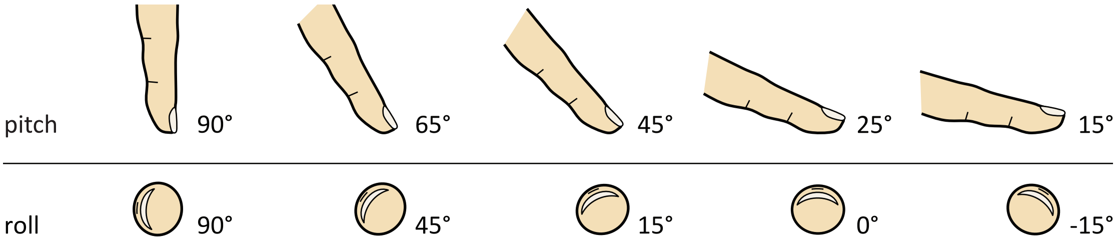


Figure 5.16: To train our classifier for finger roll and pitch prediction, participants touched Fiberio's surface using each of these combinations of finger rotations with 3 repetitions.

5.7.3 *Runtime Prediction: Matching Histograms*

During runtime, Fiberio first executes the fingerprint processing pipeline described in Section 5.6 to identify users and subsequently predicts the user's 3D finger pose. To accomplish this, Fiberio performs the steps described in the previous section, including building the contact mask and deriving the yaw rotation directly from the image.

In a second step, Fiberio extract the 18 features from the fingerprint and compares them with all records in its database. We thereby use a simple metric to compare histograms. We interpret a normalized histogram record as a 16-dimensional vector and obtain the similarity of two records by calculating the angle between the two vectors. We derive the similarity of the remaining 2 components of our feature vector by their absolute differences.

Finally, we predict roll and pitch rotation of the user's finger based on the k most closely matching database records. We select $k = 3$, which corresponds to the number of repetitions per angle in the training phase; for a larger number of training samples, k should be chosen higher. We average the roll and pitch angles associated with the top k matches using spherical linear interpolation.

From the predicted combination of yaw, pitch and roll rotation of the user's finger, we render a 3D model of a hand underneath the user's hand as shown in Figure 5.13. The 3D model is mirrored at the surface of Fiberio and mimics the user's in-place finger rotations.

5.7.4 *Limitations*

Our current approach is limited to reconstructing the 3D finger pose from a 2D fingerprint for a single finger. The 3D model shown in Figure 5.13 assumes that users touch the surface using their right index finger and that all other fingers are tucked in. Our approach does not allow recognizing the pose of the user's fingers that are not in contact with the surface or the shape of the user's hand.

Future versions of our system will include the reconstruction of several simultaneous and closely collocated touch events. From this, we intend to estimate and render the user's 3D hand shape by reconstructing the pose of each finger and using inverse kinematics and constraint solving to obtain the shape of the whole hand. While this reconstruction will still be underspecified unless all of the user's fingers are in contact with the surface, our prediction will approximate reality as more fingers touch the surface.

5.8 BIOMETRIC USER IDENTIFICATION ON TOUCHSCREENS

Since Fiberio incorporates fingerprint scanning into a touch*screen*, it solves a long-standing challenge in human-computer interaction: biometric user identification. Fiberio thereby performs user identification upon each touch, which allows us to integrate secure authentication into touch interaction with graphical user interfaces.

5.8.1 *Example Scenario: Collaborative Approval of Invoices*

Figure 5.17 shows one of the examples we have implemented to demonstrate Fiberio's capabilities in terms of *secure* authentication. A bank clerk and his manager approve invoices by pressing the 'pay' button on each invoice. When the invoice exceeds the clerk's approval limit as shown in Figure 5.17a, Fiberio refuses the transaction until (b) the clerk asks the manager to (c) approve the invoice. He does so by pushing the *same* button the clerk had pressed. This time, however, the transaction is performed under the manager's credentials, verified against his higher approval limit, and approved.

Fiberio enables this scenario by authenticating users during each touch interaction and, in this scenario, by retrieving their approval limit from a database. Fiberio does so by authenticating users based on their fingerprints—integrated seamlessly into regular interaction. This allows Fiberio to avoid the need for login procedures, identification tokens, or reduce reliability and security—something related systems currently compromise on.

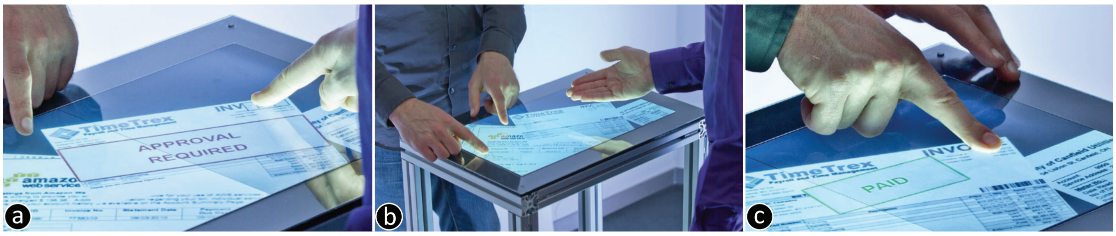


Figure 5.17: Example scenario. A bank manager (left) and clerk (right) approve invoices. (a) When the clerk encounters a bill above his approval limit, Fiberio refuses the payment transaction. (c) The manager completes the transaction by pushing the same button, but this time, Fiberio executes the operation using the manager's credentials, verifies it against his approval limit, and completes the operation.

5.9 EVALUATION

The purpose of our evaluation was to verify that Fiberio's sensor setup captures fingerprints with sufficient quality to allow it to recognize users reliably. To evaluate identification performance, we compared 30 fingers (three fingers per each of the 10 participants, ages 20–32, 2 female).

5.9.1 *Apparatus*

We conducted this evaluation using an earlier version of our prototype, which featured a lower-resolution camera (8.8 MP Flea3). Considering that the image sensor of that camera was inferior to that of our current camera and we used our current algorithms for processing input, the results from this evaluation apply to our current prototype. The study apparatus was set up to capture fingerprint images at a resolution of 500 dpi and 8 ms shutter time. We performed all processing on a 2.2 GHz Intel Core 2 Duo processor with 4 GB of RAM and an NVIDIA GTX 680 graphics card using the described algorithm. The projector was switched off.

5.9.2 *Task and Procedure*

As shown in Figure 5.18, participants touched the screen region captured by the high-resolution camera during each trial, each time using one of their right hand's index, middle or ring finger. Participants thereby used their finger pad for touch input and repeated input five times, performing fifteen trials overall. Due to the limited frame rate of the camera we used during the evaluation, participants were required to hold a touch for around 400 ms. This allowed the camera to capture frames reliably. This is no longer required for our current system due to the substantially larger frame rate of our current camera. For each trial in the evaluation, Fiberio processed only a single frame, namely the one in which the contact area of the touch was maximal. Participants received no feedback during the evaluation.

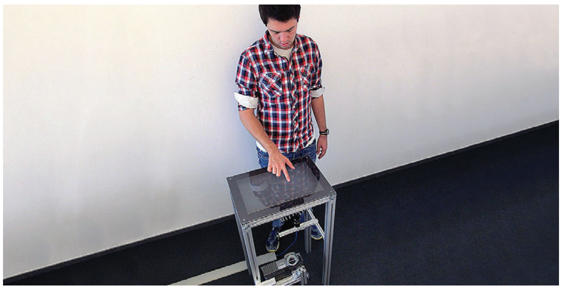


Figure 5.18: A study participant during the evaluation. The prototype was configured to provide no feedback.

5.9.3 *Processing*

The evaluation resulted in 150 captured fingerprints, from which we extracted the minutia sets and created a database. We then performed minutia-based matching on each of the captured prints against all 149 other records.

5.9.4 *Results and Discussion*

The cross-validated analysis resulted in 148 of 150 fingerprints being *correctly* matched, 0 *wrong* matches, and 2 *no matches* (i. e., samples that produced less than the minimum number of 10 minutiae needed for identification). The average processing time for matching a minutia set against all others was 267 ms.

These findings show that Fiberio identifies users reliably by their fingerprints and at interactive rates. Since the speed of user identification scales linearly with the number of samples in the database, this process runs asynchronously to still support responsive interaction.

Of course, participants used their finger pads when providing input, which allowed for optimal feature extraction. While flat fingers exhibit more than 100 minutiae, fingertips contain fewer features ($0.18/\text{mm}^2$ [93]). However, 12–15 visible features suffice to identify users when touching, which fingertips may provide depending on their tilt. Fiberio's height of 38″ facilitates touching with flat finger angles, which is optimal to extract a multitude of features. While users might be less careful during regular use, a live system could produce feedback on their touch events and ask for repeated input upon unsuccessful identification.

Note that a lack of visible features in a touch does not lead Fiberio to misidentify users. If a fingerprint exhibits too few features, Fiberio does not attempt to identify users.

5.10 benefits and limitations

In addition to resolving touch events at a fingerprint level, Fiberio implements a standard diffused illumination table. This results in additional desirable properties, such as the ability to detect hover and fiducial markers as shown in Figure 5.19. On the flip side, similar to other diffused illumination setups, Fiberio's rear projection requires space and it is susceptible to interference by strong infrared light sources in the environment.

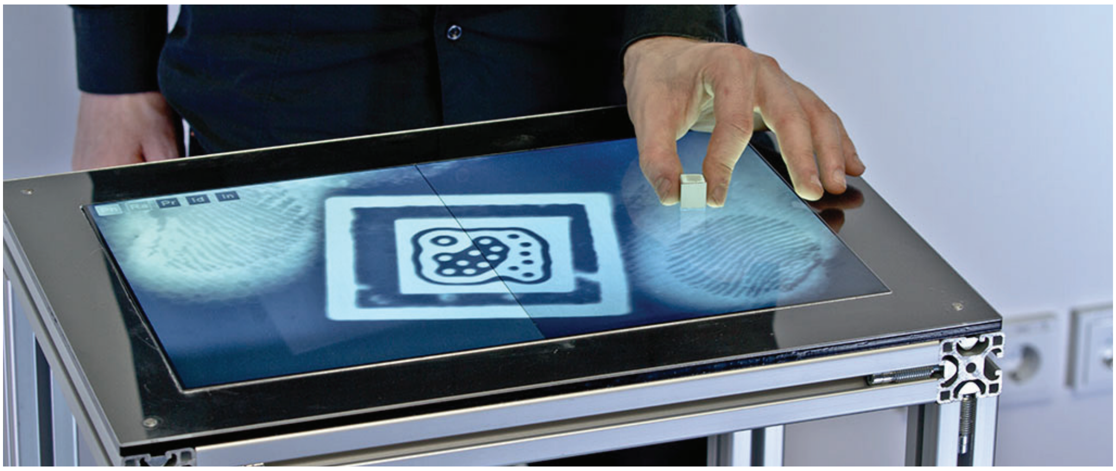


Figure 5.19: Fiberio recognizes touch, but also hovering objects (here fingers) as well as fiducial markers. This reactTivision fiducial marker [79] measures 3 mm × 3 mm.

A positive side effect of the fiber optic plate is that Fiberio is inherently free of parallax. Users see the projected output *on top* of Fiberio's screen; when users touch that output, Fiberio's cameras see this touch contact exactly where it occurs, because touch contacts appear at the bottom surface of the fiber optic plate. Combined, this allows for particularly precise input.

Finally, Fiberio is subject to the same limitations as other biometric authentication mechanisms, such as the risk of spoofing using fake fingerprints [93], as well as concerns in terms of surveillance and respecting users' privacy. To evaluate Fiberio's capabilities in identifying users amongst a large population, a deeper evaluation of the system with a large number of participants of a large span of ages and a wider range of demographics.

5.11 CONCLUSIONS

In this chapter, we presented Fiberio, a touchscreen that senses fingerprints. The key to making this possible is the fiber optic plate, which offers both specular reflection and diffuse transmission. This allows Fiberio to simultaneously display images and scan users' fingerprints on the same surface.

Fiberio's capabilities allow us to implement all the concepts that we introduced in this and previous chapters on an interactive touch*screen*. Fiberio reconstructs 3D information about the space above the screen from the 2D image it observes. It accomplishes this by analyzing each user's fingerprint, identifying the user, and reconstructing the user's finger pose in 3D. These capabilities enable Fiberio to redeem our findings in Chapter 3 and implement high-accuracy touch sensing.

As a side effect, Fiberio solves a long-standing problem in human-computer interaction in that it identifies each user during a touch event based on their fingerprint. For fifteen years, researchers have hypothesized the existence of such a touchscreen, be it for activity logging [138] or access control [34] in collaborative scenarios. Fiberio enables such multi-user applications and implements secure and reliable access control for each touch event. Fiberio thereby extends the notion of touch-input events by the user's identity: In addition to (x, y) coordinate pairs that user interface controls receive, each such control now receives the user's identity.

6 GENERALIZING 3D FROM 2D TOUCH TO LARGE FLOORS

This chapter is based on results of the group project "Multitoe," published in [10, 22].

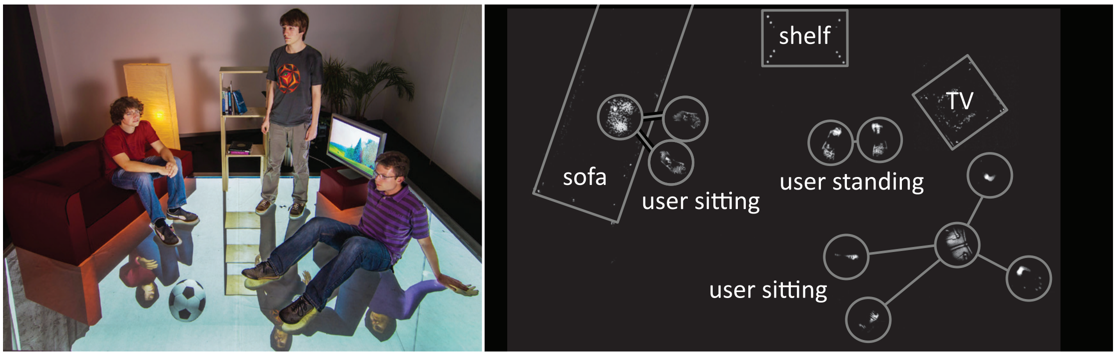


Figure 6.1: 3D reconstruction from 2D touch contacts extended to large multitouch floors. Left: Our prototype recognizes input from three users and three pieces of furniture solely in the form of 2D high-resolution per-pixel pressure sensing in the floor. Right: From the sensed 2D pressure image, we reconstruct an understanding of the physical 3D world: the position and orientation of multiple users, the identity of users in the form of personalized avatars and users' 3D body poses. The system displays a mirrored representation of this understanding on the floor (left).

In this chapter, we describe how the same principles of reconstructing 3D information about the objects that are in contact with the 2D touch sensor extend to larger surfaces. We demonstrate this at the example of a multitouch floor that observes touch in the form of high-resolution per-pixel pressure.

The main contribution of this chapter is a new approach to 3D user and object tracking in a smart room based on a single touch sensor embedded into the floor. While the sensor is limited to sensing *contact* with the surface, we demonstrate how to reconstruct a range of objects and events that take place on as well as in the 3D space *above* the surface, such as user's poses and collisions with virtual objects. We thereby build on the concepts that we introduced in Chapter 3 and Chapter 5 for reconstructing 3D from 2D touch contacts on surfaces that users interact with using their hands. We now generalize them to floors, which observe different kinds of input: soles as users walk across the floor, the texture of users' clothing as they kneel or sit down as well as the imprints left by passive objects in a room, such as furniture. We demonstrate how all such input still leaves characteristic imprints, which we analyze to identify users, reconstruct the 3D posture of users' bodies, and enable high-precision touch input using feet—similar to how we processed users' fingerprints to reconstruct additional information in previous chapters.

To explore this approach, we present a large back-projected floor installation that senses high-resolution pressure input. Incorporated into a room, we use our prototype to demonstrate our vision true to scale as shown in Figure 6.1. We first demonstrate how to advance multitouch interaction from traditional touch devices, such as mobile devices and tabletops, to a large floor that is operated based on foot input. This comprises concepts, such as precise direct manipulation and techniques to overcome the inherent uncertainty of input of floors, including as inadvertent activation and the lack of input modes. We then show how to reconstruct the users and objects in the room in 3D from just the 2D touch contacts they leave on the floor.

6.1 DIRECT MANIPULATION ON MULTITOUCH FLOORS

The nature of *direct* touch input known from tablets or tabletops limits the size of such devices. Contents can only be touched if located within arm's reach. While this is no problem on mobile devices or even small tabletop systems, tables larger than arm's length pose a challenge for users, as they need to artificially extend their reach. In order to preserve the direct touch concept, current tabletop makers have opted to create coffee table-sized devices, such as the Microsoft 30″ Surface table [99] or the SMART 27″ tabletop system [143].

However, the size constraints of tabletops have limited applications that run on horizontal surfaces to those that fit available sizes. This excludes applications in which users interact with *thousands* or *ten thousands* of on-screen objects, such as complex visual sensemaking applications.

In this chapter, we explore interaction with direct touch surfaces that are orders of magnitude larger than tables. For this purpose, we integrate high-resolution multitouch technology into back-projected floors. Unlike tabletop users that stand along the table's *perimeter*, floor users walk *across* these surfaces, allowing them to reach any part of the floor—independent of the size of the installation.

In order to enable direct manipulation on floors, we base our design on frustrated total internal reflection with high-resolution camera as shown in Figure 6.2. Unlike earlier floor installations that achieved display size at the expense of input resolution (e. g., [29] and [85]), we demonstrate how the use of FTIR allows us to maintain the direct

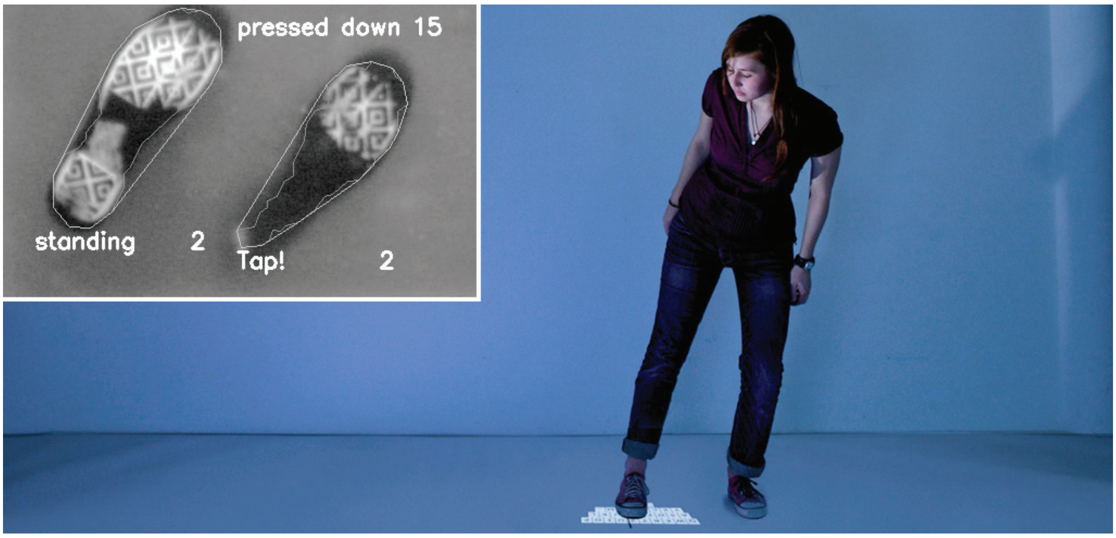


Figure 6.2: Integrating high-resolution FTIR into a back-projected floor allows the floor to see the pressure distribution of the user's soles (inset top left, as seen from below). In the shown situation, the floor ignores the foot on the right based on its posture, yet allows the foot on the left to interact. By identifying the user based on her sole patterns, the floor has attached a user-specific high-precision pointer to her foot, which allows her to operate tiny controls, here a keyboard.

manipulation interaction model of tabletop systems, despite the dramatically different size.

As suggested by its name, the interaction concept of tabletops, i. e., direct *mani*pulation was designed with hands in mind. The adaptation to interactive floors results in the following series of challenges.

First, users stand on floors. We address inadvertent activation by making the floor ignore all input unless users demonstrate a specific foot posture (i. e., tapping in our case as shown in Figure 6.3a).

Second, distances on floors are potentially very large. We address this with location-independent pop-up menus that users invoke by jumping (Figure 6.3b).

Third, to allow for a consistent interaction model, the floor needs to know which parts of their feet/shoes users expect to be "active." We conducted a simple study in which most participants expected not just the contact area, but the entire *projection* of their shoes to actively trigger interactions.

Fourth, feet are roughly 200 times larger than fingertips and less precise. When necessary, we offer a high-precision mode that condenses a user's foot into a single "hotspot" (Figure 6.3c). Since users disagree about the location of this hotspot, we allow them to customize its location. To enable personalization, the floor recognizes users based on their sole patterns (Figure 6.3d).

After establishing basic touch interaction on floors, we take a closer look at algorithms and at the additional functionality enabled by FTIR floors: how to track users' heads based on the pressure distribution in their soles (Figure 6.3e), and how to enable high-degree of freedom interaction (Figure 6.3f).

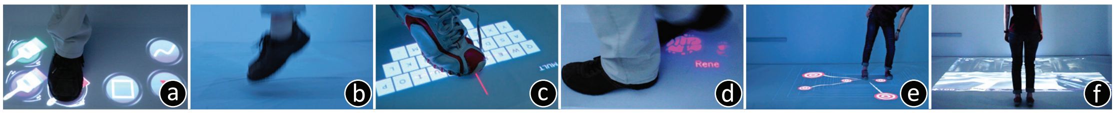


Figure 6.3: (a) Users trigger interactions with a specific foot posture, (b) invoke menus by jumping, (c) interact precisely using a hotspot, which is (d) enabled by sole-based user recognition. (e) FTIR-based tracking also allows controlling applications using body posture and (f) foot posture.

6.2 PROTOTYPE PLATFORM TO SIMULATE INTERACTIVE MULTITOUCH FLOORS

Most of the functionality of our floor design is enabled by FTIR. In this section, we explain the technology in additional detail and juxtapose it to other technologies we have tried.

6.2.1 *Tracking Using Front DI and High-Resolution FTIR*

We initially experimented with traditional rear-diffuse illumination. While we found it to work well with light soles, it produced no effect when users wore shoes with black soles as shown in Figure 6.4a.

The key step was to add FTIR. The main benefit of FTIR in our application scenario is that it makes pressure visible, as was explored earlier in the context of wall displays [30]. As illustrated by Figure 6.4c, bringing FTIR to floors reveals weight distributions; here we see that the foot on the left bears most of the weight, while the foot on the right does not, indicating basic properties of the user's posture.

Unlike other floor designs, we use a camera whose resolution is comparable to the resolution offered by multitouch *tables*, i. e., a pixel size of 1.0 mm (comparable to the components used inside a Microsoft Surface table). As illustrated by Figure 6.4c, this reveals the next level of structure inside the sole, such as patterns and logos of the shoe manufacturer.

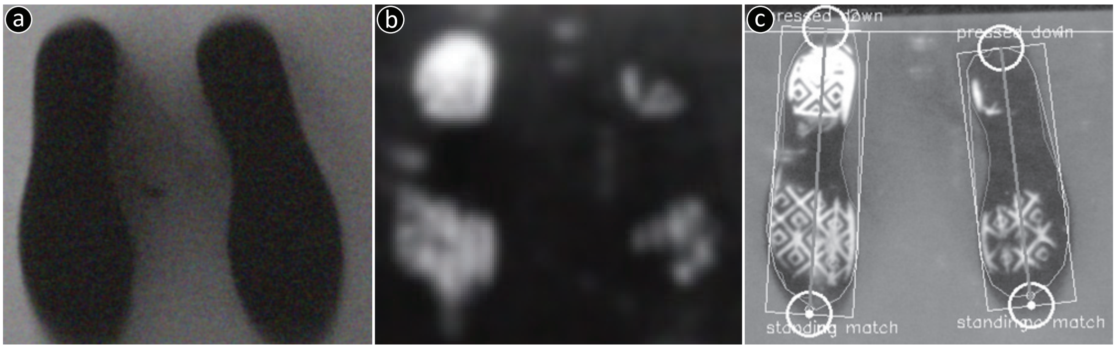


Figure 6.4: A user wearing shoes with black soles is standing on our floor prototype. (a) Front Diffused Illumination, (b) Frustrated Total Internal Reflection, (c) Front DI with FTIR, at 1 mm resolution camera.

6.2.2 *Materials and ½ m^2 Prototype*

To prototype an interactive floor, we started by assembling a large set of tables and incorporating an FTIR-based system into one of the tables. Figure 6.5 shows the first floor prototype we built to explore foot-based interaction. In order to keep material expenditure reasonable during the exploration phase, FTIR input on this prototype was limited to a small sub-region (70 cm × 50 cm). The floor's projection resolution is 0.6 mm per pixel; the camera resolution is 1.0 mm/pixel.

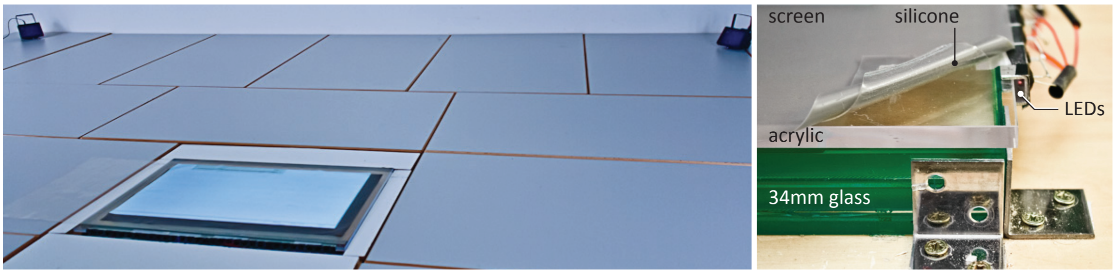


Figure 6.5: Left: Our first working prototype was composed of multiple tables. Only a sub-region of one table was interactive. (In the picture, the projection screen and compliant surface is not shown.) Right: Our prototype measures ½ m^2 and uses 34 mm safety glass, 8 mm acrylic, Tectosil 185 silicone, and a Rosco projection screen.

The setup of our *Multitoe* prototype was reminiscent of traditional FTIR systems, appropriated to support standing and walking users. Figure 6.5 (right) shows the stack-up of our floor surface. A three-layer 3.4 cm/1.34″ glass pane provides structural support, an 8 mm layer of acrylic serves as the waveguide, and a Rosco projection screen [131] creates the image. Between waveguide and screen, we use a layer of Tectosil 185 500 µm silicone as compliant surface. As with all FTIR devices [58], a compliant surface helps obtain a well-defined amount of frustration for a given amount of pressure. Tectosil 185 is a stiff silicone, which allows us to distinguish pressure at the upper end of the scale, e. g., to distinguish a user resting the entire weight on the ball of one foot from a user standing straight.

On the table prototype, we combine FTIR with front diffused illumination. In contrast to regular diffused illumination [96], front DI is ignorant of shoe color and allows us to track the shadows casts by shoes. Including front DI in our floor design produces the rough outline of users' shoes in the camera image, which facilitates extracting which part of the shoe is actually in contact with the floor (Figure 6.4a).

6.3 RESOLVING INADVERTENT ACTIVATION

Unlike tabletop devices, floor users are in contact with the floor at basically all times. To enable direct manipulation, we need a mechanism that distinguishes between intentional action (the analog to touch on tabletop) and standing/walking (the analog to hover on tabletop).

For most foot-operated devices, this distinction is handled spatially. Users step onto the gas to accelerate; in order to not accelerate, they rest their foot elsewhere. We can port the same concept to our floor design by inserting pathways of touch-insensitive areas between controls. Unfortunately, this prevents us from using large controls, such as the painting surface of a painting program. We thus need a gesture that allows users to not interact even though they are standing *on* a control.

Several alternative designs seem possible: Users could jump onto a button to activate it or stomp on it, etc. Not all of them are equally ergonomic though and it is unclear how intuitive they are. To find out what works for users, we conducted a simple user study to inform our design. Our study was inspired by the study conducted by Wobbrock et al. on the gestures users perform on tabletops [171].

6.3.1 *User Study: How to* Not *Activate a Button On the Floor*

The purpose of this study was to help design a mechanism that matches users' intuition and allows the floor to distinguish intentional user action from regular walking and standing. Participants' task was to walk across four "buttons," such that two of them would be triggered, while the other two would remain in their current state. We observed participants' strategies and interviewed them.

Interfaces and Task

As illustrated by Figure 6.6, the interface was 'implemented' using four paper buttons taped to the floor. There were a small and a large button labeled 'ok,' and a small and a large button labeled 'cancel.' Large buttons measured 40 cm×60 cm, small buttons 10 cm×10 cm.

During the study, participants walked across the four buttons. Half of the participants were tasked to "activate" the two 'cancel' buttons and get across the 'ok' buttons without activating them; the other half was instructed to activate the 'ok' buttons instead. An experimenter observed participants' strategies. Finally, participants explained their rationale in a verbal interview.

Participants

We recruited 30 participants (6 female) from our institution; they were between 21 and 29 years old.

Results

Figure 6.6 (right) shows selected participants performing the task. Together, participants demonstrated nine different strategies as shown in Figure 6.6.

Strategy	**To activate**	**To _not_ activate**	**#**
Part of foot	tap (ball only)	walk	8
	walk	_tiptoe**_	1
	walk on ball	_walk on heel**_	1
Pressure amount	stomp	walk	5
	jump onto	walk	2
	double-tap	walk	2
	dwell (stand)	_walk quickly**_	5
Left-right	right foot	_left foot**_	1
Spatial	cross center	_walk edge**_	5

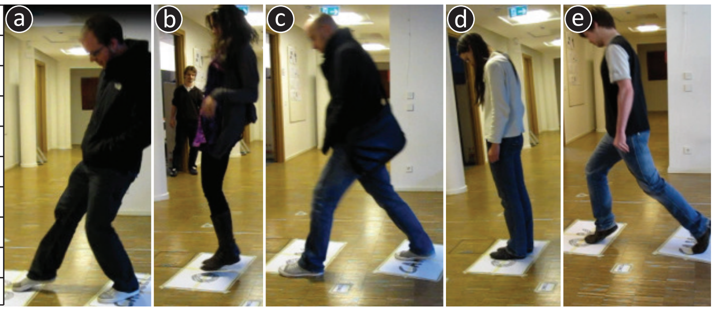


Figure 6.6: Left: Strategies and number of participants who employed them. (∗ does not work with densely packed controls, ∗∗ raises ergonomic concerns). Right: Five participants demonstrate how they activate a button: (a) tapping, (b) jumping, (c) walking on center, (d) dwelling, and (e) stomping.

Discussion

The breadth of strategies emphasizes that there is no widely accepted model for interaction using feet.

Not all demonstrated strategies are applicable to all scenarios. The strategy "walk along edge of button" fails for densely packed arrangements of tiny controls, such as the pixels of a painting program—some pixels are always hit straight on. Four other strategies raise ergonomic concerns (marked ∗∗ in Figure 6.6 left): Walking on heels and tiptoeing can get tiring over time. Activating by dwelling requires users to walk perpetually in order not to activate. Activation with the right foot requires users to hop on their left foot when crossing large controls without activating.

The remaining four strategies (*tap*, *stomp*, *jump*, and *double tap*) seem suitable. The strategy demonstrated by the largest number of participants was *tap* with 8 participants. Based on these findings, we implemented *tap* into our system.

6.4 INVOKING A MENU

We face similar requirements when designing a mechanism for invoking menus. In theory, a spatial strategy is possible, such as a toolbar or the corner buttons used by the Microsoft Surface table [99]. However, since interactive floors can become arbitrarily large, so can the distances to a stationary menu. Fixed menus therefore only make sense if replicated at a large number of locations and/or for very infrequent tasks. For the majority of tasks, users will prefer a *location-independent* interface [118].

To invoke such a context menu, we can pick any of the leftover invocation strategies from User Study 1, i. e., *stomp*, *jump*, and *double tap*. We found *jump* to offer the best recognition rate—it also virtually never occurs unintentionally. The implementation of jumping is straightforward. Our floor tracks users and if both of their feet go out of range for more than 200 milliseconds, the system invokes the menu.

6.5 SELECTING OBJECTS BY STEPPING

In order to manipulate objects on the floor, users need a basic pointing technique when using their feet for input. Since shoes occupy a substantial amount of space, they can hit objects in many different spatial relationships, such as with their edge or arch as shown in Figure 6.7. When defining a pointing technique, we need to decide which parts of a shoe should be used for hit testing.

Candidates include a point-like *hotspot* (i. e., as in Chapter 3), the entire *contact area* [26, 103], and the projection of the shoe (outline of the shoe projected onto the floor). In order to understand which of these models matches the users' conceptual model or whether we need an entirely different model, we conducted a brief user study inspired by the study we presented in Section 4.3.

User Study: Conceptual Model of Stepping

The goal of this study was to understand which area of their soles users consider to be active in targeting and should thus be considered in hit testing.

Task and Procedure

Participants stepped onto the multitouch floor with their dominant foot wearing shoes. A honeycomb grid was displayed under the participant's shoe (Figure 6.7). The cells of the grid were described to the participants as defunct "buttons." For each such "button," an experimenter asked the participant if it should be depressed based on the participant's foot position. If the answer was "yes," the experimenter "set" the respective buttons which caused it to change color (Figure 6.7). All participants completed the task in 5 minutes or less.

Apparatus

We used the ½ m^2 floor prototype shown in Figure 6.5.

Participants

We recruited a set of 20 participants for this study. Participants were between 20 and 29 years old and 6 were female.

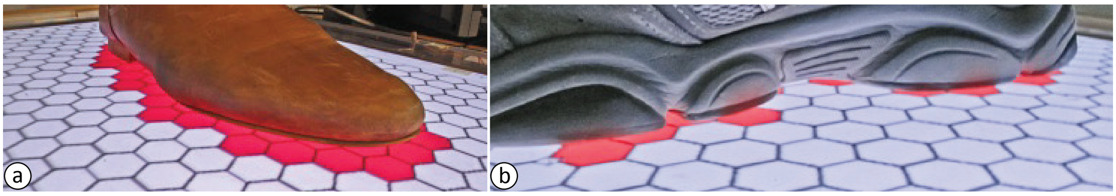


Figure 6.7: (a) 18 of 20 participants felt that the area under the arch should be included, while (b) the remaining two felt buttons under the arch should be excluded.

Results

Figure 6.8 shows shoes and button states for all 20 participants. 8 of 20 participants matched the PROJECTION MODEL, i. e., they set every button if it was covered by the projection of the participant's shoe at least at a certain percentage. This included tip and arch. Another 7 participants matched the PROJECTION MODEL, but left occasional omissions along the outline (Figure 6.8b).

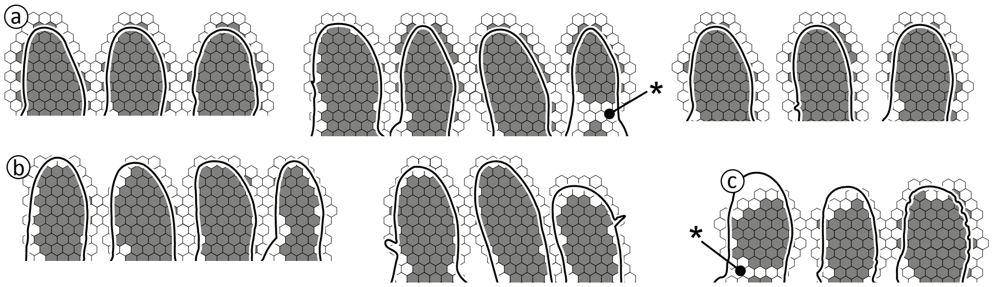


Figure 6.8: Resulting conceptual models. (a) 10 of 20 participants' model is PROJECTION; (b) PROJECTION with minor omissions (c) 3 excluded the upward curved tip. (∗) Only 2 excluded the arch.

Three participants excluded the curved up tip of the shoe (Figure 6.8c); 2 excluded the arch (Figure 6.8∗). One of them wore 5 cm heels, the other, a male participant, wore sneakers. He rationalized that the arch was not touching the floor, suggesting that his conceptual model was based on CONTACT AREA.

Note that when participants referred to contact area, they did so in an idealized way. This does not necessarily correspond to the reality on an FTIR floor, where pressure and outlines change as users change body postures over time as shown in Figure 6.9.

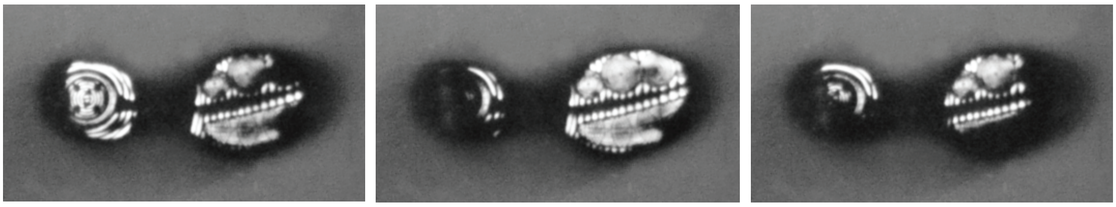


Figure 6.9: The FTIR contact area changes as the user changes posture over time.

Discussion

These finding suggest that the most common conceptual model of stepping is *projection*, even though some users erode the area a bit. The tracking model of FTIR, i. e., contact area, in contrast does generally *not* match the conceptual model of the majority of users.

We therefore implemented stepping primarily based on the front DI component of our system (the dark outline in Figure 6.9). To alleviate the sensitivity of front DI to shadows that are cast by the user's body, we combine the approach with some FTIR support.

6.6 HIGH-PRECISION POINTING USING A HOTSPOT

While the PROJECTION MODEL makes for a good default model of floor interaction, it prevents application designers from packing controls tighter than a foot. As illustrated by Figure 6.10, huge controls require users to walk between buttons or extend themselves in order to reach them. However, our purpose of switching from a table-size form factor to the floor was to allow for interaction with a large number of objects. In order to allow for complexity, we need to allow applications to create small objects and to pack them densely.

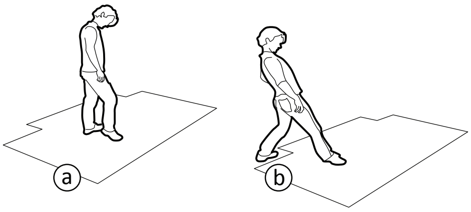


Figure 6.10: Operating a keyboard with foot-size buttons requires users to (a) walk between buttons or to (b) extend themselves in order to reach the keyboard.

In order to allow for interaction with dense clusters of on-screen objects, we introduce a high-precision mode in which users' feet are reduced to a single *hotspot*. In analogy to the previous section, we started by investigating users' conceptual model, i. e., which part of their foot they consider to be the hotspot. Is there a single global hotspot or how much variation is there across users?

6.6.1 *User Study: Conceptual Model of the Hotspot*

The purpose of this study was to survey what point on their shoe (the *hotspot*) users use to interact with point targets. We also wanted to find out how much agreement there was about the location of the hotspot: Strong agreement would suggest a single global solution, while little agreement would suggest the need for personalization.

Task

As illustrated by Figure 6.11a, participants wearing shoes stood on a "waiting" position marked with circles. (b) For each trial, a target marked with crosshairs appeared 30 cm in front of the participant. Participants placed their preferred foot onto the crosshairs, such that the foot's hotspot was located directly over the crosshairs. Participants then confirmed their selection by pressing a button on a wireless presenter tool. (c) Pressing the button recorded the floor's FTIR image of the user's foot as well as a photo of the user's foot from above. Finally, participants stepped back into the waiting position.

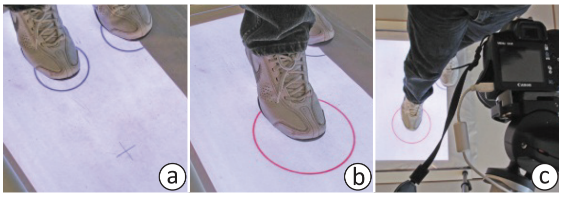


Figure 6.11: (a) When the crosshairs appeared, (b) participants stepped onto it. (c) We recorded the FTIR image and a photo from above during each trial.

Independent Variables and Procedure

Each participant performed the task in four conditions, three repetitions each. The first time, they were not given any further instructions (FREE CHOICE condition). In the other three conditions, participants were instructed to aim using the BALL of their foot, the BIG TOE of their foot, and the TIP OF THEIR SHOE. In order to prevent the more specific conditions from influencing the FREE CHOICE condition, participants always started with the FREE CHOICE condition. The order of the remaining conditions was then counterbalanced.

Participants

We recruited 24 participants (8 female) from our institution. All participants were between 20 and 29 years old. Two participants were left-footed and thus performed all trials with their left foot.

Apparatus

We again used our FTIR floor prototype, as well as a Canon EOS 1000D SLR camera to record participants' shoes from above.

Results

Figure 6.12 shows participants' hotspots mapped onto outlines of their shoes. Black dots denote contacts made during FREE CHOICE trials. The three triplets of white dots belong to SHOE TIP, BIG TOE, and BALL from tip downwards.

FREE CHOICE condition: As illustrated by Figure 6.12, we classified FREE CHOICE point triplets according to which of the other triplets they were closest to. Based on this, 7 participants seemed to aim using the tip of their foot, 6 participants used their big toe and another 6 participants used it at an offset from the toe. 2 participants used their ball and another 3 participants aimed with a point slightly above the ball.

Figure 6.13 illustrates how participants' hotspots relate to each other by overlaying shoe outlines, so that the respective hotspots (centroid of all three trials) align. The

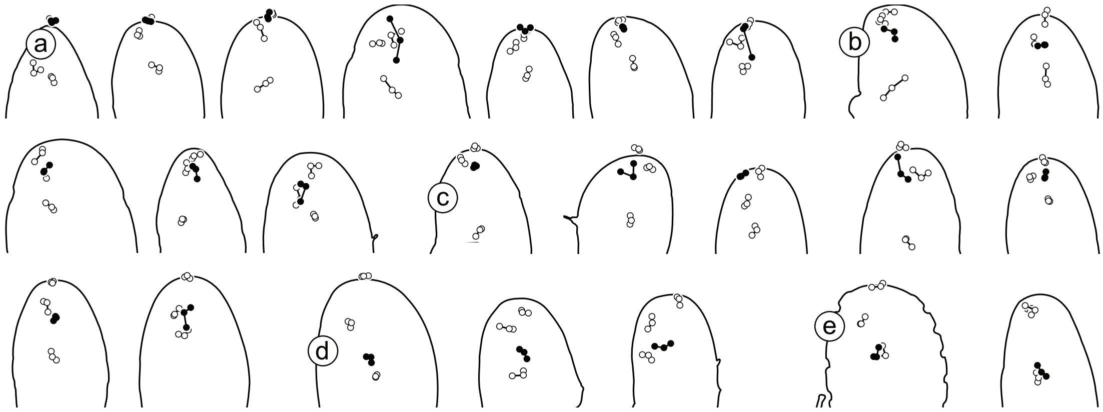


Figure 6.12: Participants acquire the target using these points on their soles (black dots: FREE CHOICE; white dots are SHOE TIP, BIG TOE, BALL from tip downwards). Participants aimed using (a) tip, (b) big toe, (c) offset from toe, (d) offset from ball, and (e) ball.

spread of 8.4 cm in Figure 6.13a suggests substantial disagreement between the FREE CHOICES of participants.

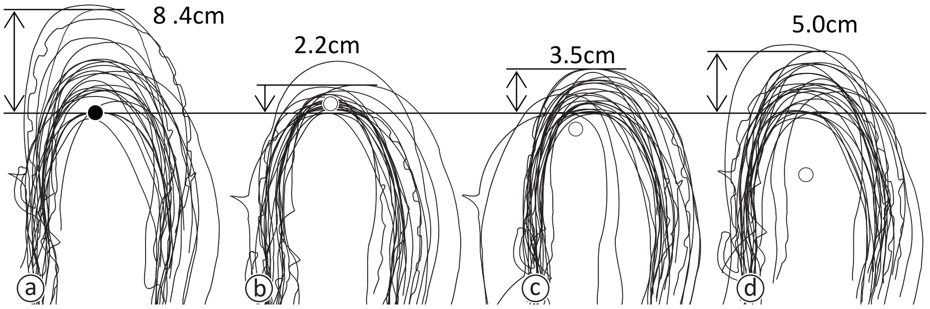


Figure 6.13: Outlines of participants' soles centered on the centroid of contact points for (a) FREE CHOICE, (b) TIP, (c) BIG TOE, and (d) BALL. The dot indicates the position of the cross.

Discussion and Resulting Implementation

The substantial spread among FREE CHOICE hotspots implies that the use of a global hotspot for all users would incur a large targeting error. This error can be reduced by instructing users to aim with a specific part of their shoe, in particular the tip, which would reduce the error to 2.2 cm in the case of the sample shown in Figure 6.13b. However, such an approach would fit the conceptual model of only 7 of the 24 participants.

To eliminate the necessity to train users to use a specific hotspot, we allow users to customize their hotspot. When users step on the floor for the first time or with a new pair of shoes, a dialog with crosshairs appears. Users then step onto the crosshairs according to their hotspot, which assigns it permanently to the respective location in the FTIR sole pattern of this pair of shoes.

6.6.2 *User Study: Targeting Precision With Custom Hotspot*

As discussed earlier, our purpose for including FTIR into floors is to allow for direct manipulation of complex applications with large numbers of objects. This implies the necessity to support interaction with small objects. In order to inform application design, we conducted another study to determine the lower bound on the size of such objects. Participants' task was to enter text using on-screen foot keyboards of three different sizes.

Interfaces

All on-screen keyboards offered 28 keys (a–z, ⟨space⟩ and '.') in a localized QWERTY layout (Figure 6.14a). All three interfaces were identical except for scale. We picked a range of sizes that would capture a wide range of error rates. Overall the three keyboards measured $52.0 \times 23.2\,\mathrm{cm}^2$, $31.0 \times 14.0\,\mathrm{cm}^2$, and $15.0 \times 6.8\,\mathrm{cm}^2$. Figure 6.14b illustrates the key sizes we used. Note that the keys on the SMALL keyboard were smaller than keys on a physical QWERTY keyboard (Figure 6.14c). Space bars were 3 times wider than regular keys.

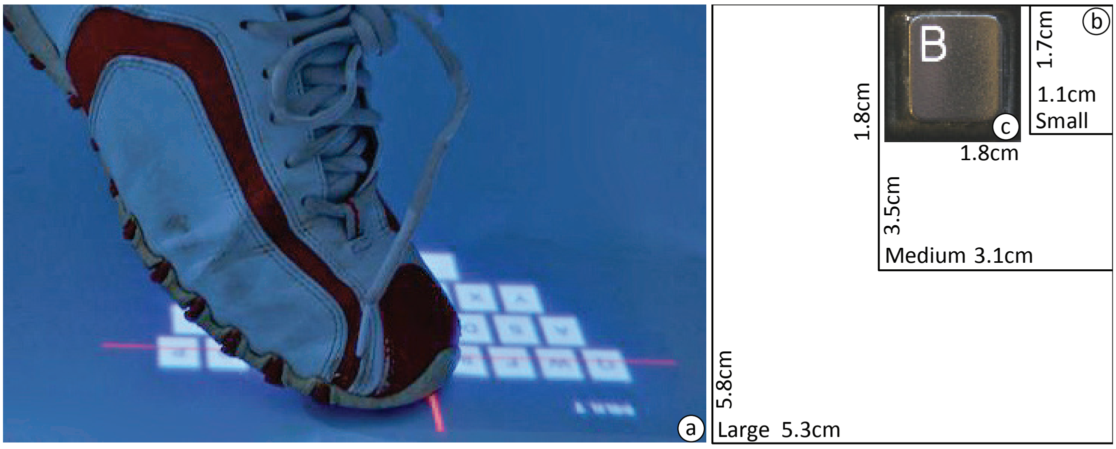


Figure 6.14: Size of the keyboards participants typed on during the study: (a) the small keyboard, (b) the sizes of the keys on the large, the medium, and the small keyboard, (c) key of a physical keyboard for reference scale. All illustrations are to scale.

Participants wore their own shoes and targeted using a self-selected hotspot. Since our goal was to study the limits of *users'* abilities, we minimized tracking-related inaccuracy by attaching an extruded dot (a ⌀11 mm nut) to the user's sole at the location of their hotspot, which eliminated remaining tracking errors.

Task

For every trial, participants entered the sentence " the lazy brown dog." (including the leading blank and the trailing period), which was shown above the keyboard. Participants typed by tapping keys with one foot, while standing on the other foot. Typing the first character started the timer. Correct key presses turned the letter in the display green, incorrect key presses red. In addition, a brief sound indicated whether a key press was correct or incorrect. Participants had to retype erroneously entered letters until they got it right, but did not have to delete erroneous entries using backspace. Typing the trailing period stopped the timer.

Procedure

Participants typed the sentence twice on each of three keyboards for an overall number of six repetitions. The order of keyboard sizes was counterbalanced. Finally, participants filled in a questionnaire. All participants completed the study in less than 10 minutes.

Participants

We recruited 26 participants (9 female) for this study. Participants were between 19 and 29 years old.

Apparatus

We used the same setup as in the previous studies.

Results

Figure 6.15 summarizes error rates and task times for the regular buttons of the three keyboards. Error rates for space bars were comparable (2.0%, 8.6%, and 23.8%). As expected, error rate and task time increased with decreasing button sizes. Note that about half of the error rate came from tapping outside the keyboard, a strategy we saw participants employ to avoid tapping incorrect letters. Task time mirrors the trends seen in error rate.

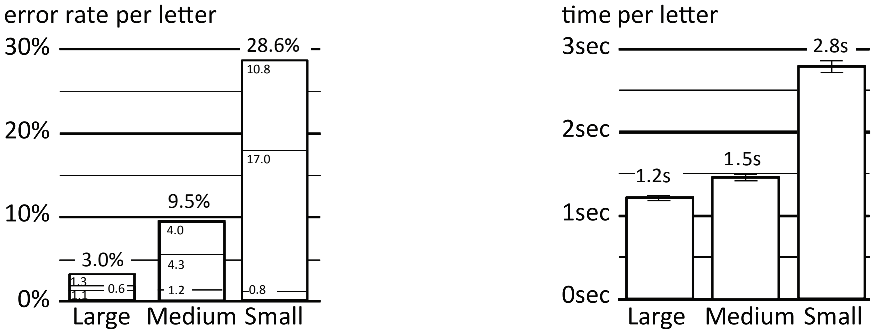


Figure 6.15: (a) error rate per letter (error types top-to-bottom: outside keyboard, neighboring key, wrong key) (b) time per letter.

Figure 6.16 shows the complete targeting data of all trials. The fact that contact point cluster centroids are centered on button centers suggests that all remaining error is indeed noise, rather than a systematic effect, such as in the case of touch input (see Figure 3.5).

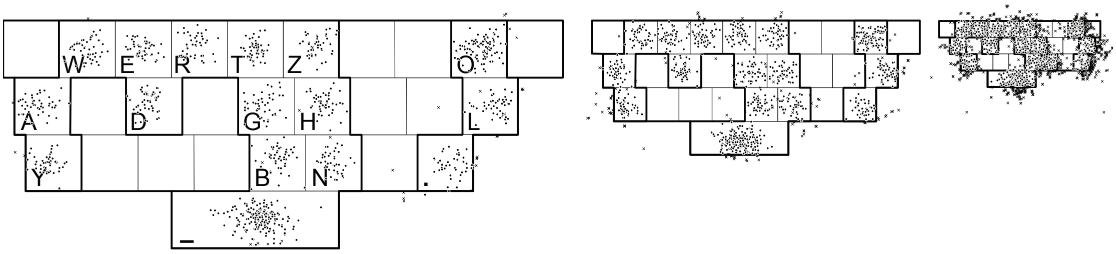


Figure 6.16: The contact points for all trials of all participants.

In the final questionnaire, half of the participants selected the LARGE keyboard as their favorite. Interestingly, 10 of 26 participants picked the MEDIUM keyboard, where they found buttons easier to *reach*. The remaining 3 participants were indifferent between the large and the medium keyboard.

Discussion

With error rates close to 30%, the 1.1 cm keys on the SMALL keyboard were clearly too small. The 3.1 cm buttons of the MEDIUM keyboard, however, might be acceptable for some applications where packing density is more important than error rate. Using a tiled layout, a 3×4 m^2 floor could pack 10,000 of such interactive objects, supporting our goal of bringing highly complex applications to multi-touch surfaces.

The large keyboard, finally, offers very good error rates below 2.9%, which is fully comparable to error rates on interactive tabletops. Note that at 5.3 cm, buttons on the large keyboard are still quite compact and an order of magnitude smaller than the 1″+ buttons required by the projection model.

This completes our effort to transition multitouch input techniques from surfaces, such as tabletops to multitouch floors. FTIR and high-resolution camera input played a key role here, e. g., because they allow the floor to distinguish users and thus personalize the interaction. High-resolution FTIR enables a range of other possibilities, such as 3d reconstruction, which we will explore in the remainder of this chapter.

6.7 algorithms for processing per-pixel pressure input

We now discuss the underlying algorithms that enable user identification and tracking of foot postures and balance. We also demonstrate how the same algorithms can be used to implement a simple type of head tracking and to enable foot interaction with high degrees of freedom.

6.7.1 *General Processing*

Figure 6.17 illustrates the pipeline we implemented to process users' shoe soles. By using higher illumination intensity for FTIR than for front DI (Figure 6.4c), we can extract the *FTIR image* from the raw image by thresholding (Figure 6.17b). (c) We extract the *DI image* by replacing the FTIR portion in the raw image with shoe color, i.e. black, and (d) find connected components on the thresholded DI image. Fitting an ellipse onto the blob determines the main axis of the sole, from whose end we determine the front of the shoe by testing which half of the convex hull of its contour is wider. The previous two steps also produce the shoe rotation. (e) The oriented bounding rectangle finally yields the width and height of the user's shoe sole.

Additional processing is done based on the requirements of the application, as we discuss in the following.

6.7.2 *Identifying Users*

We implemented a preliminary identification algorithm for our table-size prototype. Our system identifies users based on their shoe soles by searching through a database of shoe prints for a match upon observing a shoe sole on the floor. Our algorithm starts by pre-selecting candidates from the database based on similarity in sole length, width,

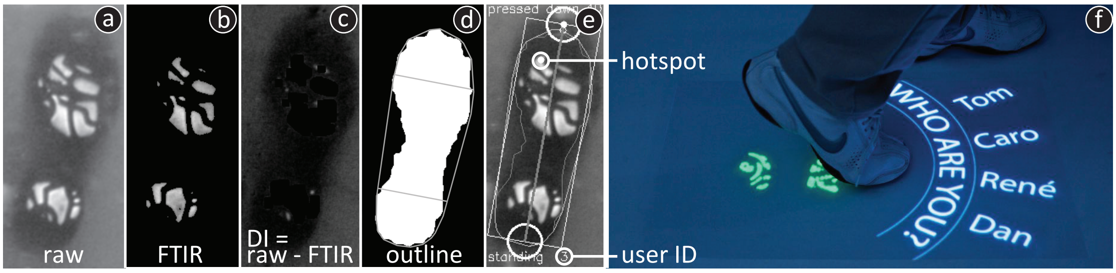


Figure 6.17: Sole processing and user identification. (a) Raw image, (b) thresholding extracts FTIR, (c) DI = raw minus FTIR, (d) Thresholding (blur(DI)) with convex hull and widths, and (e) annotated. (f) When the floor sees a pair of soles for the first time, it asks for identification.

and surface area, as well as the sole's grayscale histogram. We then perform a series of comparisons by sliding template images over the observed image to find the position with the best match in the image. If the absolute difference between the two images is below a threshold, the footprint in the database is considered a match. FTIR brings out the unique line patterns and logos that shoe makers embed into their soles, which helps recognition.

If a foot print is not recognized for several frames, it is added as a new pair of shoes to the database. At this point it is labeled *anonymous* and assigned a random ID. In addition, the system brings up a dialog that allows users to identify themselves (Figure 6.17f). This allows the floor to assign the user's name to the new shoes.

6.7.3 *Analysis of Pressure Distribution*

All other functions, such as walking vs. tapping and head tracking, are computed based on the *pressure distribution* of soles on the floor. All functions have in common that they partition the FTIR image of each foot into one or more cells, estimate the physical weight resting on each cell, and then compare this cell pressure with other cells.

Our algorithm to process pressure distributions within a shoe sole proceeds as follows. (1) Mask the FTIR image with the DI blob. (2) Partition the FTIR image into a set of cells. (3) Translate pixel color into pressure. It is important to compensate for the non-linear pressure response of FTIR by applying the inverse of the pressure-to-pixel brightness function shown in Figure 6.18 (left). We created this function by sampling the material-specific pressure response of our floor. (4) Sum up the pressure per pixel per cell. (5) Compare cell pressure with other cells.

We can implement the aforementioned functions using an appropriate partitioning of cells.

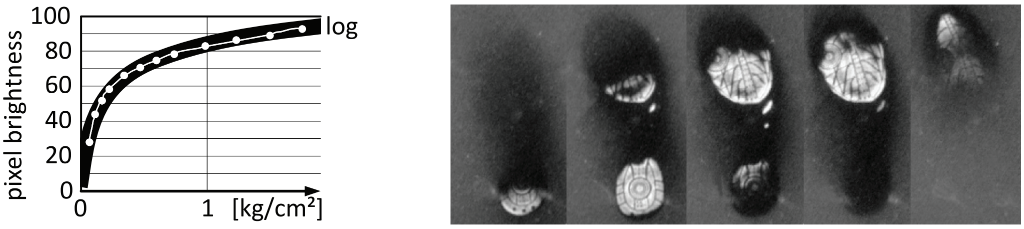


Figure 6.18: Left: We found the brightness response of our stack-up to be roughly logarithmic with pressure (acrylic waveguide with Tectosil 500 silicone and a Rosco projection screen). Right: Using FTIR, "walking forward" is identified as a heel-ball pressure sequence.

6.7.4 *Classifying Tapping Versus Walking*

In order to distinguish walking from taping, we partition the user's sole into front ("ball") and back ("heel"). Now we port the algorithm presented by Choi and Ricci [27] to FTIR: The floor observes the pressure patterns of the two partitions over time and when it sees "nothing, ball, entire foot, heel, nothing," it classifies the user as walking (Figure 6.18 right); if it sees "nothing, ball, nothing," it classifies the user as tapping.

6.7.5 *Tracking the User's Center of Gravity*

Unlike immersive and stereoscopic installations, such as *CAVEs* [29] or smart rooms [25], the position of body or head plays only a subordinate role in the context of direct manipulation scenarios. Nonetheless, FTIR-based pressure sensing allows us to obtain a simple approximation of the user's posture.

Again, we partition users' soles into front and back, which gives us four partitions whenever both feet are in contact with the floor. We determine the user's left-right balance as the pressure difference between the partitions of each foot; we determine the user's front-back balance as the pressure difference between front and back partition.

To enable *fish tank virtual reality* [161] using our prototype (Figure 6.19a), we recorded pairs of user head position and pressure distribution for the centered and extreme forward, backward, left, and right positions. This allows us to compensate for posture biases and the typical 60:40 pressure ratio between ball and heel. During the fish tank VR experience, we interpolate angles linearly.

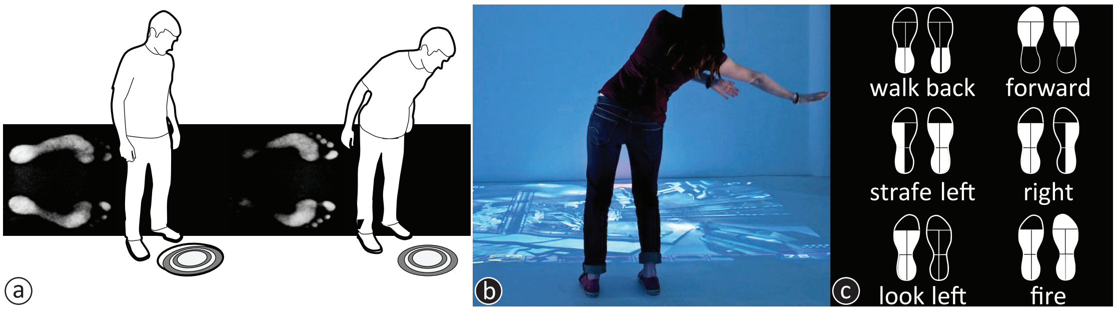


Figure 6.19: (a) Sensing pressure using FTIR enables fish tank VR. (b) This user is playing a first person shooter based on balance and foot posture (Unreal Tournament 2004). (c) Subset of the mapping between input pressure distributions and commands.

6.7.6 *Additional Degrees of Freedom*

Finally, we can create additional degrees of freedom by subdividing soles further. Figure 6.19b shows a user playing a first person shooter on our prototype, hands-free, by controlling the game using her feet alone. We obtained 10 degrees of freedom by subdividing each foot into five zones; we then use a subset of them to implement functions for moving, strafing, and shooting, a subset of which is shown in Figure 6.19c. Note that users fire and alt-fire using their left and right large toes. Surprisingly, this continues to work inside of shoes.

6.8 FLOOR SENSING IN AN ENTIRE ROOM TO RECONSTRUCT ACTIVITY

In the previous section, we described how we reconstruct the user's center of gravity from the distribution of pressure inside the user's shoe prints. We now generalize our approach to an entire room and explore how much the room can infer about its inhabitants and their activity as well as passive objects solely based on the pressure imprints people and objects leave on the floor. Similar to how we reconstructed 3D finger poses from users' fingerprints in Chapter 5, we intend to *reconstruct* the 3D configuration of people and objects from just the 2D imprints they leave on a multitouch floor.

To obtain 3D information from the 2D touch contacts that people and objects leave on the floor, we apply the same principles as in previous chapters. We first analyze the texture of 2D touch contacts and classify each touch contact to obtain the type of object that is making contact with the floor. When we detect a shoe sole or a marker pattern, the system matches it against a database to determine who the user is or which object it is, respectively. Finally, we derive which *part* of the user is in contact with the floor and what the pressure distribution is in order to reconstruct users' 3D configuration inside the room.

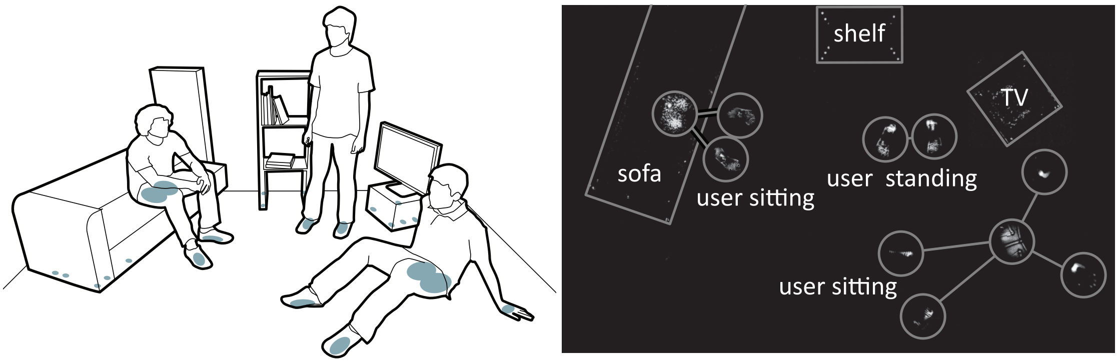


Figure 6.20: Left: Gravity pushes people and objects against the ground, where they leave *imprints* that we sense using a high-resolution per-pixel pressure sensor that spans the floor in an entire room. Right: Our system GravitySpace analyzes the 2D pressure image (here the scene on the left), which contains a set of imprints (circles, lines, and text added for clarity). From the imprints, GravitySpace reconstructs the 3D configuration of users and objects inside the room.

Our approach is based on the general principle of gravity, which pushes people and objects against the floor, causing the floor to sense pressure imprints as illustrated in Figure 6.20. Our 2D pressure sensor is thereby still limited to sensing only contact with the ground, but concludes 3D information about the objects that are in contact with it. We therefore refer to our system as *GravitySpace*.

6.9 gravityspace: 3d activity reconstruction from 2d touches in a room

Figure 6.1 shows our floor installation running GravitySpace. Three users and three pieces of furniture are on the floor, which GravitySpace detects. To illustrate what the system senses and reconstructs about the physical world, the prototype displays its understanding of the physical world using a *mirror metaphor*, so that every object stands on its own virtual reflection. Based on this mirror world, we see that GravitySpace recognizes the position and orientation of multiple users, the identity of users as demonstrated by showing their personalized avatars, selected poses, such as standing and sitting on the floor and on furniture, and tracking of leg movements to interact with virtual objects, here a soccer ball.

GravitySpace updates in real-time and runs a physics engine to model the room above the surface. All the tracking and identification shown in Figure 6.1 is thereby solely based on pressure imprints all objects leave on the floor.

Figure 6.20 (right) shows the scene as perceived by GravitySpace. This is the only information GravitySpace uses to reconstruct the scene above the ground. GravitySpace implements four main concepts: (1) recognition of poses based on classifying contact types, such as hands or buttocks, and their spatial arrangement, (2) prediction of leg movements by analyzing pressure distributions, and (3) pressure-based markers that allow GravitySpace to detect objects, such as furniture. In addition, GravitySpace recognizes users based on their shoe prints, thereby extending the concepts described in Section 6.7.2, but optimized for the 20 times larger floor size and a larger number of simultaneous users.

6.9.1 *Prototype Hardware: A Room-Size 8 m^2 Floor Installation*

Figure 6.21 shows our current GravitySpace prototype hardware, which is a essentially a scaled-up version of our previous table-size prototype (Figure 6.5 in Section 6.2.2). It also senses pressure based on FTIR using a camera located in the room below the floor surface. The interaction surface measures 8 m^2 in a single seamless piece and delivers 12 megapixels overall pressure sensing resolution at a pixel size of $1 \times 1\,\mathrm{mm}^2$. Our prototype also offers 12-megapixel back projection. While not necessary for tracking, it allows us to visualize the workings of the system as shown in Figure 6.1.

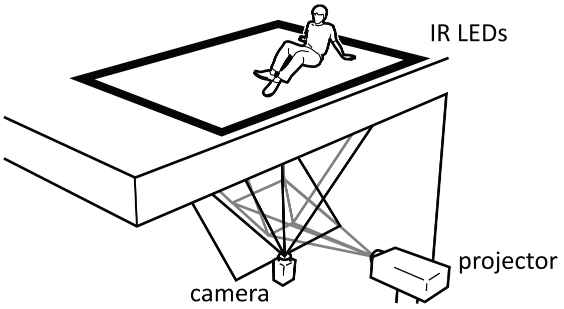


Figure 6.21: The prototype we use in GravitySpace senses per-pixel pressure input across the floor of an entire room. The system senses 25 dpi pressure input and projects across an active area of 8 m^2 in a single seamless piece.

We expect sensing hardware of comparable size and resolution to soon be inexpensive and mass available, for example in the form of a large, thin, high-resolution pressure sensing foil (e. g., UnMousePad [132]). We envision this material to be integrated into carpet and as such installed in new homes wall-to-wall. Since the technology is not quite ready to deliver the tens of megapixel resolution we require for an entire room, our FTIR-based prototype allows us to explore our vision of tracking based on pressure imprints *today*.

6.9.2 *Pressure-Transmitting Furniture*

To allow our floor installation to not only sense people when they are in direct contact with the floor, but also detect them when they are sitting on passive objects, such as furniture, we complemented our floor prototype with custom-made furniture. While such furniture could use active pressure sensing [105], we have created *passive* furniture that *transmits* high-resolution pressure down to the floor, rather than *sensing* it actively. This offloads sensing to a single centralized active sensing component, in our case the floor itself. Passive furniture also reduces complexity and cost, while the absence of batteries and wires makes them easy to maintain.

Everyday furniture already transmits pressure, but on a level that is too coarse to resolve fine-grained structure. Current furniture imprints are limited to representing overall weight and balance. While locating the center of gravity has been demonstrated by many earlier research systems (e. g., such as VoodooIO [155]) or commercial systems (e. g., Wii Balance board), this limits our ability to detect activities taking place on top of the furniture, such as sitting on a sofa.

In order to recognize identity and poses of the object on top in more detail, we have created the seating furniture shown in Figure 6.1. All the pieces transmit pressure in comparably high resolution. We accomplish this by using an array of "transmitters." Transmitters have to offer sufficient stiffness to transmit pressure (and also to support the weight of the person or object on top). Figure 6.22a shows a cube seat we constructed: Using regular drinking straws as transmitters makes the furniture light and sturdy, but allows propagating pressure input from the top of the seat to the bottom. 1,200 straws (8 mm in diameter) fill each cube seat; 10,000 fill the sofa, which is based on the same principle. Straws are inexpensive (e. g., € 80 for filling the sofa). The backrest and armrest of the sofa are pressure-sensitive as well—they are filled with longer "sangria" drinking straws. We obtain their curved shape by cutting the straws to length a layer at a time using a laser cutter.

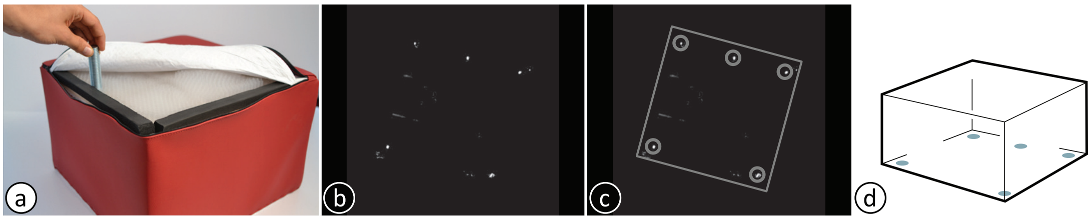


Figure 6.22: (a) Each cube seat is filled with 1,200 drinking straws. Here, one of the steel rods that form the marker pattern is inserted. (b) The cube leaves this pressure imprint on the floor. GravitySpace detects the (c) marker points and recognizes (d) the cube.

Straws are held together by a frame made from 15 mm fiberboard. We stabilize the straws in the box using a grid made from plywood connected to the frame, which essentially subdivides the box into 3×3 independent cells. The grid minimizes skewing, thus preventing the box from tipping over. We cover the bottom of the box with Tyvek, a material that crinkles but does not stretch. This prevents the bottom from sagging, yet still transmits pressure. In addition to the leather, we add a thin layer of foam as cushioning to the top of the cube seats for added comfort.

Weight shifts on top of the box can cause the box to "ride up" on the straws, which can cause an edge of the box to lose traction with the ground. To assure reliable detection, we create markers from weight rods that slide freely in plastic tubes; the tubes themselves are held in by the Tyvek. We use an asymmetric arrangement of rods to assign a unique marker ID to each piece.

6.10 ALGORITHMS

Figure 6.23 summarizes the pipeline we implemented to process touches that occur on our floor, including recognizing, classifying and tracking events, users and objects based on the pressure patterns they leave. We thereby build on the algorithms we described in Section 6.7 and optimize them to process the entire 12 megapixel image in real-time (25 fps).

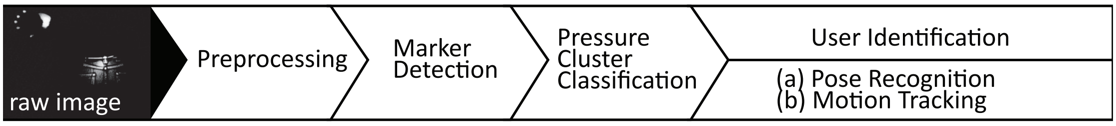


Figure 6.23: GravitySpace processes all input using this pipeline to recognize and track users, and objects.

GravitySpace recognizes objects with 2D textures, such as body parts or shoe prints by extracting the imprint features they leave in the raw pressure image. For objects with little discernible texture or changing texture (e. g., due to users sitting on furniture), we add features using pressure-based markers.

GravitySpace implements three main functions: (1) 3D Pose reconstruction by classifying 2D pressure clusters, (2) estimating users' joint locations in 3D based on 2D pressure distributions and inverse kinematics and (3) user identification based on shoe prints. The following sections detail our processing pipeline and explain how we reconstruct all this information from the pressure intensities the platform observes.

6.10.1 *Step 1: Pre-Processing Pressure Images*

All processing starts by thresholding the pressure image to remove noise after subtracting a static background. Our algorithm then segments this image and extracts continuous areas of pressure by finding connected components. In the next step, GravitySpace merges areas within close range, prioritizing areas that expand towards each other. We call the result *pressure clusters*. A pressure cluster may be for example a shoe print or the buttocks of a sitting user. GravitySpace then tracks these clusters over time.

6.10.2 *Step 2: Identifying Furniture Based on Markers*

Pressure imprints of larger objects, such as furniture, provide little distinguishable texture on their own. In addition, the overall texture of seating furniture changes substantially when users sit down. GravitySpace therefore uses dot-based pressure-markers to locate and identify furniture. Figure 6.22b shows the imprints of a sitting cube that we equipped with markers. These markers produce five or more points and are arranged in a unique spatial pattern that is rotation-invariant. We designed and implemented marker patterns for a sofa, several sitting cubes, and shelves.

To recognize markers, GravitySpace implements brute-force matching on the locations that have been classified as marker points, trying to fit each registered piece of furniture into the observed point set and minimizing the error distance. To increase the stability of recognition, our implementation keeps objects whose marker patterns have been detected in a history list and increases the confidence level of recently recognized objects. We also use hysteresis to decide when marker detection is stable based on the completeness of markers and their history confidence.

6.10.3 *Step 3: Classifying Pressure Clusters Based on Image Analysis*

For each pressure cluster in the camera image, GravitySpace analyzes the probability of being one of the contact types shown in Figure 6.24. GravitySpace distinguishes hands, knees, buttocks, and shoes, thereby further distinguishing between heel, tip, and edge of a shoe. These probability distributions are an essential part of the subsequent pose recognition. Areas covered by furniture pieces are ignored for this classification in order to minimize noise.

In order to classify each pressure cluster, GravitySpace extracts 16 fast-to-compute image features from the respective area in the image, including image moments, structure descriptors using differences of Gaussians, as well as the extents, area, and aspect ratio of the bounding box around the cluster. We trained a neural network that assigns probabilities for each type of contact to each cluster.

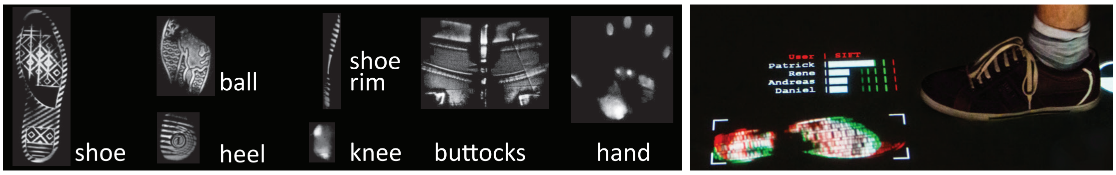


Figure 6.24: Left: GravitySpace assigns each pressure cluster the probability of being one of these contact types. Right: GravitySpace uses Harris corner detection and SIFT to match detected shoe prints against a database of registered users.

6.10.4 *Step 4: Identifying Users Based on Shoe Prints*

Whenever the users' feet are in contact with the ground—for example when standing or sitting, but not when lying—GravitySpace will recognize users by matching their shoe prints against a database of shoe soles associated with user identities. As users register with both of their shoes, our approach also distinguishes left and right feet.

Due to the large floor area, our previous approach to user identification as explained in Section 6.7.2 is not sufficient on GravitySpace. Identifying shoe soles using simple template matching does not scale reliably to the area GravitySpace needs to support, both in terms of speed as well as accuracy.

To match shoe prints with the same resolution as previous systems (1 mm per pixel), GravitySpace uses an implementation of SIFT [91] that runs on the GPU to achieve interactive rates. Using Harris corner detection as the feature extractor and SIFT as the descriptor algorithm allows us to match shoes with rotation invariance. To identify a user by the shoe sole, GravitySpace counts the number of features that match in each of the shoe images in the database and the observed shoe print as shown in Figure 6.24. A feature thereby matches if the angular distance between the two descriptor vectors is within close range. Since the number of detected features varies substantially between different sole patterns, we normalize the distance by dividing by the maximum number of features in either observed or database image.

Stitching a Shoe Imprint from a Sequence of Frames

When user walk on the floor, only a small part of their shoe appears in the image at first and then becomes larger as their shoe sole rolls over the floor from heel to toe (Figure 6.18 right). Since the camera consecutively captures images of each such partial shoe imprint, GravitySpace merges all partial observations in successive frames into an aggregated imprint, which allows us to capture an almost complete shoe sole. This concept is also commonly used to obtain a more encompassing fingerprint by rolling a finger sideways while taking fingerprints.

Recovering Shoe Orientation Based on Phase Correlation

To predict the location of the user's next steps when walking, GravitySpace leverages the orientation of the shoes that are on the floor. GravitySpace determines shoe orientations directly after matching shoe prints by registering front and back of each database shoe print with the observed shoe on the floor. Our system transforms both shoe prints into spectrum images and applies log polar transforms to then compute the translation vector and rotation angle between the two shoe prints using phase correlation. All shoes in the database thereby have annotated locations of heel and toes, which happens automatically upon registration by analyzing the direction of walking.

6.10.5 *Step 5a: Pose Recognition Based on Spatial Configurations of Pressure Clusters*

To classify body poses from the observed pressure imprints, GravitySpace performs pose matching based on the location and classified type of observed pressure clusters. For example, GravitySpace observes the spatial configuration of pressure clusters shown in Figure 6.25a, i. e., the imprints of buttocks, two feet, and two hands as a user is sitting on the floor.

To match a pose, GravitySpace uses a set of *detectors*, one for each pose that is registered with the system (Figure 6.25c). Each detector is a set of rules based on contact types and their spatial arrangement. GravitySpace currently distinguishes five poses: standing, kneeling, sitting on the floor, sitting on cube seat or sofa, and lying on a sofa.

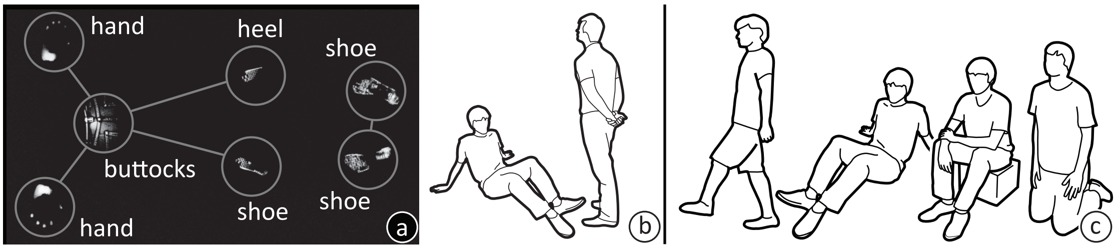


Figure 6.25: (a) Based on the classified pressure clusters and their spatial arrangement, GravitySpace recognizes (b) a sitting and a standing user. (c) GravitySpace currently detects these four poses.

GravitySpace feeds all pressure clusters to all detectors. Each detector creates a set of hypotheses. Each hypothesis, in turn, contains a set of imprints that match the pose described by the detector. For example, hypotheses returned by the sitting detector contain buttocks and two feet. Optionally, there may also be two hands if users support themselves while leaning backwards as shown in Figure 6.25a. Each detector returns all possible combinations (or hypotheses) of imprints that match the pose implemented by this detector. Each hypothesis thus explains a subset of all imprints. We compute the probability of a hypothesis by multiplying the classification probabilities of all contained imprints with a pose-specific prior.

From these *individual* hypotheses (explaining a single pose), we compute a set of *complete* hypotheses; each complete hypothesis explains all detected imprints by combining individual hypotheses. We calculate the probability of a complete hypothesis as joint probability of individual hypotheses, assuming that individual poses are independent from each other. We track complete hypotheses over multiple frames using a Hidden Markov Model with complete hypotheses as values of the latent state variable.

6.10.6 *Step 5b: Tracking Based on Pressure Distributions*

If possible, GravitySpace also tracks body parts that are *not* in direct contact with the floor, such as the locations of feet above the ground while walking or kicking. GravitySpace also builds on the tracking of body tilt that we explained in Section 6.7.5, for example when a user leans left or right while sitting. In GravitySpace, this allows for predicting the user's steps before making physical contact with the floor, which reduces the tracking latency of our system. It also allows sensing interactions that occur above the floor, such as users interacting with virtual objects. Obviously, our approach cannot sense events taking place in mid-air, such as raising an arm or changing the gaze direction.

We estimate the location of in-air joints by analyzing the changing centers of gravity within each pressure cluster. We then try to best fit a skeleton to the computed locations of all joints using a CCD implementation of inverse kinematics. GravitySpace finally visualizes the reconstructed body poses with 3D avatars as shown in Figure 6.1.

Deriving the Location of Feet Above the Ground

To infer the location of users' feet when they are above the ground, GravitySpace reconstructs its location by analyzing the changing pressure distribution of the *other* foot, which is on the ground as shown in Figure 6.26a–b. Our algorithm first calculates the vector from the center of pressure of the cluster aggregated over time to the center of pressure of the current cluster. This vector corresponds to the direction that a person is leaning towards, and is used to directly set the position of the foot in mid-air.

We again derive a skeleton using inverse kinematics, which enables animating the remaining joints of the avatar for output.

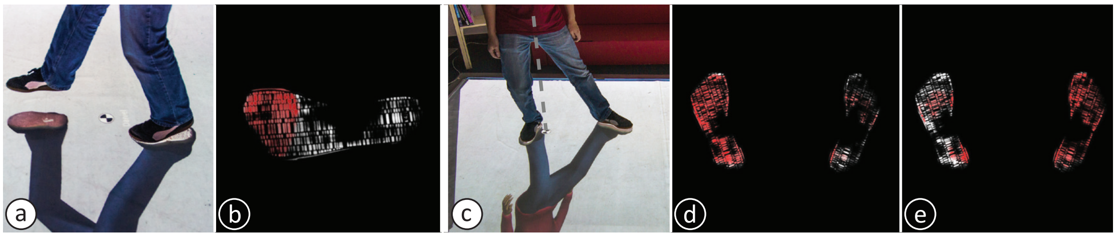


Figure 6.26: GravitySpace derives (a) the location of a foot above the floor based on (b) the pressure distributions of the user's other foot. (c) We track the user's center of gravity based on the joint center of pressure of both feet (d) and (e).

Tracking Body Tilt

To track the user's body tilt, GravitySpace observes multiple pressure clusters as shown in Figure 6.26c–e. The system first computes the joint center of pressure over all pressure clusters of a user by summing up zero and first order moments of the individual pressure images. We then exploit that the center of pressure directly corresponds to a body's center of gravity projected on the floor. Once the center of gravity is determined, GravitySpace sets the corresponding endpoints of the skeleton's kinematic chains; all other joints then follow automatically based on the inverse kinematic.

6.11 EVALUATION

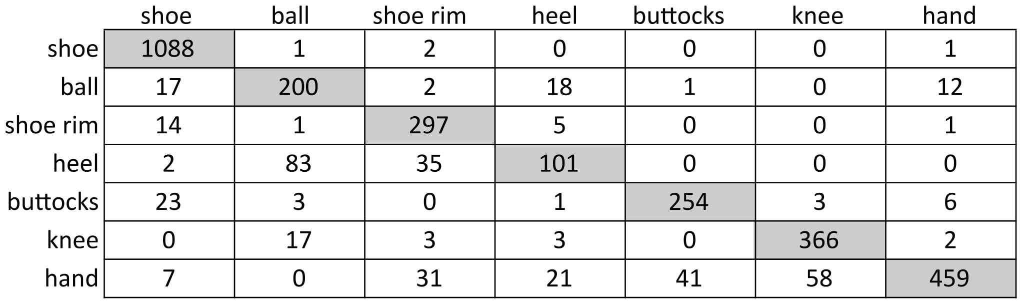
We conducted a technical evaluation of three system components: pressure cluster classification, user identification, and pose recognition. In summary, the algorithms of our prototype system allow for (1) distinguishing different body parts on the floor with an accuracy of 92.62% based on the image analysis of 2D pressure clusters, (2) recognizing four 3D body poses with an accuracy of 86.12% based on type and spatial relationships between 2D pressure clusters, and (3) identifying 20 users against a 120-user database with an accuracy of 99.82% based on shoe-print matching.

6.11.1 *Pressure Cluster Classification*

To evaluate pressure cluster classification, we trained a neural network with data from 12 participants, and tested its classification performance with data from another four participants.

Training Data

We asked 12 participants to walk, stand, kneel, and sit on the floor, in order to collect data of the seven different contact types required for pose recognition, namely hand, shoe (we distinguish between the entire shoe, ball, rim, and heel), knee, and buttocks. In total, we collected 18,600 training samples.



	shoe	ball	shoe rim	heel	buttocks	knee	hand
shoe	1088	1	2	0	0	0	1
ball	17	200	2	18	1	0	12
shoe rim	14	1	297	5	0	0	1
heel	2	83	35	101	0	0	0
buttocks	23	3	0	1	254	3	6
knee	0	17	3	3	0	366	2
hand	7	0	31	21	41	58	459

Table 6.1: Confusion matrix for pressure cluster classification.

Test Data

Following the same procedure, we collected data from another four participants for testing. This resulted in 3,127 samples.

Evaluation Procedure

We manually annotated all training samples to provide ground truth. We then fed the test data into the trained neural network, taking the contact type with the highest probability as outcome. Note that our algorithm does not discard the probability distributions provided by the neural network, but feeds them into the following pose recognition as additional input.

Results

Our approach achieved a classification accuracy of 86.94% for the seven contact types shown in the confusion matrix of Table 6.1. If the entire shoe, ball, rim, and heel are grouped and treated as a single contact of type *shoe*, as done by the pose recognition, classification accuracy reaches 92.62%.

6.11.2 *Pose Recognition*

We evaluated our pose recognition implementation with five participants. Since pose recognition is based on descriptors of spatial contact layouts, no training data was required for this evaluation.

Test Data

We collected data from five participants, who each performed the four poses shown in Figure 6.25c, including standing/walking, sitting on the floor, sitting on furniture, and kneeling. For each participant and pose, we recorded a separate pressure video sequence.

Evaluation Procedure

To provide ground truth, we manually annotated all frames with the currently shown pose. We then ran our algorithm on all frames of the recorded videos, and compared the detected poses to ground truth annotations.

Results

86.12% ($SD = 13.5$) of poses were correctly identified within a time window of 1.5 s as tolerance. In comparison, FootSee achieves recognition rates of 80% (tested with a single subject) for five standing only activities [175].

6.11.3 *User Identification*

We determined the user identification accuracy of our implementation with 20 participants.

Registration

To populate the user database, each participant walked in circles for about 35 steps on the floor. GravitySpace now selected one left and one right shoe print for each participant, choosing the shoe print with the minimum distance in the feature space compared to all other shoe prints of the same participant and foot.

Test Data

After a short break, participants walked a test sequence of about 60 steps. Shoe prints were in contact with the floor for an average of 0.92 s (*SD* = 0.13). Participants then did another round. This time, however, they were instructed to walk as fast as possible, resulting in a sequence of about 70 steps with a lower average duration of 0.38 s (*SD* = 0.11).

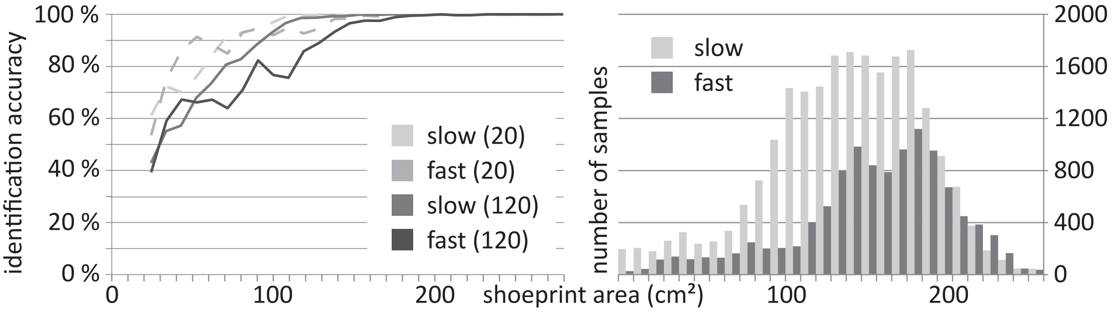


Figure 6.27: Identification accuracy for walking slow and fast, using databases of 20 and 120 users. Aggregating multiple frames leads to more complete shoe prints and better identification accuracies.

Evaluation Procedure

We evaluated the identification performance by running our algorithms on the recorded test data. Obviously, the slower participants walked, the longer their feet were in contact with the floor, and the more frames were available. The part of the foot that is in contact with the floor varies while walking, rolling from heel to toe. As described above, our algorithm reconstructed shoe prints by merging successive pressure imprints. We ran our identification algorithms on all aggregated imprints with an area greater than 30 cm^2, which is the minimum area for discernible shoe contacts as determined during the previous pressure cluster evaluation.

Results

We evaluated the test set against two databases, one containing 20 study participants, and one enlarged with data from 100 additional people, who were lab members and visitors. Figure 6.27 shows the identification accuracy using these two databases for both walking slow and fast. As expected, larger shoe prints aggregated from more frames resulted in better recognition. For the 20-user database, the classification accuracy reached 99.94% for shoe prints with an area between 180 and 190 cm^2 (the average area of shoe prints). When walking fast, recognition rates slightly dropped to a maximum classification accuracy of 99.19%. This is expected as shoe prints were more blurry.

We then reran the classification against the 120-user database. Our approach correctly identified shoe prints with 99.82% accuracy prints when walking slowly, and with 97.56% accuracy when walking fast. In comparison, the Smart Floor by Orr and Abowd identifies users based on their footstep force profiles and achieves recognition rates of 93% for 15 users [111]. Qian et al. correctly recognize 94% of 10 users based on gait analysis [122].

Speed

On average, feature extraction took 47.7 ms (SD = 11.4) per shoe print, which is independent of the number of registered users. Identification took 251.4 ms (SD = 81.2) using a database of 120 users. Each additionally registered user currently increases the runtime by 2 ms.

To maintain a frame rate of 25 fps, GravitySpace runs user identification asynchronously. Before identification is completed, users are tracked based on heuristics (e. g., distance and orientation of shoe prints). Once identified, user tracking relies on this information. To reduce delays due to identification, GravitySpace caches recently seen users: New contacts are first compared to this short list of before falling back to the entire participant database.

6.11.4 *Benefits and Limitations*

Compared to traditional camera-based solutions, the pressure-based floor sensing approach we employ in GravitySpace offers four main benefits.

First, floor-based sensing provides consistent wall-to-wall coverage of rooms. In contrast, camera-based systems have a pyramid-shaped viewing space. Motion capture installations resolve this by leaving space along the edges, which is impractical in regular rooms and leads to uneven or spotty coverage. Floor-based tracking can be flat, integrated into the room itself, and provides consistent coverage across the room.

Second, floor-based tracking is less susceptible to occlusion between users. The perspective from below is particularly hard to block—simply because people tend to stand next to each other, resulting in discernible areas of contact. From a more general perspective, the benefit of pressure sensing is that mass is hard to "hide." Mass has to manifest itself somewhere, either through direct contact or indirectly through another object it is resting on. Camera-based systems, in contrast, may suffer from users occluding each other if the cameras mounted in one spot (e. g., LightSpace [170]). Systems distributing multiple cameras around a room still suffer from dead spots, such as in the midst of groups of users [101].

Third, floor-based sensing allows for the use of simpler, more reliable recognition algorithms. Our approach reduces the recognition problem from comparing 3D objects to comparing 2D patterns, because all objects are flat when pressed against a flat surface. Objects can therefore only change in their 2D translation, 1D rotation in the plane, and the pressure intensities within, which allow us to match objects using algorithms from digital image processing [31, 50].

Fourth, pressure-based tracking is less privacy-critical. While floor-based sensing captures a lot of information relevant to assisted living applications (e. g., [82]), it never captures actual photos or video of its inhabitants. This mitigates privacy concerns, such as recording users while getting dressed or using the bathroom.

On the other hand, our floor-based approach is obviously limited in that it can recognize objects only when they are in direct contact with the floor. While we reduce the impact of these limitations using 3D models based on inverse kinematics, events taking place in mid-air can obviously not be sensed, such as the angle of an arm being raised or a user's gaze direction. The approach is also inherently subject to lag in that the floor learns about certain events only with a delay. We cannot know the exact position of a user sitting down until the user makes contact with the seat. As we place the avatar in between, it is subject to inaccuracy.

6.12 CONCLUSIONS

In this chapter, we have presented high-resolution FTIR-based floors as a means to create interactive multi-touch surfaces beyond the size of tabletop computers. We made first steps toward enabling the interaction model of touch, i. e., direct manipulation, on such a floor. FTIR played the key role in this process, as it allowed us to reliably distinguish interactions from walking and because it provides the high accuracy required to acquire and manipulate small objects. We argue that the combination of small objects (3–6 cm, comparable to objects on tabletops) with the dramatically larger scale of interactive floors forms an interesting platform for enabling complex applications that deal with ten-thousands of objects.

Combining high-resolution FTIR with a projected floor also resulted in the design of additional interactions and capabilities for indoor sensing. One of them is user identification based on sole patterns, which works in part because users are bound to floors by gravity—very different from tabletops. We showed that our floor allows classifying input events into body parts or passive objects and that we can analyze the pressure distribution inside such contacts to extract more degrees of freedom from input events on multitouch floors.

Finally, we have shown how to use the 2D per-pixel pressure image we sense to track people and furniture in 3D on a high-resolution 8 m^2 multitouch floor. While our sensor is limited to sensing *contact* with the surface, we have demonstrated how to conclude a range of objects and events that take place *above* the surface, i. e., in the 3D space of the room. This includes reconstructing 3D user poses and 3D interactions with virtual objects above the floor. We also demonstrated how to extend the range of this approach by sensing *through* passive furniture that propagates pressure to the floor.

7 CONCLUSIONS AND NEXT STEPS

In this dissertation, we have shown how to enable flat 2D devices to sense 3D input upon touch and implement 3D natural user interfaces. Offering interaction through such 3D interfaces had previously been limited to either stationary installations or setups that impede mobility. With our contributions, we bring this type of sensing to the familiar flat form factor of mobile devices. Since the fact that these devices are flat is the reason that has allowed them to achieve mass adoption and mobility in the first place, we have enabled 3D natural user interfaces to also find application in mobile scenarios.

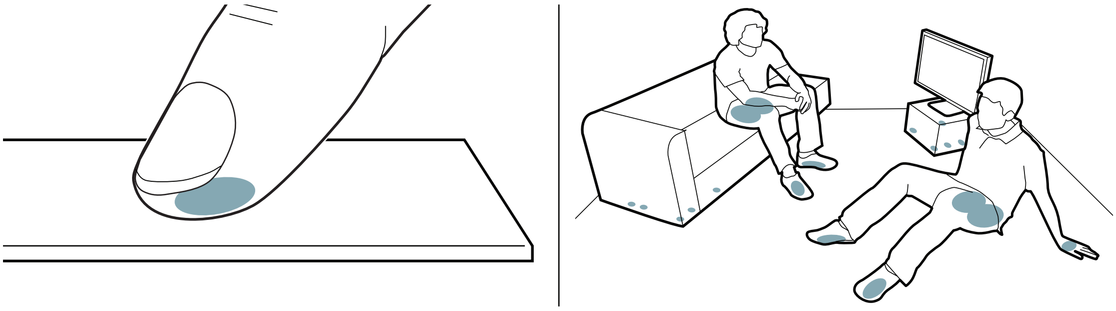


Figure 7.1: Our main contribution is reconstructing 3D information from 2D touch input. We analyze the 2D imprints of all touch contacts at a high resolution, which provides us with the *texture* of each object. We then determine which *part* of the object caused the visible imprint to reconstruct its 3D pose—a feature in the 3D space above the surface. Left: In the case of finger input, the texture represents the user's fingerprint, which we use to identify the user biometrically and to reconstruct the 3D finger pose. This allows touch devices to detect touch input with three times higher accuracy than existing devices. Right: On multitouch floors, we analyze the texture to identify users based on their shoe soles, to detect furniture, and to classify all imprints to derive users' 3D body poses using the arrangement of imprints and inverse kinematics.

7.1 MAIN CONTRIBUTION: RECONSTRUCTING 3D INFORMATION FROM 2D INPUT

Our main contribution is that we obtain the 3D information by reconstructing it from the data captured by flat devices: 2D data from a touch sensor. We have demonstrated our approach on small and table-size platforms, commonly operated through touch input using fingers. We have also shown that our approach generalizes beyond such devices to larger touch devices, such as smart rooms that detect touch from users' feet or other body parts, as well as passive objects.

The core element of our approach, reconstructing 3D from 2D touch, is considering touch input not only as spatial input modality, but also extracting the *texture* each touching object leaves on a surface as shown in Figure 7.1. The texture of each contact thereby is a manifestation of the user's body part that is touching the device in a particular 3D input pose. By analyzing the 2D texture and classifying its contents, we can reconstruct additional degrees of information about the objects that touch the device, i. e., objects that live in the 3D space around it.

One engineering outcome of our approach is touch input as a high-precision input modality on touch devices. By analyzing touch contacts on a fingerprint level, identifying the user, and reconstructing the user's finger in 3D, we have substantially increased the accuracy with which devices detect spatial touch input. Our approach allows users to reliably touch targets as small as 4.3 mm per side, which is a threefold increase in accuracy compared to all mobile devices that are in use today. With respect to the size of elements in graphical user interfaces, this allows devices to reliably and accurately sense users' touch input for elements that are an order of magnitude smaller. These results indicate that touch input really is a 3D input operation; since current touchscreen devices implement touch detection as an operation in 2D, they perceive additional degrees of freedom as input error. This explains why touch input, so far, has been considered an inaccurate input modality.

By introducing our new 3D perspective on touch, we have made a contribution in the theoretical domain. By investigating how users *conceptually* acquire targets using touch, we derived the perspective that users target by aligning *visual* features with a target on the screen—not by using the contact area between their finger and the device itself, as implemented in all current mobile touch devices. Our new model thereby not only explains the effects we exploit in the high-precision touch devices we presented, but also explains the error perceived by current touch devices as an effect of parallax. While users align visual features on the *top* of their fingers with touch on the screen, current devices observe the touch contact, i. e., a feature at the *bottom* of the users' fingers.

To demonstrate the feasibility of our approach on touch*screens*, we presented our prototype Fiberio, which incorporates displaying output and scanning fingerprint input into the *same* surface. The implementation of such a device has been hypothesized in the human-computer interaction community for over a decade but never realized. Fiberio achieves these desired capabilities using a new type of surface material for touchscreens: a fiber optic plate. This material enables Fiberio to use the surface for both displaying output as well as capturing the reflections needed for fingerprint scanning. By analyzing the 2D fingerprint, Fiberio identifies each user upon touch and reconstructs the 3D rotation of their finger in real-time. This allows Fiberio to implement high-precision touch input on a fully interactive touchscreen device.

As part of precise touch, Fiberio implements biometric user identification. This finally enables touchscreens to perform reliable touch-to-user association in collaborative settings as well as authentication on a per-user-interface element level. It also makes Fiberio suitable for all scenarios that involve single-display groupware.

The concepts we described in this dissertation transcend the prototype touch devices we built for the purpose of demonstration and studying. While they have a certain form factor due to the requirement of sensing touch input at a fingerprint level, our goal remains to incorporate our concepts into devices that can achieve broad adoption. For mobile devices, the determining factor of success has been the fact that they are flat. To implement our concepts on mobile devices, we thus need to reconsider the touch sensor itself to provide input at a fingerprint resolution.

One type of touch sensing technology that has the potential to be a suitable hardware platform is in-cell sensing, which places photocells between regular screen pixels of a screen. In-cell screens perceive the reflection from objects above the display similar to ThinSight [65], but at a much higher resolution, which allows them to sense the diminutive structure of fingerprints. A few commercial products have started to use in-cell sensing (e. g., Microsoft PixelSense [100]), albeit at resolutions that are an order of magnitude too low for high-quality fingerprint scanning. Sharp showed an image of a fingerprint captured on a small 2.6″ touchscreen using in-cell technology at VGA input resolution [23].

While it is unclear whether the quality and resolution of this technology suffices for reliable processing, the nature of the technology itself is promising. Once in-cell sensors achieve the size and resolution required to provide fingerprint scanning on mobile devices, they will have the potential to implement 3D from 2D reconstruction along with all the outcomes that we have presented in this dissertation. This, in turn, will also allow the contributions of our dissertation to be implemented on mass-available devices.

By extending 3D from 2D reconstruction to a pressure-sensitive multitouch floor, we have demonstrated that our approach transfers from small touch devices to very large touch surfaces. We showed how the touch principles known from table-size systems transfer to large multitouch floors, including touch interaction and precise input using feet. We also showed that even though input on floors is substantially different from that on regular touch devices, our system GravitySpace implements the same concepts that we described for regular touch devices to perform 3D from 2D reconstruction. As gravity forces all objects onto the floor, GravitySpace senses the resulting 2D imprints to recognize users and other objects in the room in 3D, including users' 3D body poses. Therefore, GravitySpace provides the type of sensing required for 3D natural user interfaces on a flat surface, which will be beneficial in environments that require this form factor, such as future homes.

All of the devices we have proposed in this dissertation implement a variation of 3D natural user interfaces by sensing input on a *surface*. In conjunction with the developments of touch sensors outlined above, this will enable these devices to assume a flat form factor and forgo additional sensors, such as the commonly-used cameras, which substantially add to the size of current setups and makes them comparably bulky. By *reconstructing* 3D information from 2D surfaces instead, a single sensor is sufficient to provide data and can be integrated directly into the device.

Therefore, reconstructing 3D from the 2D touch allows us to accomplish the type of sensing required for applications with 3D natural user interfaces on the very surfaces we stand on or the surfaces that we interact with through touch. Our approach allows us to use *flat* touch sensors and therefore sensors that can be embedded into everything: large surfaces, such as walls and floors, as well as small and mobile surfaces.

7.2 FROM LARGE DEVICES TO FLAT DEVICES TO ULTRA-SMALL DEVICES

In this dissertation, we have presented a solution to advancing systems that implement 3D natural user interfaces from their current space-requiring form factors to flat form factors. Such flat touch devices maintain the ability to sense 3D input by reconstructing it from the 2D touch contacts they observe. As shown in Figure 7.2, our contributions allow devices to shrink substantially in their *z* dimension. In our future work, we plan on exploring next-generation devices that will additionally shrink in their *x* and *y* dimension, thus overall becoming diminutively small.

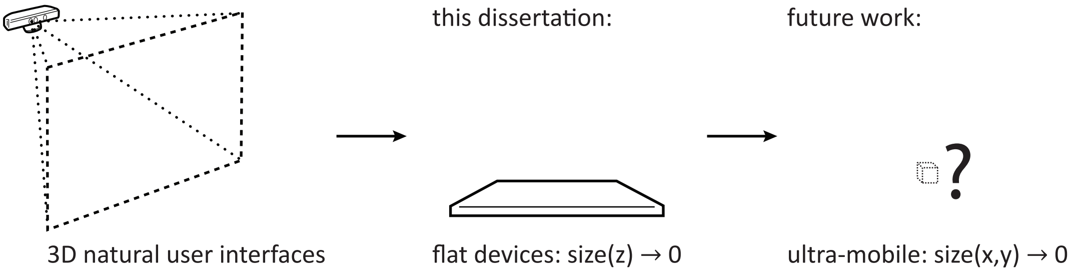

this dissertation:
future work:
3D natural user interfaces
flat devices: size(z) → 0
ultra-mobile: size(x,y) → 0
?

Figure 7.2: 3D natural user interfaces originated in stationary installations. In this dissertation, we showed how to sense 3D input on flat devices using 2D touch input. For future work, we plan on exploring the interaction with ultra-mobile devices that virtually disappear because of their size.

Two Forces That Drive Form Factors: Mobility vs. Content Consumption

The miniaturization of mobile devices has recently come to a halt, even though interactive devices have traditionally continued to shrink in size. Over the past few decades, computers have become smaller and now high-performance technology fits in the form factor of a laptop or even tablet. However, this development has stopped, which is a result of the two opposing forces that have driven recent developments and that have now reached a sweet spot. One force has propelled miniaturization with the benefit of *mobility*; because of ever-shrinking form factors, devices are now so small that users carry them in their pockets, such as mobile phones, or wear them on their body, such as wrist watches. The other force has been the increasing use of mobile devices for *consuming content*; continuously increasing device sizes have allowed users to enjoy media content that benefits from large resolutions on mobile devices, such as photos and videos. Larger mobile devices also aid users who create content, by offering more space for interaction.

Extrapolating into the future, we wonder what will be the next force in driving the evolution of device sizes. Assuming that all three dimensions of future miniature devices will trend towards zero, one resulting question is what the viable form factors will be. The goal of our future work is to advance this process and to miniaturize users' interactive devices so as to still allow those devices to support the tasks offered by current mobile devices.

We envision a future in which users will get used to technology that is so small and unobtrusive that it might become reasonable to wear miniature devices at all times. While devices that are smaller than current mobile devices are within our grasp, such as smart watches, we are interested in exploring what will happen ultimately.

To start our exploration, we chose a scenario whose context demands such fundamental changes and thus requires rethinking interaction with miniature devices on a fundamental level: implanted devices. In the next chapter, we present an initial investigation of this topic and embed it in the context of ultra-small future mobile devices.

8 OUTLOOK FOR ULTRA-MOBILE DEVICES

This chapter is based on results published in [69].

The transition of interaction with computers from desktop workstations to mobile devices has been comparably obvious. Users now communicate with one another and access information on mobile devices—anytime and anywhere.

At the same time, however, there has been another, almost invisible transition to miniature technology in the medical domain: People have started to receive small implanted devices for medical purposes, such as pacemakers and hearing aids. While such devices are invisible to other people, about 1% of the population in the United States, for example, has implanted pacemakers [51].

In this chapter, we investigate how we could support people that will have future implanted devices, explore how users might interact with them, and extend the capabilities of such implants.

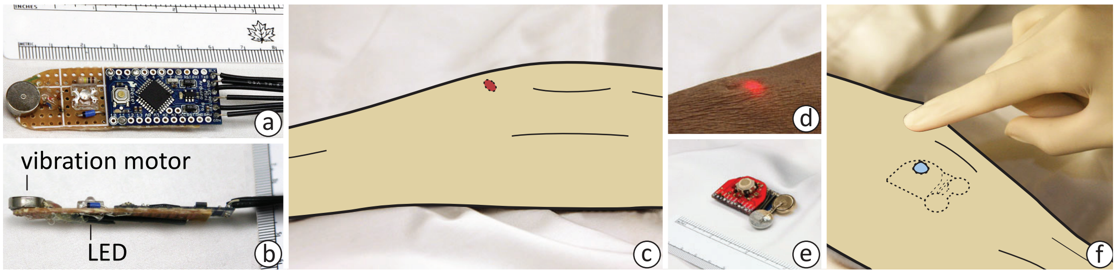


Figure 8.1: Implanted user interfaces allow users to interact through human skin. (a–b) These output devices are implanted underneath the skin of (c) a specimen arm. (d) Actual photograph of the LED output through the skin. (e) This standalone prototype senses input from an exposed trackball and (f) illuminates in response.

8.1 INTERACTION WITH DEVICES IMPLANTED UNDERNEATH HUMAN SKIN

Implanted devices currently do not support interactive tasks. To check on the status of their pacemakers, for example, users cannot directly interact with the device, but need to see a physician. If a pacemaker runs out of battery, the patient needs replacement surgery, which is not only expensive, but comes along with all the risks and side effects of surgery.

Since it is unclear how a user might interact with an implanted device directly, we explore four core areas of interfaces that implanted devices provide: accepting input from the user, providing output to the user, communicating wirelessly with external devices, as well as wireless powering to avoid the need for physical cable connections.

After discussing the space of possible solutions to each of these four challenges, we perform a technical evaluation, where we surgically implant seven devices into a specimen arm. We evaluate and quantify the extent to which traditional interface components, such as LEDs, speakers, and input controls work through skin (Figure 8.1b and c). Our main finding is that traditional interface components do work when implanted underneath human skin, which provides an initial validation of the feasibility of implanted user interfaces.

Motivated by the results, we conduct a small qualitative evaluation using a prototype device (Figure 8.2 left), for the purpose of collecting user feedback. As a substitute for actually implanting this device in a live person, we place it under a layer of artificial skin made from silicon, which affixes on the user's skin (Figure 8.2 right).

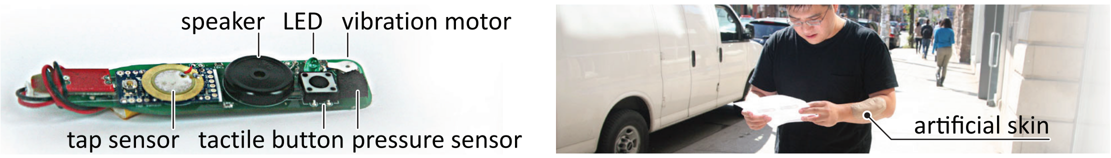


Figure 8.2: Left: We covered a prototype device with a layer of artificial skin to collect qualitative feedback from use in an outdoor scenario (right). Participants received output triggers through the artificial skin and responded by providing input through the artificial skin.

The insights from our work extend beyond interactive implanted devices and find applications to miniature mobile devices as well. Implanted devices share a number of the properties that recent developments in mobile and wearable technology [146] have been geared towards: Devices along with the information they store always travel with the user; there is no need for manually attaching them and the user can never forget or lose them. Thus, implanted devices are available at all times [136].

We conclude our exploration of implanted user interfaces with a comprehensive discussion of medical assessment, limitations, and a final outlook.

8.2 design space of implanted user interfaces

We consider *implanted devices* as devices that are surgically and permanently inserted under the human's skin. Implanting devices that possess user interfaces allows users to *directly* interact with them. This enables such devices to support a wide range of applications and tasks, beyond the medical usages prevalent today.

Implanted devices have several advantages over mobile and wearable devices. First, users typically carry mobile devices inside pockets. Retrieving them to start interacting imposes a significant overhead on usage time [9]. As devices shrink to smaller sizes, researchers have started to incorporate them into clothing [112, 80, 124] and users have started attaching them directly to their bodies [146, 147]. Implanted devices, in contrast, do not need to be manually attached to the user's body. They stay out of the way of everyday or recreational activities (e. g., swimming or showering). Second, implanted devices have the potential to be completely *invisible*. This would avoid any social stigmas of having such devices. Third, implanted devices, along with the information they store and provide, always travel with the user; the user can never lose or forget to take them. The devices and applications become *part* of the user.

Despite these potential benefits, there has been little or no investigation of implanted user interfaces from a human-computer interaction perspective. Given the continuous miniaturization of technology [107], we believe implanted user interfaces could become a reality in the future. Below, we outline some of the core design considerations, with the hope of bringing these issues to the attention of the HCI community.

8.2.1 *Design Considerations*

We see four core challenges associated with implanted user interfaces and their use through human skin: 1) providing input to and sensing input on implanted devices, 2) perceiving output from and producing output from implanted devices, 3) communication among implanted devices and with external devices, and 4) power supply to implanted devices.

8.2.2 *Input Through Implanted Interfaces*

Since implanted devices sit under the skin, they are not directly accessible through their interfaces. This makes providing input to them an interesting challenge.

One option is to use contact-based input through the skin, such as a button, which would additionally offer tactile and audible feedback to the user compared to regular taps on various locations of the arm [60]. Tap and pressure sensors allow devices to sense how strongly touches protrude the skin, while brightness and capacitive sensors detect a limited range of hover. Strategic placement of touch-based sensors could form an input surface on the skin that allows for tapping and dragging. Audio is an alternative implanted user interface. A microphone could capture speech input for voice activation.

Fully implanted and thus fully concealed controls require users to learn their locations, either by feeling them through skin or by indicating their location through small marks. Natural features such as moles could serve as such marks. Partial exposure, in contrast, would restore visual discoverability and allow for direct input. Exposing a small camera, for example, would allow for spatial swiping input above the sensor (e. g., input generated by the user's tongue [72, 137]). All such input components, whether implanted or exposed, are subject to accidental activation, much like all wearable input components. Systems have addressed this, for example, by using a global on/off switch or requiring a certain device posture [62].

Alternatives to *direct* interaction include sensing input from the body. Users can provide input through flexing their muscles [136] or by focusing their thoughts on one particular aspect (brain-computer interfaces [172]).

8.2.3 *Output Through Implanted Interfaces*

Device output typically depends on the senses of sight (i. e., visual signals in the form of projection), hearing (i. e., audio signals), and touch (e. g., vibration and moving parts). Stimulation of other senses, such as taste and smell, is still only experimental (e. g., taste interfaces [144]).

The size constraints of small devices require sacrificing spatial resolution and leave room for only individual visual signals, such as LEDs. Furthermore, visual output may go unnoticed if the user is not looking directly at the source. While audio output is not subject to such size constraints, its bandwidth is similar to the visual output of a single signal: varying intensities, pitches, and sound patterns [107]. Tactile output of single elements is limited to the bandwidth of pressure to the body and intensity patterns (e. g., vibration [174]). Tactile feedback may be particularly suited towards implanted user interfaces, since it could provide output noticeable only to the host user and no one else.

Rather than providing direct output, implanted devices could stimulate the user's body to provide output to the user. The user would notice such output through self-observation, such as device-induced body motions (electro stimulation [151]), a changing sense of balance [44], or speech production [24].

8.2.4 *Communication and Synchronization*

To access and exchange data amongst each other or with external devices, implanted devices need to communicate.

If devices are fully implanted under the skin, communication will need to be wireless. Bluetooth is already being used to replace wired short-range point-to-point communication, such as for health applications (e. g., in body area networks [77] for in- and on-body wireless communication [153]). Wi-Fi, as an alternative, transmits across longer distances at higher speeds, but comes at the cost of increased power usage and processing efforts.

Equipping implanted devices with an *exposed port* would enable tethered communication. Medical ports are already used to permit frequent injections to the circulatory system [86]. Ports and tethered connections are suitable for communication with external devices, but not amongst two devices implanted at different locations in a user's body. Such devices would still require wireless communication.

8.2.5 *Power Supply Through Implanted Interfaces*

A substantial challenge for implanted devices is how they source energy. As power is at a premium, implanted devices should employ sleep states and become fully active only after triggering them.

A simple way to power an active implanted device is to use a replaceable battery. This is common with pacemakers, which typically need surgical battery replacement every 6–10 years. Rechargeable batteries would avoid the need for surgery, and recharging could be wireless, through technology known as inductive charging [121].

If the implanted device is close to the skin surface, either powering devices completely through the skin (e. g., RFID implants [95]) or inductive charging may work through the skin [102]. Alternatively, an exposed port could provide tethered recharging to an implanted device.

Finally, an implanted device could harvest energy directly from using the device [11] or from body functions (e. g., heartbeats [90] or body heat [148]). We direct the reader to Starner's overview for more information [148].

8.2.6 *Summary: Challenges of Implanted Interfaces*

We have described some of the key challenges, and discussed possible components that could support the interface between the human and the implantable. However, there is little understanding of how well these basic interface components actually function underneath human skin.

8.3 EVALUATING THE USER INTERFACES OF INTERACTIVE IMPLANTED DEVICES

The purpose of this evaluation was to examine to what extent input, output, communication, and charging components remain useful when implanted underneath human skin. In addition, we provide a proof of concept that these devices can in fact be implanted, both fully under the skin and with exposed parts.

We performed this evaluation in collaboration with the Department of Surgery in the Division of Anatomy at the University of Toronto, Canada. The procedure of the study underwent full ethics review prior to the evaluation and received approval from the Research Ethics Board.

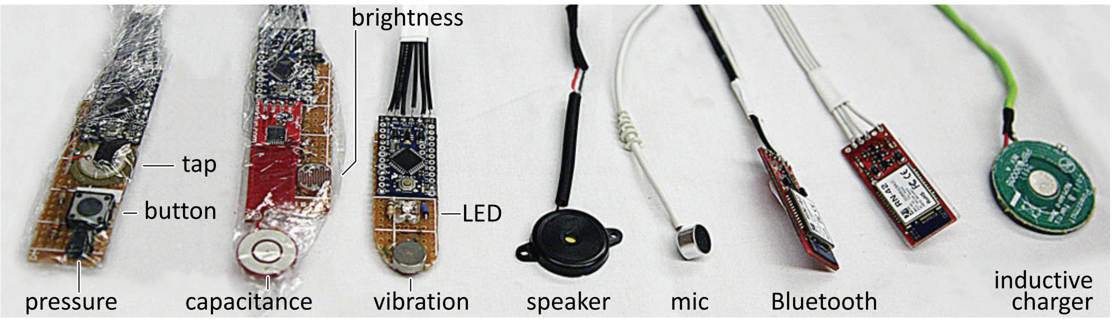


Figure 8.3: These devices were implanted during the study. Plastic bags around devices prevent contact with tissue fluid.

8.3.1 *Devices*

We evaluated seven devices featuring twelve controls in total, which were traditional input and output components as well as components for synchronization and powering common in conventional mobile devices. As shown in Figure 8.3, we tested three basic sensors for direct touch input: button, pressure sensor, tap sensor. In addition, we tested two devices that could potentially detect hover above the skin: capacitive and brightness sensor. We also tested a microphone for auditory input. For output, we tested an LED (visual), vibration motor (tactile), and speaker (audio). For charging, we evaluated an inductive charging mat, and for communication, we tested Bluetooth data transfer. These devices do not exhaust all possible implanted interface components, but we chose them as some of the more likely components that could be used.

Cables connected each of the devices to a laptop computer to ensure reliable connectivity and communication with the devices throughout the study (Figure 8.4). The laptop logged all signals sent from the input components on the devices, including device ID, sensor ID, sensed intensity, and timestamp. The laptop also logged time-stamped output triggers, including output component ID, intensity, and frequency.

All active devices used an ATmega328 microcontroller with a 10-bit precision AD converter. The chip forwarded all measurements to the laptop and also computed the length of impact as well as average and maximum intensities. We also recorded all background intensities separately.

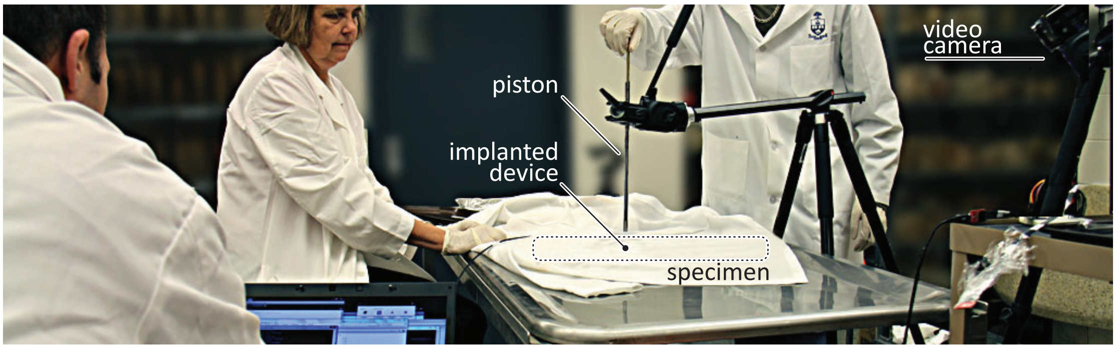


Figure 8.4: Study setup including the study apparatus to evaluate input components. A piston repeatedly dropped from controlled heights onto the sensors to produce a controlled input signal.

8.3.2 *Experimenters*

The study was administered by an experimenter and an experimenter assistant, both with backgrounds in human-computer interaction, and an anatomy professor, who carried out all of the surgical procedures (Figure 8.4). Because the focus of this study was on the technical capabilities of the devices themselves, external participants were not necessary.

8.3.3 *Procedure*

We conducted the evaluation in two sessions. In the baseline session, the devices lay on the table shown in Figure 8.4. In the implant session, each of the seven devices was implanted into a cadaveric specimen, one at a time. An external video camera documented the entire implant session and parts of the baseline session. The experimenter configured and initialized the devices through the laptop and monitored the incoming data, while the assistant performed the necessary interactions with the devices.

8.3.4 *Medical Procedure*

One lightly embalmed cadaveric upper limb specimen (dark-skinned male, 89 years old) was used for this study. With light embalming, the tissues remained pliable and soft, similar to fresh and unembalmed tissue [5]. The skin and subcutaneous tissues remained mobile.

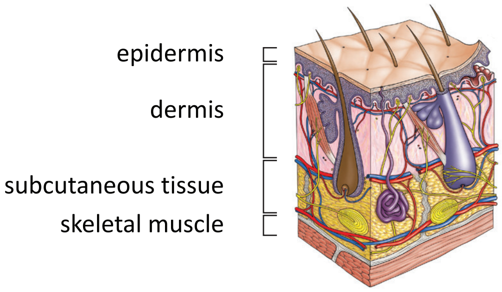


Figure 8.5: Illustration of skin layers. All devices were implanted between the skin and the subcutaneous fatty tissue.

Each of the seven devices was enclosed by two thin transparent plastic bags to prevent malfunction due to penetration by tissue fluid (as shown by the left-most two devices in Figure 8.3). To insert devices, the skin was incised and separated along the tissue plane between the skin and underlying subcutaneous tissue at the cut end of the limb, about 7.5 cm proximal to the elbow joint, which was 20 cm from the insertion point. Once the plane was established, a long metal probe was used to open the plane as far distally as the proximal forearm, creating a pocket for the devices. Each of the devices was inserted, one at a time, into the tissue plane and the wires attached to the devices were used to guide the device into the pocket between the skin and subcutaneous tissue of the proximal forearm (Figure 8.5). Distal to the insertion site of the device, the skin remained intact. All devices were fully encompassed by skin, with no space between device and skin or tissue, or any opening.

8.3.5 *Study Procedure and Results*

We now report the study procedure along with results separately for each of the seven devices.

Touch Input Device (Pressure Sensor, Tap Sensor, Button)

To produce input at controlled intensities, we built a stress test device as shown in Figure 8.4. The assistant dropped a piston from controlled heights onto each input sensor to produce predictable input events.

For the pressure and tap sensors, the piston was dropped from six controlled heights (2 cm to 10 cm in steps of 2 cm), repeated five times each, and the intensities from the sensors were measured. For the button, the piston was dropped from seven heights (3 mm, 7 mm, 1 cm, 2 cm–10 cm in 2 cm steps), also repeated five times each, and we recorded if the button was activated. Subjectively, the piston dropping from 10 cm roughly compared to the impact of a hard tap on a tabletop system. Dropping from 1 cm produces a noticeable but very light tap.

Apparatus Details

The pressure sensor used a voltage divider with a circular 0.2″ Interlink Electronics force sense resistor (100 g–10 kg) and a 10 kΩ resistor. The button was a 12 mm (4.3 mm high) round PTS125 hardware button. The touch sensor was a Murata 20 mm piezoelectric disc. The microcontroller captured events at 30 kHz. The piston was a 60 g metal rod.

Results

For all three touch-input sensors, we recorded the peak input intensities reported by each component. We then averaged the observed peak values across all repetitions to compensate for noise in our measurement process.

Force Sensor

Skin softened the peak pressure of the dropping piston, whereas the softening effect shrunk with increasing impact force (Figure 8.6 left). We analyzed the measured voltages and, by relating them back to the force-resistance mapping in the data sheet, obtained an average of 3 N in differences of sensing impact between conditions.

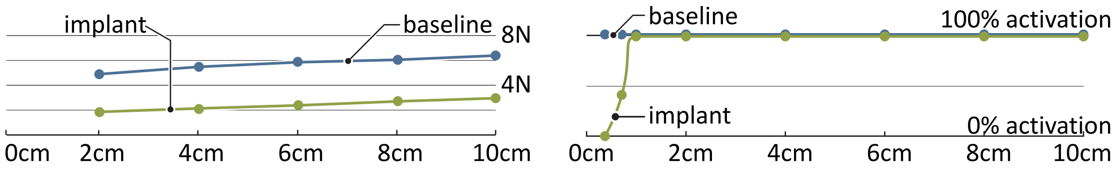


Figure 8.6: Left: On average, skin accounts for 3 N overhead for impact forces on pressure and touch sensors. Right: The piston activated the button from all tested heights in the baseline condition, but activated the button reliably only from a height of 1 cm and up when implanted.

Button

Figure 8.6 (right) illustrates the effect of skin dampening on the impact of the dropping piston. In the BASELINE condition, the piston always activated the button, whereas only dropping from a height of 1 cm and higher achieved enough force to activate the button through the skin at all times when IMPLANTED.

Tap sensor

In both conditions, the piezo tap sensor produced the maximum voltage our device could measure in response to the impact of the piston from all tested heights. The piston therefore activated the tap sensor reliably with all forces shown in Figure 8.6 (left).

8.3.6 *Hover Input Device (Capacitive and Brightness Sensor)*

To produce hover input, the assistant used his index finger and slowly approached the sensor from above over the course of 3 s, rested his finger on it for 3 s, and then slowly moved his finger away. The assistant repeated this procedure five times for each of the two sensors.

Apparatus Details

The capacitive sensor was a 24-bit, 2-channel capacitance to digital converter (AD7746). The brightness sensor used a voltage divider with a 12 mm cadmium sulfide 10 MΩ photo resistor and a 10 kΩ resistor. Both sensors captured hover intensities at 250 Hz. Three rows of fluorescent overhead lighting illuminated the study room.

Results of the Brightness Sensor

We smoothed the raw measurements by averaging the five curves of measured signal intensities to account for noisy measurements. Without the finger present, skin diffused incoming light, which resulted in reduced brightness (Figure 8.7 left). The environmental light explains the differences in slopes between baseline and implant condition; as the finger approaches the sensor, light reflected from surfaces can still fall in at extreme angles in the baseline condition. Skin in contrast diffuses light and thus objects approaching the sensor result in a less pronounced response.

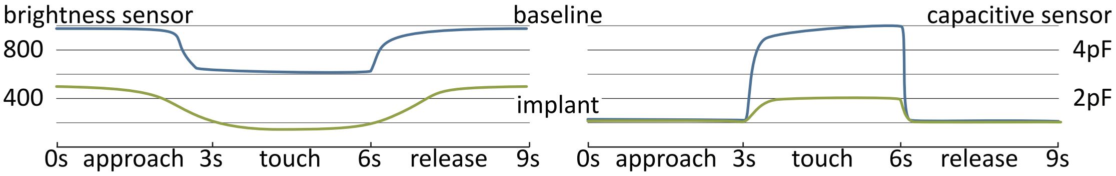


Figure 8.7: Impact of the skin on (left) sensed brightness and on (right) sensed capacitance. Curves average the values of all five trials.

Results of the Capacitive Sensor

As with the brightness sensor, we averaged the five measured series to compensate for noisy measurements. Similar to the brightness sensor, the capacitive levels were offset when sensing through the skin (Figure 8.7 right). The signal of a touching finger was comparably strong in the baseline condition, but caused only a milder difference in sensed capacitance through the skin.

8.3.7 *Output Devices (Red LED and Vibration Motor)*

To evaluate the LED and motor, we used a descending staircase design to determine minimum perceivable intensities [28, 89]. For each trial, the experimenter triggered components to emit output at a controlled intensity level for a duration of five seconds. The assistant, a 32 year old male, served as the participant for the staircase study to determine absolute perception thresholds. The method started with full output intensity, which the participant could clearly perceive. The experimenter then decreased the intensity in discrete steps, and the participant reported if he could perceive it. If he did not, the experimenter increased output intensities in smaller steps until the participant could perceive it. We continued this procedure until direction had reversed eight times [78, 89]. The last four reversal values then determined the absolute perception threshold [78].

At each step, we also measured the actual output intensities. We captured the LED output with a camera focusing onto the LED at a fixed distance, aperture and exposure time as shown in Figure 8.8 (left). An accelerometer placed directly above the vibration motor captured output intensities (Figure 8.8 right).

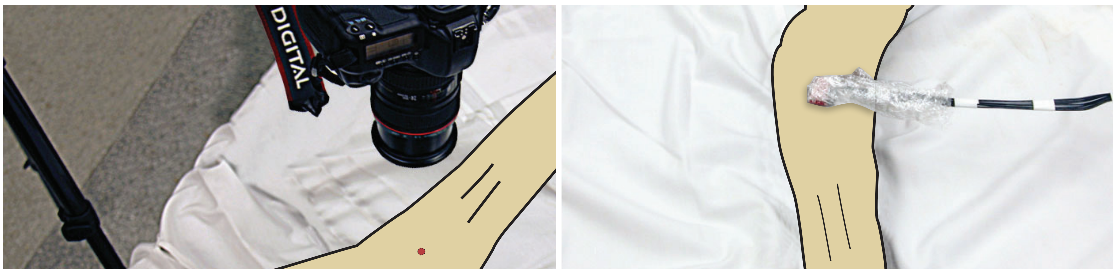


Figure 8.8: Left: A camera captured the intensity of produced light and an accelerometer measured vibration intensities (right).

Apparatus Details

The LED was a red 3000 mcd square light. The vibration motor was a 10 mm (3.4 mm high) Precision Microdrives shaftless 310-101 vibration motor. The external camera was a Canon DSLR EOS5D and captured 16-bit RAW images.

Results of the LED

The staircase methodology yielded the absolute threshold for perceiving LED output at 8.1% intensity required in the BASELINE condition and 48.9% intensity required through the skin in the IMPLANTED condition. Figure 8.9 (left) shows the actually produced intensities determined by the external camera.

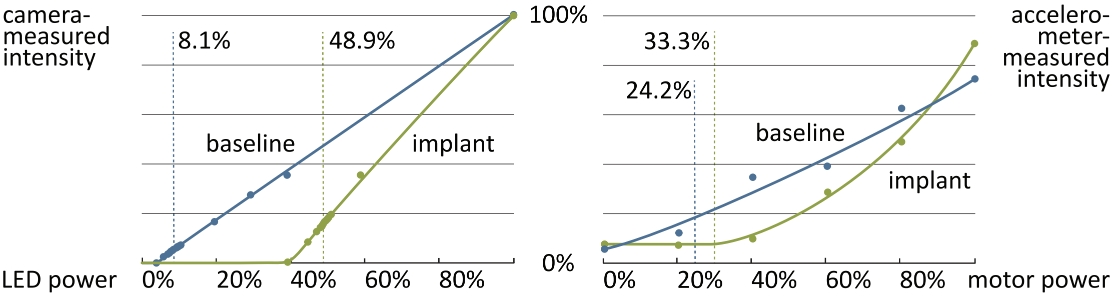


Figure 8.9: Left: Minimum perceivable LED intensity. Right: The accelerometer did not pick up a signal through skin at motor intensities of 40% and lower. Dotted lines indicate the participant's absolute perception thresholds.

Results of the Vibration Motor

The accelerometer captured a signal through the skin only when the motor was powered at 40% intensity and higher; lower intensities were indistinguishable from background noise (Figure 8.9 right). The BASELINE condition with the accelerometer resting on the motor directly shows an expected linear decay. The shown values represent the mean standard deviation of the three values read by the accelerometer. The difference in personal perception of the vibration was small (24.2% vs. 33.3%).

8.3.8 *Audio Devices (Speaker and Microphone)*

To evaluate the speaker, we again used a descending staircase design to determine minimum perceivable audio levels. We conducted the evaluation from two distances: 25 cm (CLOSE) and 60 cm (FAR). These distances simulated holding the arm to one's ear to listen to a signal (CLOSE) and hearing the signal from a resting state with the arms beside one's body (FAR). The stimulus was a 1 kHz sine wave signal [49]. During each step, an external desktop microphone measured actual output signals from 5 cm away in the CLOSE condition, and 60 cm away in the FAR condition.

To evaluate the implanted microphone, we produced audio as input from two distances (25 cm, 60 cm). Two desktop speakers pointed at the microphone and played five pre-recorded sounds at ten volume levels (100%, 80%, 60%, 40%, 20%, 10%, 8%, 4%, 2%, 1%). Four of the sound playbacks were voice ("one," "two," "three," "user"), one was a chime sound.

Apparatus Details

The implanted microphone was a regular electret condenser microphone. The external microphone was an audio-technica AT2020 USB. The speaker was a Murata Piezo 25 mm piezoelectric buzzer. The laptop recorded from both microphones at 44.1 kHz with the microphone gain set to 1.

Results of the Speaker

We first applied a bandpass filter of 100 Hz to the recorded signal around the stimuli frequency to discard background noise. The assistant could perceive the stimuli sound at a level of 5.2 dB at only 0.3% output intensity in the BASELINE session, and at 7% in the IMPLANT session (Figure 8.10). The perceivable decibel levels compare to other results [49]. Figure 8.10 illustrates the additional output intensity needed to achieve comparable sound pressures.

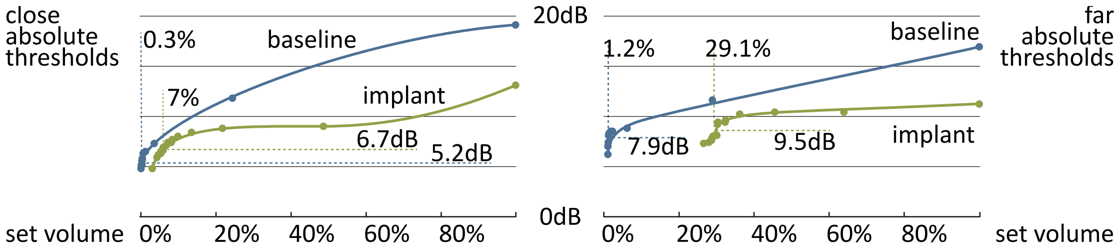


Figure 8.10: Left: Sound perception through skin is possible, but skin substantially takes away from the output intensity. Right: This effect grows with the distance between listener and speaker. Dotted lines indicate absolute perception thresholds.

Results of the Microphone

The skin accounted for a difference in recorded sound intensities of 6.5 dB (±3 dB) for the close-speaker condition and 6.24 dB (±2.5 dB) in the far-speaker condition. At full output volume, skin dampened the volume of the incoming sound by less than 2% when close by 25 cm away, but almost 10% with speakers 60 cm away as shown in Figure 8.11.

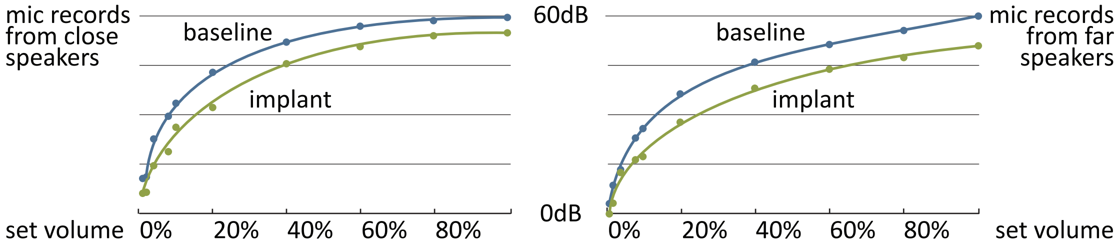


Figure 8.11: The differences in perceived sound intensities were nearly constant between the IMPLANT and the BASELINE session.

8.3.9 *Powering Device (Powermat Wireless Charger)*

To evaluate the powering device, we docked the receiver to the powering mat as shown in Figure 8.12 (left). In the BASELINE session, the two devices docked directly. In the IMPLANT session, the receiver was implanted, and the powering mat was placed on the surface of the skin directly above the implant.

Once docked, we separately measured the voltages and currents the receiver supplied with a voltmeter and an ampere-meter. We took five probes for each measurement, each time capturing values for five seconds for the meters to stabilize. We measured the provided voltages and the drawn current with four resistors: 2 kΩ, 1 kΩ, 100 Ω, 56 Ω.

Apparatus Details

The powering device was a PMR-PPC2 Universal Powercube Receiver with a PMM-1PA Powermat 1X. The voltmeter and ampere-metre was a VC830L digital multimeter.

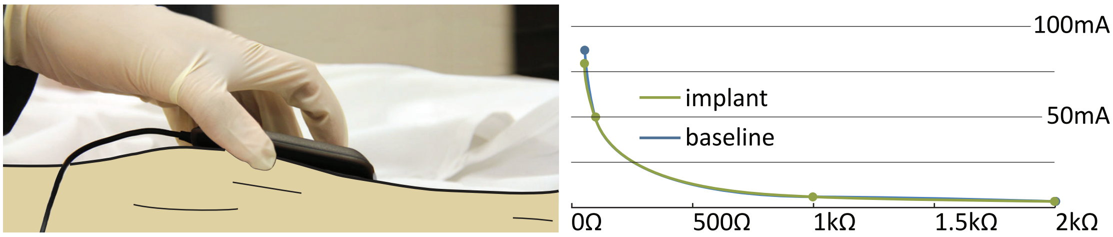


Figure 8.12: Left: The wireless charging mat docks to the receiver, which is implanted inside the specimen. Right: Skin affected the current provided through the wireless connection only at higher current values.

Results of the Powering Device

The Powermat receiver output a nominal voltage of 5.12 V in the baseline condition. Through skin, the provided voltage was unsubstantially smaller (5.11 V).

As shown in Figure 8.12 (right), skin did not impact the current drawn by the device for low resistances. For the 56 Ω resistor, the difference was 7 mA, still providing 80 mA, which should easily power an Arduino microcontroller.

8.3.10 *Wireless Communication Device (Bluetooth Chip)*

To test the performance of the wireless connection between two chips, one was external and one implanted with no space between chip and encompassing skin in the implant session. The baseline session tested both devices when placed outside. We evaluated the connection at two speed levels (slow: 9600 bps, fast: 115,200 bps), sending arrays of data in bursts between the devices (16 kB, 32 kB, 128 kB) and calculating checksums for the sent packages. The receiving device output time-stamped logs of number of received packages and its calculated checksum. The test was fully bidirectional, repeated five times and then averaged.

Apparatus Details

The Bluetooth modules were Roving Networks RN-42 (3.3 V/26 μA sleep, 3 mA connected, 30 mA transmitting) connected to an ATmega328 controller. The RN-42 featured an on-board chip antenna with no external antenna.

Results of the Wireless Communication Device

For the slow transmission speed, no packet loss occurred in either condition. The effective speed rate was 874 B/s in both conditions (Figure 8.13 left).

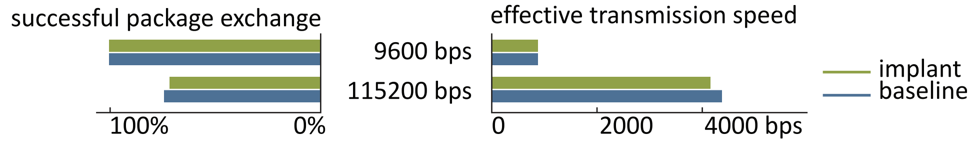


Figure 8.13: Left: Bluetooth exchanges data reliably when running slow, but comes with data loss when running fast. Right: Implanting affected fast transmission rates negatively.

For the fast transmission speed, the devices received 74% of the sent packages on average in the BASELINE condition and 71% when sent through skin in the IMPLANTED condition (Figure 8.13 right). The effective speed differed by 200 B/s (4.4 kB BASELINE vs. 4.2 kB IMPLANTED). We found no differences in direction.

8.3.11 *Discussion*

Overall, all traditional user interface components that were implanted worked under the skin: sensing input through the skin, emitting output that could be perceived through the skin, and charging and communicating wirelessly.

Regarding input, skin expectedly required user input to increase in intensity to activate sensor controls. Despite this required intensity overhead, all tested input sensors did perceive input through the skin, even at the lower levels of intensity we tested. This leaves enough dynamic range for the sensors' additional degrees of freedom, such as detecting varying pressure. As for hover detection, skin incurs an offset of brightness and diminishes capacitive signals, but both sensors responded to the approaching finger.

While output appears diminished through the skin, detection is possible at low-enough intensity levels, such that output components, too, can leverage a range of intensities for producing output.

Powering the device through the skin yielded enough voltage to have powered any of the implanted devices. It is also enough to power our *3in3out* prototype device, which we will describe in the next section. More measurements with lower resistances remain necessary to determine the maximum throughput of the tested inductive power supply beyond the 100 mA levels.

While skin affected the throughput of the fast wireless communication and accounted for a 3 pp higher loss of packages and a 0.2 kB/s drop in speed, it did not affect the slow condition. The flawless wireless communication in 9600 bps enables reliable data exchange. Results found in the related area of body-area networks differ, as transmission goes through the body or arm, not just skin [64].

8.3.12 *Exploring Exposed Components*

In addition to quantitatively evaluating input components, we wanted to prototype an exposed implanted interface component. To do so, we mounted a Blackberry trackball control on the back of an Arduino Pro Mini 8 MHz board and soldered batteries to it. The trackball was a fully autonomous standalone device. We programmed the trackball to emit a different light color when the user swiped the ball horizontally or vertically.

To expose the roller ball after it was implanted into the specimen arm, the skin and plastic cover over the roller ball were carefully incised using a scalpel. The incision was about 3 mm in length, so only the roller ball was exposed. Once implanted into the specimen, the experimenters took turns interacting with the device, which worked as expected. Figure 8.1f illustrates the exposed trackball. Note that this exploration took place *after* the quantitative evaluation had fully finished. The incision made for this exploration had no effect on our earlier evaluation.

8.4 qualitative evaluation

To explore initial user feedback on implanted user interfaces, we built and deployed a prototype device covered with a layer of artificial skin on users. Our goal was to gain initial insights on how users may feel about walking around with an interactive implanted device and to demonstrate how such devices can be prototyped and tested outside controlled lab conditions.

Study Device

We built the *3in3out* device specifically for the qualitative evaluation as shown in Figure 8.2. It features three input controls (button, tap sensor, pressure sensor) and three output components (LED, vibration motor, piezo buzzer). A Li-Po battery powers the standalone *3in3out* device.

The device implemented a game as an abstract task that involved receiving output and responding with input. At 30–90 second intervals, a randomly chosen output component triggered the user, who had to respond using the correct input: pressure sensor for the LED, tap sensor for the motor, and button for the speaker. While the LED kept blinking, the speaker and vibration motor repeated their output trigger every 10 s. Without a response, the trigger timed out after one minute. Participants received points based on the speed and accuracy of their responses.

8.4.1 *Simulating Implants: Artificial Skin*

We created artificial skin to cover our prototype and simulate actual implantation with the aid of a professional prosthetics shop, which had years of experience modeling body parts. The artificial skin was diffuse and diffused light, dampened sound and vibration in roughly the same manner to the real skin in our evaluation. Participants in the study confirmed that the artificial skin qualitatively felt like real skin. As the focus of this study was on obtaining qualitative feedback, we did not calibrate the characteristics of the artificial skin to match the absolute quantitative properties of skin we measured in our evaluation. We did not need the artificial skin to match the Bluetooth properties of skin either, because our qualitative study did not include communication devices.

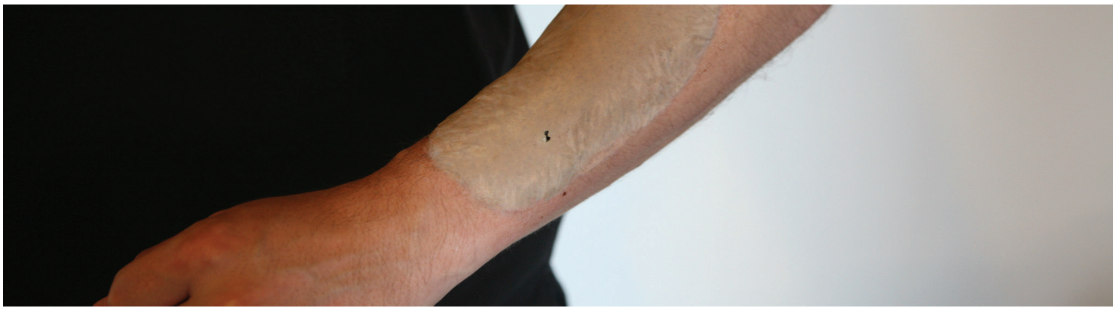


Figure 8.14: Artificial skin, created from silicone, covered the *3in3out* device to simulate implantation and allow for testing.

To create the artificial skin, we mixed Polytek Platsil-Gel 10 with Polytek Smith's Theatrical Prosthetic Deadener, which is known to produce silicone with skin-like feel and consistency. We added skin-color liquid foundation and enhanced the skin look with red, blue and beige flocking. We then poured the silicone mixture into a mold customized to fit a human arm, added the device wrapped in Seran foil, and positioned a clay arm, so that the silicone assumed the correct shape. We then affixed the artificial skin to users' arms using ADM Tronics Pros-Aide medical grade adhesive. The final layer of artificial skin measured 4.5″×2″ (Figure 8.14) and was 1–2 mm thick above the device (i. e., similar to anterior surface skin [46], which we studied).

8.4.2 *Task and Procedure*

We designed a set of six primary tasks to distract from wearing the prototype device, which interrupted participants while carrying out those tasks: 1) ask a person for the time, 2) board public transport and exit after two stops, 3) ask a person for directions to the post office, 4) pick-up a free newspaper, 5) buy a coffee, and, finally, 6) sit in a park, finish the coffee and read the newspaper. Participants' secondary task was to respond to the triggers that the *3in3out* device emitted, and try to achieve a high score. The device recorded response times, errors and point totals. The study took place in downtown Toronto, Canada on a summer day, which represented a realistic worst-case scenario; both, direct sunlight and noise levels were very intense.

Participants first received a demonstration of the device and practiced its use. Participants then left the building to perform all primary tasks, and returned after approximately 60 minutes. Participants filled out a questionnaire after the study, sharing their impression when using the device in public environments and any the reactions they received.

8.4.3 *Participants*

We recruited 4 participants (1 female) from our institution. Participants were between 28 and 36 years old and wore the prototype device on their left arm. We reimbursed participants for using public transport and buying coffee.

8.4.4 *Results*

Overall, participants found the device easy to use. All liked the tap sensor ("easy to use") and button ("easy to find," "haptic feedback"), but none enjoyed the pressure sensor. For output components, all ranked the LED lowest for perception relative to the other output components, the speaker medium, and the vibration motor best ("really easy to notice"). While these results suggest that the device might work better in environments quieter and/or darker than the noisy city setting in direct sunlight, participants were able to see the LED blinking when looking at it.

While participants mentioned receiving curious looks from others when interacting with their arm, no external person approached a participant, even though they spent time in casual settings (e. g., coffee place and public transport).

Most importantly, the results of our study demonstrate that implanted user interfaces can be used to support interactive tasks. This evaluation also provides a methodology to pave the way for future evaluations and mockups of more elaborate devices and applications of implanted user interfaces.

8.5 MEDICAL CONSIDERATIONS OF INTERACTIVE IMPLANTED DEVICES

While the primary goal of our work is to consider implanted user interfaces from an HCI perspective, it is also important to discuss some of the medical considerations. Below we discuss some of the issues surrounding the feasibility of implanted user interfaces.

8.5.1 *Location*

In our study, the devices were implanted under the skin on the front of the forearm, just distal to the elbow joint. This location was chosen as the devices could be easily activated by an individual with their other hand and would not be in an area where damage by impact is likely. For the most part, these devices could be implanted deep into the skin in the subcutaneous tissue anywhere in the body where the devices are accessible and can transmit signals. This includes the upper and lower limbs, the chest wall, abdomen, etc. Areas covered by thick skin, such as the palms and soles of the feet, would not be suitable for implantables, as the skin is too thick and tough to interact. The thickness of human skin ranges between 0.5 mm on the eyelids to 4+ mm on the palms and soles of the feet [46].

The superficial placement of the devices, directly under the skin, facilitates device activation and signal transmission. The devices can be inserted between the skin and subcutaneous tissue, providing a minimally invasive approach. The deep underlying tissues, e. g., muscles, would not be disrupted. Similarly, pacemakers are placed under the skin in the chest or abdominal regions and the wires that are extending from the heart are connected to the pacemaker. Only a small skin incision that is later closed with sutures is needed to insert the pacemaker. The device remains stationary in its implanted location due to the fibrous nature of subcutaneous tissue.

The tracking ball was the only device we implanted that required surface exposure. The device worked very well under the experimental conditions, but much work needs to be done to assess the medical implications of a long-term insertion of an exposed device.

8.5.2 *Device parameters*

Tissue fluid will penetrate a device that is not encased in a protective hull, and affect its function. The hull's material must be carefully chosen to be pharmacologically inert and nontoxic to body tissues. For examples, pacemakers are typically made from titanium or titanium alloys, and the leads from polyether polyurethanes. *In vivo* testing would need to be carried out to determine what materials are most suitable.

The device should be as small as possible, so it is easily implantable and cosmetically acceptable to the recipient. Functionality and minimal disruption of the contour of the skin are important considerations.

8.5.3 *Risks*

The main medical risk of implanting devices is infection. Infection can be caused by the procedure of implanting the devices. There are also possible risks to muscles if the device is implanted any deeper than the subcutaneous tissue. The material used for the casing could also possibly cause infections, so it will be important that the material being used passes through proper testing. It is very difficult to hypothesize about other types of risks without performing testing. The wear of skin depends on the pressure applied to it; while paraplegics get sore skin from body weight resting on single spots through bones, skin is unlikely to wear from manual pressure. The proposed input with implanted devices is short and low in force and intensity, making skin unlikely to wear. One risk that is relatively low is that of the skin actually tearing. Skin is very strong and it is unlikely the small devices would cause any damage. However, determining the long-term effects of interactions with implanted devices on skin requires further studies.

8.5.4 *Implications and Future Studies*

All of the input and output devices were functional under the experimental conditions of this study. Further cadaveric study is needed to determine if gender, skin color, and site of implantation affect device function. In the next phase, testing would also be carried out on unembalmed tissue, although the skin of lightly embalmed and unembalmed specimens is similar, loose and pliable in both cases. Finally, the medical implications of long-term insertion of devices of this nature require detailed study.

8.6 DISCUSSION AND LIMITATIONS

The results of our study show that traditional interfaces for input, output, wireless communication, and charging still function when embedded in the subcutaneous tissue of the forearm. Having obtained an evaluation of common components establishes the foundation for future investigations into more complex devices to explore the many other aspects of implanted user interfaces.

For example, we disregarded security concerns in our exploration. Wireless implanted devices need to prevent malicious activities and interactions from users other than the host user, such as stealing or altering stored information and manipulating the devices' operating system [33, 57].

The processing capabilities of the devices that were implanted during the technical evaluation, as well the *3in3out* device, require only simple processing on the microchip. More work is necessary to investigate if and how implanted devices can perform more computationally intensive operations (e. g., classification tasks using machine learning [136]) and how this affects the power-supply needs.

Social perception of implanted interfaces, both by host users as well as the public, requires more studying. Although this has been studied with implanted *medical* devices [32], social perception of invisible and implanted *user interfaces* and devices remain to be examined.

We conducted our qualitative evaluation with participants in the summer, which is why all participants wore short-sleeve shirts. In the winter, cloth will additionally cover implanted input and output components [124] and interfere with interaction, which raises new challenges.

Study Limitations

Our technical evaluation comprised a single specimen. In addition, we carried out the staircase evaluations with a single participant. As such, the metrics we have collected can serve as baselines for future experimentations, but should be generalized with caution. Furthermore, our evaluation captured technical metrics from the devices, and not human factor results. In the future, it may be interesting to have external participants interact with the implanted devices and study task performance levels.

8.7 OUTLOOK

The technological transition society has made in the past 30 years is astounding. Technology, and the way we use it, continues to evolve and no one can tell what the future holds. Even though today, the primary motivation for implanted devices will almost always stem from the medical domain [37], the future of interactive implanted devices is likely connected to the future of (ultra-)mobile devices. Implanted devices already fulfill all the properties mobile devices are striving towards, such as providing always-available interaction and restoring or augmenting users with new functionality.

Our work takes a first step towards understanding how interacting with ultra-small devices might be accomplished, and begins to ask and answer some of the important technical, human factors, and medical questions. Our results have the potential to provide medical implants with new capabilities, and more broadly change the relationship between humans and interactive miniature devices.

Future interactive devices—ultra-wearable devices, i. e., devices worn at all times, as well as implanted devices—will raise a range of questions. On the highest level, we will have to examine which "natural" interaction modalities will prove practical to use devices' functionality in the context of certain social settings. Further pushing the minimum size of future devices will spark additional questions: What form factors will be viable? How are the form factors going to evolve in the next ten years? Where on or in the body will users wear such devices?

Considering the specific future of implanted devices, we will have to examine their purpose on a general level: What applications are worth being promoted to the level of an implanted device? Will implanted devices be special-purpose, similar to implanted medical devices? Will they merely reside in the body or interconnect with the user's senses? Will we use implanted interfaces to augment human abilities, such as by adding "senses" of direction or time?

To answer these questions, many technical challenges need to be tackled, such as research on materials for such devices and security aspects of communicating with them. Of course, questions about human values are paramount, such as ethical concerns and the medical risks that currently accompany the use of implanted devices.

In conclusion, we envision a future in which technology itself fades into the background and augments users directly. We thereby expect technology to further blend into our lives in the form of ultra-wearable or even future implanted devices.

BIBLIOGRAPHY

[1] Abate, A. F., Nappi, M., Riccio, D., and Sabatino, G. 2D and 3D face recognition: A survey. *Pattern Recognition Letters* 28.14 (2007), 1885–1906 (cited on page 22).

[2] Agarwal, A., Izadi, S., Chandraker, M., and Blake, A. High Precision Multi-touch Sensing on Surfaces using Overhead Cameras. In *Proceedings of TABLETOP*. 2007, 197–200 (cited on pages 16, 70).

[3] Akizuki, S. Touch Pad Having Fingerprint Detecting Function and Information Processing Apparatus Employing the Same. Patent 6,360,004 (US). 2002 (cited on page 24).

[4] Albinsson, P.-A. and Zhai, S. High precision touch screen interaction. In *Proceedings of CHI*. 2003, 105–112 (cited on pages 19, 20).

[5] Anderson, S. D. Practical light embalming technique for use in the surgical fresh tissue dissection laboratory. *Clinical Anatomy* 19.1 (2006), 8–11 (cited on page 148).

[6] Annett, M., Grossman, T., Wigdor, D., and Fitzmaurice, G. Medusa: a proximity-aware multi-touch tabletop. In *Proceedings of UIST*. 2011, 337–346 (cited on pages 22, 23).

[7] Inc., A. *iOS Human Interface Guidelines*. URL: `http://developer.apple.com/library/ios/\#documentation/userexperience/conceptual/mobilehig/Characteristics/Characteristics.html` (cited on page 19).

[8] Ashbaugh, D. R. *Quantitative-Qualitative Friction Ridge Analysis: An Introduction to Basic and Advances Ridgeology*. CRC Press, 1999 (cited on page 23).

[9] Ashbrook, D. L., Clawson, J. R., Lyons, K., Starner, T. E., and Patel, N. Quickdraw: the impact of mobility and on-body placement on device access time. In *Proceedings of CHI*. 2008, 219–222 (cited on page 143).

[10] Augsten, T., Kaefer, K., Meusel, R., Fetzer, C., Kanitz, D., Stoff, T., Becker, T., Holz, C., and Baudisch, P. Multitoe: high-precision interaction with back-projected floors based on high-resolution multi-touch input. In *Proceedings of UIST*. 2010, 209–218 (cited on page 99).

[11] Badshah, A., Gupta, S., Cohn, G., Villar, N., Hodges, S., and Patel, S. N. Interactive generator: a self-powered haptic feedback device. In *Proceedings of CHI*. 2011, 2051–2054 (cited on page 145).

[12] Bailly, G., Müller, J., Rohs, M., Wigdor, D., and Kratz, S. ShoeSense: a new perspective on gestural interaction and wearable applications. In *Proceedings of CHI*. 2012, 1239–1248 (cited on page 3).

[13] Barrett, G. and Omote, R. Projected-Capacitive Touch Technology. *Information Display* 26.3 (2010), 26–21 (cited on page 14).

[14] Bartindale, T. and Harrison, C. Stacks on the surface: resolving physical order using fiducial markers with structured transparency. In *Proceedings of ITS*. 2009, 57–60 (cited on page 28).

[15] Baudisch, P. and Chu, G. Back-of-device interaction allows creating very small touch devices. In *Proceedings of CHI*. 2009, 1923–1932 (cited on pages 18, 19).

[16] Baudisch, P., Becker, T., and Rudeck, F. Lumino: tangible blocks for tabletop computers based on glass fiber bundles. In *Proceedings of CHI*. 2010, 1165–1174 (cited on pages 14, 26, 28).

[17] Bay, H., Tuytelaars, T., and Gool, L. SURF: Speeded Up Robust Features. In *Proceedings of ECCV*. 2006, 404–417 (cited on page 41).

[18] Beaudouin-Lafon, M. and Mackay, W. E. Reification, polymorphism and reuse: three principles for designing visual interfaces. In *Proceedings of AVI*. 2000, 102–109 (cited on page 32).

[19] Benko, H., Wilson, A. D., and Baudisch, P. Precise selection techniques for multi-touch screens. In *Proceedings of CHI*. 2006, 1263–1272 (cited on pages 17, 19, 20).

[20] Bjorn, V. and Belongie, S. Configurable multi-function touchpad device. Patent 6,950,539 (US). 2005 (cited on page 24).

[21] Borchers, J., Ringel, M., Tyler, J., and Fox, A. Stanford interactive workspaces: a framework for physical and graphical user interface prototyping. *Wireless Commun.* 9.6 (Dec. 2002), 64–69 (cited on page 28).

[22] Bränzel, A., Holz, C., Hoffmann, D., Schmidt, D., Knaust, M., Lühne, P., Meusel, R., Richter, S., and Baudisch, P. GravitySpace: tracking users and their poses in a smart room using a pressure-sensing floor. In *Proceedings of CHI*. 2013, 725–734 (cited on page 99).

[23] Brown, C., Kato, H., Maeda, K., and Hadwen, B. A Continuous-Grain Silicon-System LCD With Optical Input Function. *Solid-State Circuits* 42.12 (2007), 2904–2912 (cited on pages 24, 137).

[24] Brumberg, J. S., Nieto-Castanon, A., Kennedy, P. R., and Guenther, F. H. Brain-computer interfaces for speech communication. *Speech Commun.* 52.4 (Apr. 2010), 367–379 (cited on page 144).

[25] Brumitt, B., Meyers, B., Krumm, J., Kern, A., and Shafer, S. A. EasyLiving: Technologies for Intelligent Environments. In *Proceedings of HUC*. 2000, 12–29 (cited on pages 27, 28, 117).

[26] Cao, X., Wilson, A., Balakrishnan, R., Hinckley, K., and Hudson, S. ShapeTouch: Leveraging contact shape on interactive surfaces. In *Proceedings of TABLETOP*. 2008, 129–136 (cited on pages 17, 106).

[27] Choi, I. and Ricci, C. Foot-mounted gesture detection and its application in virtual environments. In *Proceedings of SMC*. 1997, 4248–4253 (cited on pages 27, 117).

[28] Cornsweet, T. N. The staircase-method in psychophysics. *The American journal of psychology* 75.3 (1962), 485–491 (cited on page 151).

[29] Cruz-Neira, C., Sandin, D. J., and DeFanti, T. A. Surround-screen projection-based virtual reality: the design and implementation of the CAVE. In *Proceedings of SIGGRAPH*. 1993, 135–142 (cited on pages 28, 100, 117).

[30] Davidson, P. L. and Han, J. Y. Extending 2D object arrangement with pressure-sensitive layering cues. In *Proceedings of UIST*. 2008, 87–90 (cited on pages 27, 102).

[31] Davies, E. R. *Machine Vision: Theory, Algorithms, Practicalities*. Morgan Kaufmann Publishers Inc., 2004 (cited on page 132).

[32] Denning, T., Borning, A., Friedman, B., Gill, B. T., Kohno, T., and Maisel, W. H. Patients, pacemakers, and implantable defibrillators: human values and security for wireless implantable medical devices. In *Proceedings of CHI*. 2010, 917–926 (cited on page 162).

[33] Denning, T., Matsuoka, Y., and Kohno, T. Neurosecurity: security and privacy for neural devices. *Neurosurgical Focus* 27.1 (2009), E7 (cited on page 162).

[34] Dietz, P. and Leigh, D. DiamondTouch: a multi-user touch technology. In *Proceedings of UIST*. 2001, 219–226 (cited on pages 14, 21, 98).

[35] Dowling Jr., R. and Knowlton, K. Fingerprint Acquisition System With a Fiber Optic Block. Patent 4,785,171 (US). 1988 (cited on page 25).

[36] Downs, R. Using resistive touch screens for human/machine interface. *Analog Applications Journal* 5 (2005), 5–10 (cited on page 13).

[37] Drew, T. and Gini, M. Implantable medical devices as agents and part of multiagent systems. In *Proceedings of AAMAS*. 2006, 1534–1541 (cited on page 163).

[38] Acrylite. *LED (Endlighten)*. URL: `http://www.acrylite-shop.com/US/us/0e011-9fpcttqkh6o~p.html` (cited on page 84).

[39] Engelbart, D. C. X-Y Position Indicator for a Display System. Patent 3,541,541 (US). 1970 (cited on page 1).

[40] Ferrari, A. and Tartagni, M. Touchpad Providing Screen Cursor Movement Control. Patent 6,392,636 (US). 2002 (cited on page 24).

[41] Feynman, R. *Lecture 7: Seeking New Laws*. The Messenger Series, BBC Science&Nature (cited on page 53).

[42] FingerWorks. *iGesture Pad (discontinued)*. URL: `http://web.archive.org/web/20090605012754/http://www.fingerworks.com/` (cited on pages 20, 34).

[43] Fitts, P. M. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology* 47.6 (June 1954), 381–391 (cited on page 17).

[44] Fitzpatrick, R. C. and Day, B. L. Probing the human vestibular system with galvanic stimulation. *Journal of Applied Physiology* 96.6 (2004), 2301–2316 (cited on page 144).

[45] Forlines, C., Wigdor, D., Shen, C., and Balakrishnan, R. Direct-touch vs. mouse input for tabletop displays. In *Proceedings of CHI*. 2007, 647–656 (cited on pages 18, 20, 31, 37).

[46] Fornage, B. D. and Deshayes, J.-L. Ultrasound of normal skin. *Journal of clinical ultrasound* 14.8 (1986), 619–622 (cited on pages 158, 160).

[47] Fujieda, O. and Sugama, S. Fingerprint Image Input Device Having an Image Sensor with Openings. Patent 5,446,290 (US). 1995 (cited on page 25).

[48] Fujieda, I. and Haga, H. Fingerprint input based on scattered-light detection. *Applied Optics* 36.35 (1997), 9152–9156 (cited on pages 25, 79).

[49] Gelfand, S. A. *Hearing: An introduction to psychological and physiological acoustics*. CRC Press, 2004 (cited on page 153).

[50] Gonzalez, R. C. and Woods, R. E. *Digital Image Processing*. 2nd. Addison-Wesley Longman Publishing Co., Inc., 1992 (cited on page 132).

[51] Greenspon, A. J., Patel, J. D., Lau, E., Ochoa, J. A., Frisch, D. R., Ho, R. T., Pavri, B. B., and Kurtz, S. M. Trends in Permanent Pacemaker Implantation in the United States From 1993 to 2009: Increasing Complexity of Patients and Procedures. *Journal of the American College of Cardiology* 60.16 (2012), 1540–1545 (cited on page 141).

[52] Grønbaek, K., Iversen, O. S., Kortbek, K. J., Nielsen, K. R., and Aagaard, L. IGameFloor: a platform for co-located collaborative games. In *Proceedings of ACE*. 2007, 64–71 (cited on pages 27, 28).

[53] Grossman, T. and Balakrishnan, R. A probabilistic approach to modeling two-dimensional pointing. *Comput.-Hum. Interact.* 12.3 (Sept. 2005), 435–459 (cited on page 17).

[54] Gust, L. Compact Optical Pointing Apparatus and Method. Patent 7,102,617 (US). 2006 (cited on page 24).

[55] Gustafson, S., Holz, C., and Baudisch, P. Imaginary phone: learning imaginary interfaces by transferring spatial memory from a familiar device. In *Proceedings of UIST*. 2011, 283–292 (cited on page 49).

[56] Hall, A., Cunningham, J., Roache, R., and Cox, J. Factors affecting performance using touch-entry systems: Tactual recognition fields and system accuracy. *Journal of Applied Psychology* 73.4 (Nov. 1988), 711–720 (cited on page 18).

[57] Halperin, D., Heydt-Benjamin, T. S., Ransford, B., Clark, S. S., Defend, B., Morgan, W., Fu, K., Kohno, T., and Maisel, W. H. Pacemakers and implantable cardiac defibrillators: Software radio attacks and zero-power defenses. In *Proceedings of SP*. 2008, 129–142 (cited on page 162).

[58] Han, J. Y. Low-cost multi-touch sensing through frustrated total internal reflection. In *Proceedings of UIST*. 2005, 115–118 (cited on pages 2, 15, 17, 21, 27, 42, 76, 78, 79, 103).

[59] Hand, C. A Survey of 3D Interaction Techniques. *Comput. Graph. Forum* 16.5 (1997), 269–281 (cited on page 3).

[60] Harrison, C., Tan, D., and Morris, D. Skinput: appropriating the body as an input surface. In *Proceedings of CHI*. 2010, 453–462 (cited on page 144).

[61] Harrison, C., Sato, M., and Poupyrev, I. Capacitive fingerprinting: exploring user differentiation by sensing electrical properties of the human body. In *Proceedings of UIST*. 2012, 537–544 (cited on page 22).

[62] Hinckley, K., Pierce, J., Sinclair, M., and Horvitz, E. Sensing techniques for mobile interaction. In *Proceedings of UIST*. 2000, 91–100 (cited on page 144).

[63] Hinckley, K., Pausch, R., Goble, J. C., and Kassell, N. F. A survey of design issues in spatial input. In *Proceedings of UIST*. 1994, 213–222 (cited on page 3).

[64] Ho, H., Saeedi, E., Kim, S., Shen, T. T., and Parviz, B. Contact lens with integrated inorganic semiconductor devices. In *Proceedings of MEMS*. 2008, 403–406 (cited on page 157).

[65] Hodges, S., Izadi, S., Butler, A., Rrustemi, A., and Buxton, B. ThinSight: versatile multi-touch sensing for thin form-factor displays. In *Proceedings of UIST*. 2007, 259–268 (cited on pages 16, 137).

[66] Holz, C. and Baudisch, P. The generalized perceived input point model and how to double touch accuracy by extracting fingerprints. In *Proceedings of CHI*. 2010, 581–590 (cited on page 29).

[67] Holz, C. and Wilson, A. Data miming: inferring spatial object descriptions from human gesture. In *Proceedings of CHI*. 2011, 811–820 (cited on page 3).

[68] Holz, C. and Baudisch, P. Understanding touch. In *Proceedings of CHI*. 2011, 2501–2510 (cited on page 49).

[69] Holz, C., Grossman, T., Fitzmaurice, G., and Agur, A. Implanted user interfaces. In *Proceedings of CHI*. 2012, 503–512 (cited on page 141).

[70] Holz, C. and Baudisch, P. Fiberio: a touchscreen that captures fingerprints. In *Proceedings of UIST*. 2013 (cited on page 75).

[71] Hossu, D. Device for Measuring Elevations and/or Depressions in a Surface. Patent 8,149,408 (US). 2012 (cited on page 25).

[72] Huo, X., Wang, J., and Ghovanloo, M. A Magneto-Inductive Sensor Based Wireless Tongue-Computer Interface. *Neural Systems and Rehabilitation Engineering* 16.5 (2008), 497–504 (cited on page 144).

[73] Izadi, S., Hodges, S., Taylor, S., Rosenfeld, D., Villar, N., Butler, A., and Westhues, J. Going beyond the display: a surface technology with an electronically switchable diffuser. In *Proceedings of UIST*. 2008, 269–278 (cited on pages 16, 78).

[74] Jackson, D., Bartindale, T., and Olivier, P. FiberBoard: compact multi-touch display using channeled light. In *Proceedings of ITS*. 2009, 25–28 (cited on page 26).

[75] Jobs, S. et al. Touch screen device, method, and graphical user interface for determining commands by applying heuristics. Patent 7,479,949 (US). 2009 (cited on page 20).

[76] Johnson, E. Touch display—a novel input/output device for computers. *Electronics Letters* 1 (8 Oct. 1965), 219–220 (cited on page 1).

[77] Jovanov, E., Milenkovic, A., Otto, C., and De Groen, P. C. A wireless body area network of intelligent motion sensors for computer assisted physical rehabilitation. *Journal of NeuroEngineering and rehabilitation* 2.1 (2005), 6 (cited on page 145).

[78] Kaiser, M. K. and Proffitt, D. R. Observers' sensitivity to dynamic anomalies in collisions. *Perception & Psychophysics* 42.3 (1987), 275–280 (cited on page 151).

[79] Kaltenbrunner, M. and Bencina, R. reacTIVision: a computer-vision framework for table-based tangible interaction. In *Proceedings of TEI*. 2007, 69–74 (cited on page 97).

[80] Karrer, T., Wittenhagen, M., Lichtschlag, L., Heller, F., and Borchers, J. Pinstripe: eyes-free continuous input on interactive clothing. In *Proceedings of CHI*. 2011, 1313–1322 (cited on page 143).

[81] Kenyon, I. R. *The Light Fantastic: A Modern Introduction to Classical and Quantum Optics*. Oxford University Press, Inc., 2008 (cited on pages 25, 81).

[82] Kientz, J. A., Patel, S. N., Jones, B., Price, E., Mynatt, E. D., and Abowd, G. D. The Georgia Tech aware home. In *Extracted Abstracts of CHI*. 2008, 3675–3680 (cited on pages 28, 132).

[83] Kim, D., Dunphy, P., Briggs, P., Hook, J., Nicholson, J. W., Nicholson, J., and Olivier, P. Multi-touch authentication on tabletops. In *Proceedings of CHI*. 2010, 1093–1102 (cited on page 22).

[84] *Kinect camera*. URL: `http://www.microsoft.com/Kinect` (cited on page 3).

[85] Krogh, P., Ludvigsen, M., and Lykke-Olesen, A. Help Me Pull That Cursor: A Collaborative Interactive Floor Enhancing Community Interaction. *Australasian Journal of Information Systems* 11.2 (2007) (cited on pages 27, 28, 100).

[86] Lambert, M. E., Chadwick, G. A., McMahon, A., and Scarffe, J. H. Experience with the portacath. *Hematological Oncology* 6.1 (1988), 57–63 (cited on page 145).

[87] LaViola Jr., J. J., Feliz, D. A., Keefe, D. F., and Zeleznik, R. C. Hands-free multi-scale navigation in virtual environments. In *Proceedings of I3D*. 2001, 9–15 (cited on page 28).

[88] Leitner, J. and Haller, M. Geckos: combining magnets and pressure images to enable new tangible-object design and interaction. In *Proceedings of CHI*. 2011, 2985–2994 (cited on page 28).

[89] Levitt, H. Transformed up-down methods in psychoacoustics. *Journal of the Acoustical society of America* 49 (1971), 467 (cited on page 151).

[90] Li, Z. and Wang, Z. L. Air/Liquid-Pressure and Heartbeat-Driven Flexible Fiber Nanogenerators as a Micro/Nano-Power Source or Diagnostic Sensor. *Advanced Materials* 23.1 (2011), 84–89 (cited on page 145).

[91] Lowe, D. G. Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vision* 60.2 (Nov. 2004), 91–110 (cited on page 124).

[92] Malik, S. and Laszlo, J. Visual touchpad: a two-handed gestural input device. In *Proceedings of ICMI*. 2004, 289–296 (cited on pages 16, 21).

[93] Maltoni, D., Maio, D., Jain, A. K., and Prabhakar, S. *Handbook of Fingerprint Recognition*. 2nd. Springer Publishing Company, Incorporated, 2009 (cited on pages 10, 23, 24, 25, 86, 87, 88, 89, 96, 97).

[94] Marquardt, N., Kiemer, J., and Greenberg, S. What caused that touch?: expressive interaction with a surface through fiduciary-tagged gloves. In *Proceedings of ITS*. 2010, 139–142 (cited on page 23).

[95] Masters, A. and Michael, K. Humancentric Applications of RFID Implants: The Usability Contexts of Control, Convenience and Care. In *Proceedings of WMCS*. 2005, 32–41 (cited on page 145).

[96] Matsushita, N. and Rekimoto, J. HoloWall: designing a finger, hand, body, and object sensitive wall. In *Proceedings of UIST*. 1997, 209–210 (cited on pages 14, 21, 77, 87, 90, 104).

[97] Meghdadi, M. and Jalilzadeh, S. Validity and acceptability of results in fingerprint scanners. In *Proceedings of MMACTE*. 2005, 259–266 (cited on page 25).

[98] Memon, S., Sepasian, M., and Balachandran, W. Review of finger print sensing technologies. In *Proceedings of INMIC*. 2008, 226–231 (cited on page 25).

[99] Microsoft. *Surface*. URL: `http://www.microsoft.com/en-us/news/press/2007/may07/05-29MSSurfacePR.aspx` (cited on pages 14, 20, 100, 106).

[100] Microsoft. *PixelSense*. URL: `http://www.microsoft.com/en-us/pixelsense/` (cited on pages 24, 137).

[101] Molyneaux, D., Izadi, S., Kim, D., Hilliges, O., Hodges, S., Cao, X., Butler, A., and Gellersen, H. Interactive environment-aware handheld projectors for pervasive computing spaces. In *Proceedings of Pervasive*. 2012, 197–215 (cited on pages 28, 132).

[102] Moore, M. M. and Kennedy, P. R. Human factors issues in the neural signals direct brain-computer interfaces. In *Proceedings of Assets*. 2000, 114–120 (cited on page 145).

[103] Moscovich, T. Contact area interaction with sliding widgets. In *Proceedings of UIST*. 2009, 13–22 (cited on pages 17, 106).

[104] Mota, S. and Picard, R. Automated Posture Analysis for Detecting Learner's Interest Level. In *Proceedings of CVPRW*. Volume 5. 2003, 49 (cited on page 28).

[105] Mutlu, B., Krause, A., Forlizzi, J., Guestrin, C., and Hodgins, J. Robust, low-cost, non-intrusive sensing and recognition of seated postures. In *Proceedings of UIST*. 2007, 149–158 (cited on pages 28, 121).

[106] Nakajima, K., Itoh, Y., Yoshida, A., Takashima, K., Kitamura, Y., and Kishino, F. FuSA2 touch display. In *Proceedings of SIGGRAPH Emerging Technologies*. 2010, 11:1–11:1 (cited on page 26).

[107] Ni, T. and Baudisch, P. Disappearing mobile devices. In *Proceedings of UIST*. 2009, 101–110 (cited on pages 143, 144).

[108] Norman, D. *The Design of Everyday Things*. Basic Books, 1988 (cited on page 51).

[109] L., O. An Overview of Fingerprint Verification Technologies. *Information Security Technical Report* 3.1 (1998), 21–32 (cited on page 25).

[110] Olwal, A. and Wilson, A. D. SurfaceFusion: unobtrusive tracking of everyday objects in tangible user interfaces. In *Proceedings of GI*. 2008, 235–242 (cited on page 23).

[111] Orr, R. J. and Abowd, G. D. The smart floor: a mechanism for natural user identification and tracking. In *Extracted Abstracts of CHI*. 2000, 275–276 (cited on pages 28, 131).

[112] Orth, M., Post, R., and Cooper, E. Fabric computing interfaces. In *Proceedings of CHI*. 1998, 331–332 (cited on page 143).

[113] Paradiso, J. A. and Hu, E. Expressive footwear for computer-augmented dance performance. In *Proceedings of ISWC*. 1997, 165–166 (cited on page 27).

[114] Paradiso, J., Abler, C., Hsiao, K.-y., and Reynolds, M. The magic carpet: physical sensing for immersive environments. In *Extracted Abstracts of CHI*. 1997, 277–278 (cited on pages 26, 27).

[115] Paradiso, J., Hu, E., and Hsiao, K.-y. Instrumented footwear for interactive dance. In *Proceedings of IDAT*. 1999, 57–60 (cited on page 27).

[116] Pavlou, M. and Allinson, N. M. Automatic extraction and classification of footwear patterns. In *Proceedings of IDEAL*. 2006, 721–728 (cited on page 27).

[117] Pavlou, M. and Allinson, N. M. Footwear Recognition. In *Encyclopedia of Biometrics*. 2009, 557–562 (cited on page 27).

[118] Pier, K. and Landay, J. *Issues for Location-Independent Interfaces*. Technical Report ISTL92-4. Xerox PARC, Dec. 1992 (cited on page 106).

[119] Piper, A. M., O'Brien, E., Morris, M. R., and Winograd, T. SIDES: a cooperative tabletop computer game for social skills development. In *Proceedings of CSCW*. 2006, 1–10 (cited on page 24).

[120] Potter, R. L., Weldon, L. J., and Shneiderman, B. Improving the accuracy of touch screens: an experimental evaluation of three strategies. In *Proceedings of CHI*. 1988, 27–32 (cited on pages 19, 20).

[121] Powermat. *Inductive Charger*. URL: `http://www.powermat.com` (cited on page 145).

[122] Qian, G., Zhang, J., and Kidané, A. People identification using gait via floor pressure sensing and analysis. In *Proceedings of EuroSSC*. 2008, 83–98 (cited on page 131).

[123] Ramakers, R., Vanacken, D., Luyten, K., Coninx, K., and Schöning, J. Carpus: a non-intrusive user identification technique for interactive surfaces. In *Proceedings of UIST*. 2012, 35–44 (cited on page 22).

[124] Rekimoto, J. GestureWrist and GesturePad: unobtrusive wearable interaction devices. In *Proceedings of ISWC*. 2001, 21–27 (cited on pages 143, 162).

[125] Rekimoto, J. SmartSkin: an infrastructure for freehand manipulation on interactive surfaces. In *Proceedings of CHI*. 2002, 113–120 (cited on pages 14, 17, 21).

[126] Rekimoto, J. Pick-and-drop: a direct manipulation technique for multiple computer environments. In *Proceedings of UIST*. 1997, 31–39 (cited on page 24).

[127] Ren, X. and Moriya, S. Improving selection performance on pen-based systems: a study of pen-based interaction for selection tasks. *Comput.-Hum. Interact.* 7.3 (Sept. 2000), 384–416 (cited on page 19).

[128] Richardson, B., Leydon, K., Fernstrom, M., and Paradiso, J. A. Z-Tiles: building blocks for modular, pressure-sensing floorspaces. In *Extracted Abstracts of CHI*. 2004, 1529–1532 (cited on page 27).

[129] Richter, S., Holz, C., and Baudisch, P. Bootstrapper: recognizing tabletop users by their shoes. In *Proceedings of CHI*. 2012, 1249–1252 (cited on pages 22, 24, 75).

[130] Rogers, S., Williamson, J., Stewart, C., and Murray-Smith, R. AnglePose: robust, precise capacitive touch tracking via 3d orientation estimation. In *Proceedings of CHI*. 2011, 2575–2584 (cited on page 20).

[131] Rosco. *Projection Surfaces for Screens*. URL: `http://www.rosco.com` (cited on page 103).

[132] Rosenberg, I. and Perlin, K. The UnMousePad: an interpolating multi-touch force-sensing input pad. In *Proceedings of SIGGRAPH*. 2009, 65:1–65:9 (cited on pages 27, 120).

[133] Roth, V., Schmidt, P., and Güldenring, B. The IR ring: authenticating users' touches on a multi-touch display. In *Proceedings of UIST*. 2010, 259–262 (cited on page 23).

[134] Roudaut, A., Huot, S., and Lecolinet, E. TapTap and MagStick: improving one-handed target acquisition on small touch-screens. In *Proceedings of AVI*. 2008, 146–153 (cited on page 19).

[135] Roudaut, A., Lecolinet, E., and Guiard, Y. MicroRolls: expanding touch-screen input vocabulary by distinguishing rolls vs. slides of the thumb. In *Proceedings of CHI*. 2009, 927–936 (cited on page 21).

[136] Saponas, T. S., Tan, D. S., Morris, D., Balakrishnan, R., Turner, J., and Landay, J. A. Enabling always-available input with muscle-computer interfaces. In *Proceedings of UIST*. 2009, 167–176 (cited on pages 142, 144, 162).

[137] Saponas, T. S., Kelly, D., Parviz, B. A., and Tan, D. S. Optically sensing tongue gestures for computer input. In *Proceedings of UIST*. 2009, 177–180 (cited on page 144).

[138] Schmidt, D., Chong, M. K., and Gellersen, H. HandsDown: hand-contour-based user identification for interactive surfaces. In *Proceedings of NordiCHI*. 2010, 432–441 (cited on pages 22, 23, 98).

[139] Schmidt, D., Chehimi, F., Rukzio, E., and Gellersen, H. PhoneTouch: a technique for direct phone interaction on surfaces. In *Proceedings of UIST*. 2010, 13–16 (cited on page 23).

[140] Schöning, J., Hook, J., Motamedi, N., Olivier, P., Echtler, F., Brandl, P., Muller, L., Daiber, F., Hilliges, O., Loechtefeld, M., et al. Building interactive multi-touch surfaces. *Journal of graphics, gpu, and game tools* 14.3 (2009), 35–55 (cited on pages 13, 15).

[141] Kume, Y. Foot interface: fantastic phantom slipper. In *Proceedings of SIGGRAPH*. 1998, 114 (cited on page 27).

[142] Siek, K. A., Rogers, Y., and Connelly, K. H. Fat finger worries: how older and younger users physically interact with PDAs. In *Proceedings of INTERACT*. 2005, 267–280 (cited on page 17).

[143] SMART Technologies. *SMART Table*. URL: `http://www.smarttech.com` (cited on pages 24, 100).

[144] Smith, D. V. and Margolskee, R. F. Making sense of taste. *Scientific American* 284.3 (2001), 32–39 (cited on page 144).

[145] Srinivasan, P., Birchfield, D., Qian, G., and Kidané, A. A pressure sensing floor for interactive media applications. In *Proceedings of ACE*. 2005, 278–281 (cited on page 27).

[146] Starner, T. The Challenges of Wearable Computing: Part 1. *IEEE Micro* 21.4 (July 2001), 44–52 (cited on pages 142, 143).

[147] Starner, T. The Challenges of Wearable Computing: Part 2. *IEEE Micro* 21.4 (July 2001), 54–67 (cited on page 143).

[148] Starner, T. Human-powered wearable computing. *IBM Syst. J.* 35.3-4 (Sept. 1996), 618–629 (cited on page 145).

[149] Streitz, N. A., Geißler, J., Holmer, T., Konomi, S., Müller-Tomfelde, C., Reischl, W., Rexroth, P., Seitz, P., and Steinmetz, R. i-LAND: an interactive landscape for creativity and innovation. In *Proceedings of CHI*. 1999, 120–127 (cited on page 28).

[150] Sugiura, A. and Koseki, Y. A user interface using fingerprint recognition: holding commands and data objects on fingers. In *Proceedings of UIST*. 1998, 71–79 (cited on page 24).

[151] Tamaki, E., Miyaki, T., and Rekimoto, J. PossessedHand: a hand gesture manipulation system using electrical stimuli. In *Proceedings of AH*. 2010, 2:1–2:5 (cited on page 144).

[152] Tekscan. *F-Scan® System in-shoe plantar pressure analysis*. URL: `http://www.tekscan.com` (cited on page 28).

[153] Ullah, S., Higgin, H., Siddiqui, M. A., and Kwak, K. S. A study of implanted and wearable body sensor networks. In *Proceedings of KES-AMSTA*. 2008, 464–473 (cited on page 145).

[154] Verbunt, M. and Bartneck, C. Sensing Senses: Tactile Feedback for the Prevention of Decubitus Ulcers. *Applied Psychophysiology and Biofeedback* 35.3 (2010), 243–250 (cited on page 28).

[155] Villar, N., Gilleade, K. M., Ramduny-Ellis, D., and Gellersen, H. The VoodooIO gaming kit: a real-time adaptable gaming controller. In *Proceedings of ACE*. 2006 (cited on page 121).

[156] Visell, Y., Law, A., and Cooperstock, J. Touch Is Everywhere: Floor surfaces as ambient haptic interfaces. *Haptics* 2.3 (2009), 148–159 (cited on page 27).

[157] Vogel, D. and Baudisch, P. Shift: a technique for operating pen-based interfaces using touch. In *Proceedings of CHI*. 2007, 657–666 (cited on pages 17, 18, 19, 20, 29, 30, 37).

[158] Walther-Franks, B., Schwarten, L., Teichert, J., Krause, M., and Herrlich, M. User detection for a multi-touch table via proximity sensors. In *Extended Abstracts of TABLETOP*. 2008 (cited on page 22).

[159] Wang, F., Cao, X., Ren, X., and Irani, P. Detecting and leveraging finger orientation for interaction with direct-touch surfaces. In *Proceedings of UIST*. 2009, 23–32 (cited on pages 21, 22, 31).

[160] Wang, F. and Ren, X. Empirical evaluation for finger input properties in multitouch interaction. In *Proceedings of CHI*. 2009, 1063–1072 (cited on pages 17, 18, 19, 20).

[161] Ware, C., Arthur, K., and Booth, K. S. Fish tank virtual reality. In *Proceedings of INTERACT*. 1993, 37–42 (cited on page 117).

[162] Weiser, M. The computer for the 21st century. *SIGMOBILE Mob. Comput. Commun. Rev.* 3.3 (July 1999), 3–11 (cited on page 28).

[163] Welch, A. The thermal response of laser irradiated tissue. *Quantum Electronics* 20.12 (1984), 1471–1481 (cited on page 79).

[164] Wellner, P. The DigitalDesk calculator: tangible manipulation on a desk top display. In *Proceedings of UIST*. 1991, 27–33 (cited on page 16).

[165] Wigdor, D., Forlines, C., Baudisch, P., Barnwell, J., and Shen, C. Lucid touch: a see-through mobile device. In *Proceedings of UIST*. 2007, 269–278 (cited on pages 16, 21, 70).

[166] Wigdor, D. and Wixon, D. *Brave NUI World: Designing Natural User Interfaces for Touch and Gesture*. 1st. Morgan Kaufmann Publishers Inc., 2011 (cited on page 2).

[167] Wilson, A. D. TouchLight: an imaging touch screen and display for gesture-based interaction. In *Proceedings of ICMI*. 2004, 69–76 (cited on page 16).

[168] Wilson, A. D. PlayAnywhere: a compact interactive tabletop projection-vision system. In *Proceedings of UIST*. 2005, 83–92 (cited on page 16).

[169] Wilson, A. D. Using a depth camera as a touch sensor. In *Proceedings of ITS*. 2010, 69–72 (cited on page 16).

[170] Wilson, A. D. and Benko, H. Combining multiple depth cameras and projectors for interactions on, above and between surfaces. In *Proceedings of UIST*. 2010, 273–282 (cited on pages 2, 3, 28, 132).

[171] Wobbrock, J. O., Morris, M. R., and Wilson, A. D. User-defined gestures for surface computing. In *Proceedings of CHI*. 2009, 1083–1092 (cited on page 104).

[172] Wolpaw, J. R., Birbaumer, N., McFarland, D. J., Pfurtscheller, G., and Vaughan, T. M. Brain–computer interfaces for communication and control. *Clinical Neurophysiology* 113.6 (June 2002), 767–791 (cited on page 144).

[173] Xia, X. and O'Gorman, L. Innovations in fingerprint capture devices. *Pattern Recognition* 36.2 (2003), 361–369 (cited on page 77).

[174] Yatani, K. and Truong, K. N. SemFeel: a user interface with semantic tactile feedback for mobile touch-screen devices. In *Proceedings of UIST*. 2009, 111–120 (cited on page 144).

[175] Yin, K. and Pai, D. K. FootSee: an interactive animation system. In *Proceedings of SCA*. 2003, 329–338 (cited on pages 27, 129).

[176] Zhang, H., Yang, X.-D., Ens, B., Liang, H.-N., Boulanger, P., and Irani, P. See me, see you: a lightweight method for discriminating user touches on tabletop displays. In *Proceedings of CHI*. 2012, 2327–2336 (cited on page 22).

All links were last followed on June 20, 2013.

DECLARATION

Ich erkläre hiermit, dass

- ich die vorliegende Dissertationsschrift "3D From 2D Touch" selbständig und ohne unerlaubte Hilfe angefertigt sowie nur die angegebene Literatur verwendet habe,
- ich mich nicht bereits anderwärts um einen Doktorgrad beworben habe oder ich einen solchen besitze und
- mir die Promotionsordnung der Mathematisch-Naturwissenschaftlichen Fakultät der Universität Potsdam, veröffentlicht in den Amtlichen Bekanntmachungen Nr. 1/2004, bekannt ist.

Berlin, Juni 2013

Christian Holz