LUDWIG-
MAXIMILIANS-
UNIVERSITÄT
MÜNCHEN

# GRASP-SENSITIVE SURFACES

## Utilizing Grasp Information for Human-Computer Interaction

**DISSERTATION**

an der Fakultät für Mathematik, Informatik und Statistik
der Ludwig-Maximilians-Universität München

vorgelegt von
Diplom-Medieninformatiker

**RAPHAEL WIMMER**

geboren am 26. Dezember 1980 in Bad Tölz

München, den 19. Dezember 2014

| | |
|---|---|
| Erstgutachter: | Prof. Dr. Heinrich Hußmann,<br>Ludwig-Maximilians-Universität München |
| Zweitgutachter: | Prof. Roderick Murray-Smith, PhD,<br>University of Glasgow |

Tag der mündlichen Prüfung: 27. März 2015

# ABSTRACT

Grasping objects with our hands allows us to skillfully move and manipulate them. Hand-held tools further extend our capabilities by adapting precision, power, and shape of our hands to the task at hand.

Some of these tools, such as mobile phones or computer mice, already incorporate information processing capabilities. Many other tools may be augmented with small, energy-efficient digital sensors and processors. This allows for graspable objects to learn about the user grasping them - and supporting the user's goals.

For example, the way we grasp a mobile phone might indicate whether we want to take a photo or call a friend with it - and thus serve as a shortcut to that action. A power drill might sense whether the user is grasping it firmly enough and refuse to turn on if this is not the case. And a computer mouse could distinguish between intentional and unintentional movement and ignore the latter.

This dissertation gives an overview of grasp sensing for human-computer interaction, focusing on technologies for building grasp-sensitive surfaces and challenges in designing grasp-sensitive user interfaces.

It comprises three major contributions: a comprehensive review of existing research on human grasping and grasp sensing, a detailed description of three novel prototyping tools for grasp-sensitive surfaces, and a framework for analyzing and designing grasp interaction:

For nearly a century, scientists have analyzed human grasping. My literature review gives an overview of definitions, classifications, and models of human grasping. A small number of studies have investigated grasping in everyday situations. They found a much greater diversity of grasps than described by existing taxonomies. This diversity makes it difficult to directly associate certain grasps with users' goals.

In order to structure related work and own research, I formalize a generic workflow for grasp sensing. It comprises *capturing* of sensor values, *identifying* the associated grasp, and *interpreting* the meaning of the grasp.

A comprehensive overview of related work shows that implementation of grasp-sensitive surfaces is still hard, researchers often are not aware of related work from other disciplines, and intuitive grasp interaction has not yet received much attention.

In order to address the first issue, I developed three novel sensor technologies designed for grasp-sensitive surfaces. These mitigate one or more limitations of traditional sensing techniques:

**HandSense** uses four strategically positioned capacitive sensors for detecting and classifying grasp patterns on mobile phones. The use of custom-built high-resolution sensors allows detecting proximity and avoids the need to cover the whole device surface with sensors. User tests showed a recognition rate of 81%, comparable to that of a system with 72 binary sensors.

**FlyEye** uses optical fiber bundles connected to a camera for detecting touch and proximity on arbitrarily shaped surfaces. It allows rapid prototyping of touch- and grasp-sensitive objects and requires only very limited electronics knowledge. For FlyEye I developed a *relative calibration* algorithm that allows determining the locations of groups of sensors whose arrangement is not known.

**TDRtouch** extends Time Domain Reflectometry (TDR), a technique traditionally used for inspecting cable faults, for touch and grasp sensing. TDRtouch is able to locate touches along a wire, allowing designers to rapidly prototype and implement modular, extremely thin, and flexible grasp-sensitive surfaces.

I summarize how these technologies cater to different requirements and significantly expand the design space for grasp-sensitive objects.

Furthermore, I discuss challenges for making sense of raw grasp information and categorize interactions. Traditional application scenarios for grasp sensing use only the grasp sensor's data, and only for mode-switching. I argue that data from grasp sensors is part of the general usage context and should be only used in combination with other context information.

For analyzing and discussing the possible meanings of grasp types, I created the GRASP model. It describes five categories of influencing factors that determine how we grasp an object:

*Goal* – what we want to do with the object, *Relationship* – what we know and feel about the object we want to grasp, *Anatomy* – hand shape and learned movement patterns, *Setting* – surrounding and environmental conditions, and *Properties* – texture, shape, weight, and other intrinsics of the object

I conclude the dissertation with a discussion of upcoming challenges in grasp sensing and grasp interaction, and provide suggestions for implementing robust and usable grasp interaction.

# ZUSAMMENFASSUNG

Die Fähigkeit, Gegenstände mit unseren Händen zu greifen, erlaubt uns, diese vielfältig zu manipulieren. Werkzeuge erweitern unsere Fähigkeiten noch, indem sie Genauigkeit, Kraft und Form unserer Hände an die Aufgabe anpassen.

Digitale Werkzeuge, beispielsweise Mobiltelefone oder Computermäuse, erlauben uns auch, die Fähigkeiten unseres Gehirns und unserer Sinnesorgane zu erweitern. Diese Geräte verfügen bereits über Sensoren und Recheneinheiten. Aber auch viele andere Werkzeuge und Objekte lassen sich mit winzigen, effizienten Sensoren und Recheneinheiten erweitern. Dies erlaubt greifbaren Objekten, mehr über den Benutzer zu erfahren, der sie greift - und ermöglicht es, ihn bei der Erreichung seines Ziels zu unterstützen. Zum Beispiel könnte die Art und Weise, in der wir ein Mobiltelefon halten, verraten, ob wir ein Foto aufnehmen oder einen Freund anrufen wollen - und damit als Shortcut für diese Aktionen dienen. Eine Bohrmaschine könnte erkennen, ob der Benutzer sie auch wirklich sicher hält und den Dienst verweigern, falls dem nicht so ist. Und eine Computermaus könnte zwischen absichtlichen und unabsichtlichen Mausbewegungen unterscheiden und letztere ignorieren.

Diese Dissertation gibt einen Überblick über Grifferkennung (*grasp sensing*) für die Mensch-Maschine-Interaktion, mit einem Fokus auf Technologien zur Implementierung griffempfindlicher Oberflächen und auf Herausforderungen beim Design griffempfindlicher Benutzerschnittstellen. Sie umfasst drei primäre Beiträge zum wissenschaftlichen Forschungsstand: einen umfassenden Überblick über die bisherige Forschung zu menschlichem Greifen und Grifferkennung, eine detaillierte Beschreibung dreier neuer Prototyping-Werkzeuge für griffempfindliche Oberflächen und ein Framework für Analyse und Design von griff-basierter Interaktion (*grasp interaction*).

Seit nahezu einem Jahrhundert erforschen Wissenschaftler menschliches Greifen. Mein Überblick über den Forschungsstand beschreibt Definitionen, Klassifikationen und Modelle menschlichen Greifens. In einigen wenigen Studien wurde bisher Greifen in alltäglichen Situationen untersucht. Diese fanden eine deutlich größere Diversität in den Griffmuster als in existierenden Taxonomien beschreibbar. Diese Diversität erschwert es, bestimmten Griffmustern eine Absicht des Benutzers zuzuordnen. Um verwandte Arbeiten und eigene Forschungsergebnisse zu strukturieren, formalisiere ich einen allgemeinen Ablauf der Grifferkennung. Dieser besteht aus dem *Erfassen* von Sensorwerten, der *Identifizierung* der damit verknüpften Griffe und der *Interpretation* der Bedeutung des Griffes. In einem umfassenden Überblick über verwandte Arbeiten zeige ich, dass die Implementierung von griffempfindlichen Oberflächen immer noch ein herausforderndes Problem ist, dass Forscher regelmäßig keine Ahnung von verwandten Arbeiten in benachbarten Forschungsfeldern haben, und dass intuitive Griffinteraktion bislang wenig Aufmerksamkeit erhalten hat.

Um das erstgenannte Problem zu lösen, habe ich drei neuartige Sensortechniken für griffempfindliche Oberflächen entwickelt. Diese mindern jeweils eine oder mehrere

Schwächen traditioneller Sensortechniken:

**HandSense** verwendet vier strategisch positionierte kapazitive Sensoren um Griffmuster zu erkennen. Durch die Verwendung von selbst entwickelten, hochauflösenden Sensoren ist es möglich, schon die Annäherung an das Objekt zu erkennen. Außerdem muss nicht die komplette Oberfläche des Objekts mit Sensoren bedeckt werden. Benutzertests ergaben eine Erkennungsrate, die vergleichbar mit einem System mit 72 binären Sensoren ist.

**FlyEye** verwendet Lichtwellenleiterbündel, die an eine Kamera angeschlossen werden, um Annäherung und Berührung auf beliebig geformten Oberflächen zu erkennen. Es ermöglicht auch Designern mit begrenzter Elektronikerfahrung das Rapid Prototyping von berührungs- und griffempfindlichen Objekten. Für FlyEye entwickelte ich einen *relative-calibration*-Algorithmus, der verwendet werden kann um Gruppen von Sensoren, deren Anordnung unbekannt ist, semi-automatisch anzuordnen.

**TDRtouch** erweitert Time Domain Reflectometry (TDR), eine Technik die üblicherweise zur Analyse von Kabelbeschädigungen eingesetzt wird. TDRtouch erlaubt es, Berührungen entlang eines Drahtes zu lokalisieren. Dies ermöglicht es, schnell modulare, extrem dünne und flexible griffempfindliche Oberflächen zu entwickeln.

Ich beschreibe, wie diese Techniken verschiedene Anforderungen erfüllen und den *design space* für griffempfindliche Objekte deutlich erweitern. Desweiteren bespreche ich die Herausforderungen beim Verstehen von Griffinformationen und stelle eine Einteilung von Interaktionsmöglichkeiten vor. Bisherige Anwendungsbeispiele für die Grifferkennung nutzen nur Daten der Griffsensoren und beschränken sich auf Moduswechsel. Ich argumentiere, dass diese Sensordaten Teil des allgemeinen Benutzungskontexts sind und nur in Kombination mit anderer Kontextinformation verwendet werden sollten.

Um die möglichen Bedeutungen von Griffarten analysieren und diskutieren zu können, entwickelte ich das GRASP-Modell. Dieses beschreibt fünf Kategorien von Einflussfaktoren, die bestimmen wie wir ein Objekt greifen:

*Goal* – das Ziel, das wir mit dem Griff erreichen wollen, *Relationship* – das Verhältnis zum Objekt, *Anatomy* – Handform und Bewegungsmuster, *Setting* – Umgebungsfaktoren und *Properties* – Eigenschaften des Objekts, wie Oberflächenbeschaffenheit, Form oder Gewicht.

Ich schließe mit einer Besprechung neuer Herausforderungen bei der Grifferkennung und Griffinteraktion und mache Vorschläge zur Entwicklung von zuverlässiger und benutzbarer Griffinteraktion.

# Danksagung

Wie oft wurde der *Kairos* der Fertigstellung durch meta-wissenschaftliche und lehrende “Ablenkung” versäumt, bevor die Erkenntnis dieses traurigen Faktums einer bemerkenswerten Mischung aus eherner professoraler Geduld (wie Liebenswürdigkeit), sanftem aber unerbittlichem familiärem Druck und wohl auch ein wenig der beklagenswerten Eitelkeit weichen durfte.

Umso umfangreicher ist deshalb die Liste von Personen, denen ich aus ganzem Herzen danken möchte.

Meinem Doktorvater **Heinrich Hußmann** danke ich für das Vertrauen und die unbedingte Unterstützung und Förderung meiner Promotion und vielgestaltiger Nebenprojekte.

Bei meinem Zweitgutachter **Rod Murray-Smith** möchte ich mich für die unkomplizierte Begutachtung und sein geduldiges Warten auf die Fertigstellung bedanken.

Beiden Gutachtern danke ich für ihr positives und konstruktives Feedback, das sie mir vor und nach der Abgabe dieser Dissertation zukommen ließen.

“E pluribus unum” - “aus vielem eines” - so könnte das Motto dieser Dissertation lauten. Zu ihr haben - auf direkte und indirektere Weise - viele Menschen beigetragen, die ich kennenlernen durfte.

Besonders bedanken möchte ich mich bei den Mitgliedern der damaligen Nachwuchsforschungsgruppe *Embedded Interaction*, **Albrecht Schmidt**, **Matthias Kranz** und **Paul Holleis**. Diese führten mich - und viele andere Studenten an der LMU - in die wissenschaftliche Praxis ein und schufen ein Umfeld, das in mir den Wunsch zur Promotion weckte. In dieser Gruppe entstanden meine ersten Experimente mit kapazitiven Sensoren, die den Grundstein für meine weitere Arbeit bildeten.

Der praktische Teil meiner Dissertation entstand von 2006 bis 2011 am Lehrstuhl für Medieninformatik der Ludwig-Maximilians-Universität München.

Ich möchte dem kongenialen, wundervollen Team dort danken, besonders **Rainer Fink**, der immer da ist, wenn man ihn braucht (und auch sonst immer), **Doris Hausen**, meiner unersetzbaren Büro-Gattin, Lebensretterin und Tratschtante, **Dominikus Baur**, mit dem ich Passwort und Pessimisums teilte, **Sebastian Boring**, mit dem ich schon Büro und Bett teilte, **Sebastian Löhmann**, mit dem ich Freud und Leid der Hardware teilte, **Alexander Wiethoff**, mit dem ich meinen Lieblingskurs *Sketching with Hardware* teilte, **Alexander De Luca**, der immer sagt, was Sache ist, **Andreas Butz**, der immer hinter einem steht, wenn man ihn braucht, **Heiko Drewes**, der immer für eine Diskussion zu haben ist, **Fabian Hennecke**, Meister des Flachwitzes, mit dem zusammen ich das *Curve*-Projekt gestartet habe, **Hendrik Richter**, der mich mit Ackerschnackern bekannt gemacht hat, **Gregor Broll**, von dem ich erstmals (und mehrmals täglich) das Wort *deliverables* gehört habe und mit dem mich eine lange (und hoffentlich endliche) Disserta-

tionsprokrastination verbindet, **Max Maurer**, unverdrießlicher Meister des Barkeepers, **Alina Hang**, die positivste Pessimistin, die ich kenne, **Sara Streng**, die den Lehrstuhl zusammengebracht hat, **Bettina Conradi**, von der ich lernte, dass die *Bane* eigentlich *Gotto* heißt, **Richard Atterer**, of *Destrictor* fame, der umfassende Kompetenz mit sonnigstem Charakter kombiniert, **Andreas Pleuß**, **Arnd Vitzthum** und **Siegfried Wagner**, die den Lehrstuhl aufbauten und mir viele wichtige Grundlagen vermittelten und den Sekretärinnen des Lehrstuhls, **Sybille Thomsen**, **Anita Szasz** und **Franziska Schwamb**, die aus einem Haufen kleiner Doktoranden ein effektives und glückliches Team gemacht haben.

Aus familiären Gründen wechselte ich 2011 an den Lehrstuhl für Medieninformatik der Universität Regensburg, wo ich dann neben meiner Lehrtätigkeit meine Dissertation schrieb.

Auch hier hatte ich das Glück, Teil eines wundervollen Teams werden zu dürfen. Ich danke besonders **Christian Wolff**, der mir stets Rückhalt, Freiraum und Denkanstöße gab, **Markus Heckner**, der mir den Wechsel nach Regensburg ermöglichte, den Einstieg erleichterte, und vieles vorantrieb, **Alexander Bazo**, mit dem ich viele Ansichten und auch ein Zimmer teile, **Patricia Böhm**, Spezialistin für tiefgründige Gespräche, **Victoria Böhm**, wunderbar wertende Stimmungsdetektorin, **Manuel Burghardt**, digitaler Humanist im breitesten Sinne, **Martin Brockelmann**, harter Kern des Lehrstuhls und dreidimensionaler Meister, **Florian Echtler**, omnikompetenter Hacker und langjähriger Freund, **Felix Raab**, **Tim Schneidermeier**, **Thomas Wilhelm**, drei überaus angenehmen Kollegen und **Ingrid Stitz**, **Ingrid Böhm** und **Susanne Klinger**, die Ordnung und Wärme in diesen Lehrstuhl bringen.

Das finale Projekt im Rahmen meiner Dissertation, *TDRtouch*, entstand in intensiver Zusammenarbeit mit **Patrick Baudisch**, dessen Vertrauen, Wissen und sanfter Druck mir unendlich geholfen haben. Auch den Mitgliedern seines Lehrstuhls, insbesondere **Christian Holz**, **Stefanie Müller**, **Sean Gustafson**, **Anne Roudaut**, **Henning Pohl**, **Pedro Lopes** und **Sieglinde Tholen** möchte ich für die Gastfreundschaft und die vielen interessanten Gespräche danken.

Mit **Katrin Wolf**, **Thomas Feix** und **Tobias Große-Puppendahl** - die ich erstmals als Doktoranden in Berlin, Wien und Darmstadt kennenlernen durfte, verbinden mich erfüllende Diskussionen über technische, wissenschaftliche, und berufsperspektivische Themen, die ich nicht missen möchte.

Auch wenn er sich vermutlich nicht mehr daran erinnert, möchte ich **Mark Gross** dafür danken, dass er mir bei meiner ersten TEI-Demo gesagt hat, wie schlecht ich diese präsentieren würde, und was ich stattdessen machen sollte. Dieses unerbetene und gleichzeitig so hilfreiche Feedback hat einem einsamen Doktoranden auf einer fremden Konferenz das Gefühl gegeben, in dieser Community doch ganz gut aufgehoben zu sein.

Meine von mir verfasste Dissertation ist über etwa sieben Jahre neben meiner Forschungs- und Lehrtätigkeit als junger Familienvater in mühevollster Kleinstarbeit

entstanden, und sie enthält fraglos Fehler. Und über jeden einzelnen dieser Fehler bin ich selbst am unglücklichsten.

Deshalb bin ich meinem Bruder **Johannes Wimmer** zu besonderem Dank verpflichtet, der sich große Teile dieser Dissertation vor Fertigstellung durchgelesen und mir wertvolle Hinweise zur Verbesserung gegeben hat.

Auch meinen anderen **Geschwistern** und meinen **Eltern** möchte ich für die stete und liebevolle Unterstützung seit vielen Jahrzehnten und besonders in den letzten Jahren danken.

Vor allem aber möchte ich meiner Frau **Monika** und meinen Söhnen **Noah**, **Elija** und **Samuel** danken, die viel zu lange und zu oft hinter meiner Arbeit zurückstehen mussten, und mich trotzdem jede Minute davon unterstützten und mich erdeten.

**You da real MVPs!**

# Table of Contents

# I

# Introduction and Related Work

# Chapter 1

# Introduction

*The ability to grasp objects, manipulate them, and use them as tools, is one of the most prominent traits of* homo sapiens. *Human hand function and prehensile manipulation have been subject of intensive research for over a century. In robotics, researchers take inspiration from human grasping to develop better robotic grippers. Only in recent years have researchers started to explore how human grasping might be utilized in human-computer interaction. In this dissertation I propose* grasp interaction *as a novel concept for interacting with graspable objects. Here, the information about the way persons grasp objects is used to support them in interacting with the object, or the digital information it represents. This dissertation is divided into three major parts: an overview of existing research on human grasping and grasp sensing, descriptions and discussions of three grasp sensing techniques I developed that support rapid prototyping of grasp-sensitive objects, and a broad discussion of important issues in grasp interaction.*

**Attribution:** This dissertation is based on and expands on research I conducted alone and together with others from 2006 to 2011. Parts of this research have been published previously. In order to document the origins and authors of all ideas and insights, I attribute such publications and my collaborators at the beginning of each chapter. This introductory chapter only contains original content and is not directly based on previous work.

## 1.1 Introduction

Along with language, grasping and prehensile manipulation might be the most outstanding human capabilities. While not only humans are able to grasp objects and use them as tools, the versatility of our hands has allowed mankind to dominate other species and dramatically change the shape of the world. The prehensile hand, gifted to humans by evolution, allowed our predecessors to replace the slow progress of evolution with the tremendously faster progress of culture.

Major waypoints of mankind's cultural development - making fire, weaving clothes, building spears, carving a wheel, writing down laws and lore, constructing a sailboat, painting the Mona Lisa, assembling a steam engine, or programming a computer - were all made possible only because several million years ago, our predecessors gained an opposable thumb, enabling them to grasp their environment and utilize it.

At least 3.39 million years ago predecessors of *homo sapiens* started using stone tools for removing flesh from bones and cracking these (McPherron et al. 2010). Since then, tools have made the human hand more and more powerful, the rock being superseded by knifes, guns, and recently joysticks allowing control of armed drones. Tools have also significantly increased the hand's precision in the form of pliers, tweezers, or remote-controlled robot-hands for minimally invasive surgery.

However, hand-held tools not only allow powerful and precise manipulation of the physical world. With the advent of the digital realm, keyboard, mouse, digital pens, and various specialized input devices have emerged as new tools that allow us to access and modify digital information. Touch screens put information directly under our fingertips. More recently, *Tangible User Interfaces (TUIs)* allow perception and manipulation of digital information through physical artifacts (Fitzmaurice, Ishii, and Buxton 1995; Ishii and Ullmer 1997). All these tools - many of them graspable - aim to make human-computer interaction more efficient, intuitive, or fulfilling.

However, these tools only capture and use a small portion of the hand's capabilities. Common input devices only sense the location of one or more fingertips - using discrete buttons or touch-sensitive surfaces - and sometimes the movement of the whole hand grasping the input device. The expressiveness of the human hand is more or less ignored. What if we could sense in which way a person grasps a hand tool, an input device, or a TUI, and use this information to make their interaction with the physical or digital world more expressive? I call this concept of utilizing grasp information for interacting with grasp-sensitive objects *grasp interaction*.

## 1.2 Why Grasp Interaction?

Grasping offer a yet mostly unused, very expressive input channel for interaction with smart objects. This input channel is always present while we are interacting with the object, and it is bi-directional, as the object can give immediate haptic feedback to the grasping hand.

Grasp interaction can enhance usability of many different graspable artifacts. A good example to illustrate the concept of grasp interaction is the grasp-sensitive mobile phone - a popular theme in research (Kim et al. 2006; Wimmer and Boring 2009; Tsukamoto, Yuta, and Okada 2014). The following *'day in the life'* scenario gives an overview of applications for such a device. The user's actions triggering grasp interaction are highlighted *in italics*.

Tom wakes up to the sound of his mobile phone's wakeup alarm. It is Thursday, 07:00. He slightly *squeezes* it, thereby silencing the alarm tone and activating the snooze function. When the phone reminds him for the third time to stand up already, Tom finally gets up and *picks up* the phone. The phone senses that Tom is standing while holding it, infers that he has to be awake to do this, and decides to switch off the alarm.

When Tom leaves the house, he picks up the phone from the table and *'feels'* its battery status like he would feel the pulse of a person. The phone recognizes this gesture and responds with a certain *haptic feedback pattern* that indicates a nearly full battery. Tom *puts the phone into his pocket*. The phone detects this and switches from acoustic to haptic notifications. It also locks the touchscreen to avoid inadvertent calls and immediately turns off the screen to save power.

At work, Tom's phone starts vibrating in his pocket. As soon as he *starts pulling it out* of the pocket, the phone recognizes that he might want to identify the caller. It reduces the strength of the vibration and activates the display. Additionally, the phone automatically identifies Tom as the person *holding it*, based on Tom's characteristic grip pattern. The phone display shows that a client of Tom's company is calling. Tom is very busy at this moment, so he *hands it over to* his coworker, Sally. The phone recognizes Tom's hand, Sally's hand, and the handover from Tom to Sally. It infers that Tom wants to allow Sally to use it. Therefore, it grants Sally limited access rights, including the right to accept the incoming call Sally *holds the phone to her ear*, thereby signaling to the phone that she wants to accept the call. She confirms the important meeting taking place on the following day and *enters the event into the calendar on Tom's phone*. The phone allows this due to the credentials set up when Tom handed her the phone.

Before Tom leaves the office later in the day, he takes his phone, *holds it like a text marker*, and circles several files on his computer screen that he wants to take home with him. The phone, having switched to 'selection mode', instantly copies these files into internal storage.

In the subway, Tom needs to write a text message. As soon as Tom *holds the phone in both hands*, thumbs on the display, the phone automatically activates the on-screen keyboard and show the 'compose text message' dialog. When Tom has to leave the subway, he has to pick up his briefcase while still typing the message. The phone recognizes that Tom now *holds it with only one hand* and automatically switches to a keyboard layout optimized for one-handed use.

> Outside the subway station, Tom sees a peacock sitting on a huge smiling cat. As this is not a common sight in such scenarios, Tom pulls out the phone and *holds it in landscape orientation* like a digital camera. The phone recognizes this grasp and it automatically activates the camera application, allowing Tom to focus on taking the photo. Tom takes the photo and *gently strokes* his versatile and attentive phone. The phone responds with soft purring because its designers thought this would be cute.

As explored in this scenario, identifying who grasps an object and inferring their goal offers many opportunities for enhancing human-computer interaction, making it more efficient, secure, or enjoyable. Furthermore, grasp interaction may also augment existing, limited input techniques on mobile devices.

With tangible user interfaces, digital information is traditionally manipulated by moving, rotating, or deforming physical representations. Knowing who grasps a tangible artifact - and with which goal - makes tangible interaction more expressive. Grasp interaction may also augment the coarse movements with a further layer of prehensile movements.

Classic tools may benefit from grasp sensing, too. For example, a power drill might only turn on when it is held in a secure grasp, reducing injuries. A golf club might offer suggestions on how to improve the grip. And a children's toy might make different sounds, depending on the way it is being held.

At the time of writing (2012 - 2014), grasp interaction is still in its infancy. As I discuss in this dissertation, research on grasp interaction has come a long way but is still limited by three systematic weaknesses.

a) Research on grasp sensing has been conducted in various fields, such as robotics, prosthetics, neuroscience, and human-computer interaction. However, oftentimes researchers are apparently not aware of existing approaches in their own and related fields. Therefore, researchers duplicate work without necessity, are unaware of existing (better) solutions for their problems, and fail to address important general issues identified earlier.
b) Most research prototypes use custom-built grasp-sensitive surfaces. As the effort to design and implement such hardware is high, researchers have little time and incentive to explore the design space more thoroughly. Thus, the research prototypes presented so far support only narrow, pre-defined interactions. The lack of versatile, easily usable prototyping tools for grasp interaction discourages researchers without a background in electrical engineering.
c) Research has so far focused on two important sub-topics: implementing grasp-sensitive surfaces and identifying grasps using machine learning. Little thought has yet gone into human factors, theoretical models, design concepts, or limits of grasp interaction.

Therefore, this dissertation contains three core contributions:

a) I provide a comprehensive overview and discussion of research on human grasping, grasp taxonomies, and grasp-sensing approaches
b) I present three novel techniques for prototyping grasp-sensitive surfaces: CapToolKit, FlyEye, and TDRtouch.
c) I propose and discuss important concepts in grasp interaction, including GRASP, a descriptive model of meaning in grasps.

## 1.3 Research Approach

This dissertation presents cross-disciplinary, exploratory research in human-computer interaction (HCI). Grasp interaction incorporates research from many other related areas: human anatomy and behavior, ergonomics, sensor design, digital signal processing, machine learning, rapid prototyping, mobile interaction, tangible interaction, and model theory. Therefore, the core contributions mentioned above had to be discovered and defined incrementally along the way instead of being postulated beforehand.

Mackay and Fayard (1997) propose a *triangulation* framework for scientific research in HCI which comprises three main classes of research: theory, design, and observation. Research within each of these classes inspires and requires research within the other classes. The framework allows for structuring the individual steps that comprise this dissertation (Figure 1.1).

HandSense (Wimmer and Boring 2009) - the first of three hardware prototypes developed for this dissertation - was originally intended as a proof-of-concept application for CapSense - a capacitive sensing toolkit I developed from 2006 to 2008. The user study, a demonstration at TEI 2009, subsequent discussions, and a literature review on this topic shaped subsequent research.

As it became apparent that research pertaining to grasp interaction was scattered across several, isolated disciplines, gathering and organizing this research was necessary. Preliminary results and thoughts were first published in my paper "Grasp Sensing for Human-Computer Interaction" (Wimmer 2011a).

The literature review also showed that research on grasp sensing was mainly technology-driven. In order to allow non-technical scientists to contribute to this research area, easily adaptable prototyping techniques are required. My research on such techniques (Wimmer and Boring 2009; Wimmer 2010a; Wimmer and Baudisch 2011) informed my theoretical research in turn.

Furthermore, it also became apparent that researchers routinely ignored the actual complexity of human grasping. In order to demonstrate this problem, I conducted a study

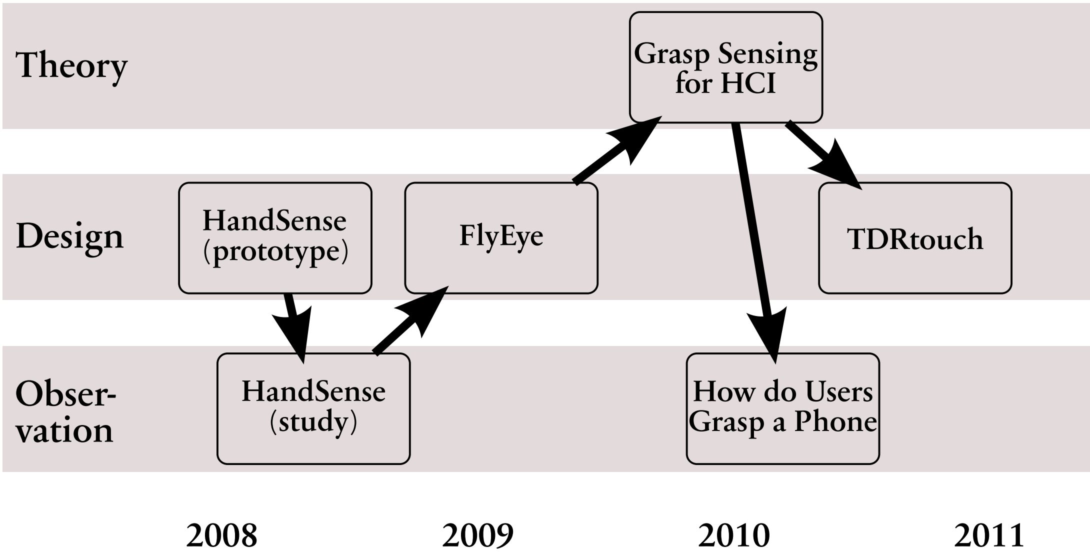


**Figure 1.1:** Timeline of my research on grasp interaction, loosely based on the *triangulation* framework by Mackay and Fayard (1997).

on variability in grasps (see Section 2.4) and proposed a descriptive model of meaning in grasps (Wimmer 2011a).

This dissertation links the findings together and gives a thematic structure to the insight that I experienced in chronological order.

## 1.4 Structure of the Dissertation

The dissertation is organized into three major parts, each comprised of multiple chapters:

Part I, *Introduction and Related Work*, continues with an overview of human grasping, grasp taxonomies, and research on everyday grasping (Chapter 2).

Chapter 3 introduces the focus of my research, presenting a new, interaction-centric definition of grasping, important definitions, and a generic grasp sensing workflow.

In Chapter 4 I give a comprehensive and in-depth overview of previous research on grasp sensing and discuss properties and limitations of existing approaches.

Part II, *Technologies for Grasp-Sensitive Surfaces* comprises descriptions and discussions of three grasp sensing techniques that I designed, implemented, and evaluated:

*HandSense* (Chapter 5), a grasp-sensitive prototype employing capacitive sensing, *FlyEye* (Chapter 6), a rapid prototyping technique using optical fiber which does not require an understanding of electronics, and

*TDRtouch* (Chapter 7), which allows for sensing multiple simultaneous touches along a single cable.

In Chapter 8 I discuss and compare the properties of these techniques.

Part III, *Towards Grasp Interaction*, discusses important issues on the path from reliable grasp sensing to rich grasp interaction.

Chapter 9 gives an introduction into challenges and promising approaches in grasp interaction.

Chapter 10 introduces a yet unpublished study investigating how users grasp a mobile phone, showing that grasps vary significantly between individuals even for simple, constrained everyday tasks.

Chapter 11 presents GRASP, a descriptive model of meaning in grasps, which allows for discussing and analyzing human grasping and serves as a framework for developing grasp-sensitive applications.

Chapter 12 discusses the relationship between grasp interaction and other interaction paradigms, such as tangible interaction. It also contains a list of design suggestions for grasp-sensitive applications.

A final discussion concludes the dissertation (Chapter 13).

Most of my research has already been published in peer-reviewed venues. Therefore, this thesis explicitly builds upon these publications. The theoretical concepts and use cases presented in this thesis were developed in conjunction with the hardware prototypes, and were therefore originally distributed across different publications. To ease reading, I have collected theoretical discussion, concepts, and use cases in separate, cohesive chapters. Each chapter has a short *summary* at its beginning that is intended to aid in skimming the thesis.

While I am the sole author of this dissertation, it is based on previous publications and the work I did together with others. Therefore, each chapter also contains an *attribution* paragraph that references the publications and persons that contributed to the research described in the chapter.

All websites mentioned in footnotes within this dissertation have been archived with WebCite[1]. To access the archived copies, prepend `http://www.webcitation.org/query?date=2015-04-19&url=` to the URL (without the "http://" protocol identifier).

[1] see `http://webcitation.org`

# Chapter 2

# Human Grasping

*This chapter gives an overview of research on human grasping. Since the beginning of the 20th century, researchers have proposed classification, taxonomies, and models for describing grasps. Napier (1956) proposed the most commonly cited distinction between* power grasps *involving the palm and* precision grasps *involving only the fingertips. Feix et al. (2009b) collected and consolidated many existing classification schemes for human grasps into a common taxonomy containing 33 distinct grasp types. While research on human grasping was driven by rehabilitation medicine in the first half of the last century, neuroscience and robotics have since contributed greatly to our knowledge. Only recently, grasping has become a topic of research within the human-computer interaction community. A small number of studies have investigated grasping in everyday situations. They found a much greater diversity of grasps than existing taxonomies propose. Grasp variability, i.e. how much a grasp changes when repeated, depends on the person, the object, and on how frequently the grasp is conducted.*

**Attribution:** Parts of this chapter are based on my paper "Grasp Sensing for Human-Computer Interaction". The extensive discussions of related work by MacKenzie and Iberall (1994) and Feix (2011) probably influenced which publications I chose to present in detail.

## 2.1 The Human Hand

> "Die Hand macht den Menschen, das vernünftige Tier, geschickt für die Handhabung aller Dinge; sie ist sein äußeres Gehirn"

*Immanuel Kant, as cited by (Schlesinger 1919, p.321)*

The human hand is made of 27 bones (Jones and Lederman 2006, p.14) operated by 29 muscles (Jones and Lederman 2006, p.16). It consists of a palm and five fingers (Figure

2.1) . The thumb is *opposable* which means that it can be positioned in a certain way so that its tip opposes the other fingertips. This allows the hand to grasp objects between the thumb and the other fingers. Hand size varies significantly between humans, depending on age, sex (Dreyfuss 1966), nationality (Imrhan 2000), occupation, and possibly other factors. Average hand length within the U.S. male/female adult population is approximately 19/17.5 cm; average hand width is 10/8.5 cm (Dreyfuss 1966).

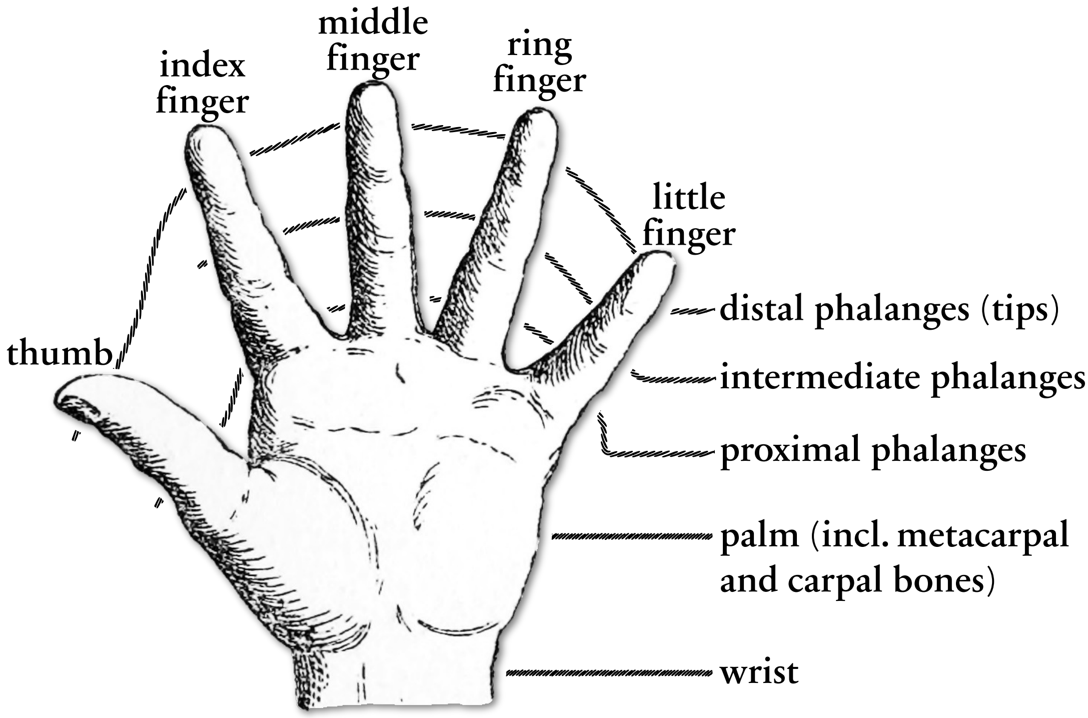


**Figure 2.1:** The human hand has a palm and five fingers. The thumb consists of two phalanges (finger bones), whereas the other fingers consist of three phalanges. The wrist connects the hand to the lower arm. Illustration adapted from Jackson (1865) (Public Domain).

When discussing the anatomy and movements of the hand, directions are designated by the terms *palmar* or *volar* (referring to the inside of the hand), *dorsal* (back of the hand), *distal* (towards the fingertips) and *proximal* (towards the wrist). *Adduction* describes the act of closing the fingers together so that their sides touch each other. *Abduction* describes the act of spreading the fingers. *Flexion* means bending the fingers inwards towards the palm. *Extension* means bending the fingers outwards into a straight posture (Figure 2.2). For further reference, Jones and Lederman (2006) discuss anatomy and function of the human hand in great detail.

While we use our hands for a variety of tasks, one of the most common actions is grasping an object. This allows for feeling the properties of an object, moving it, manipulating it, or using it as a tool.

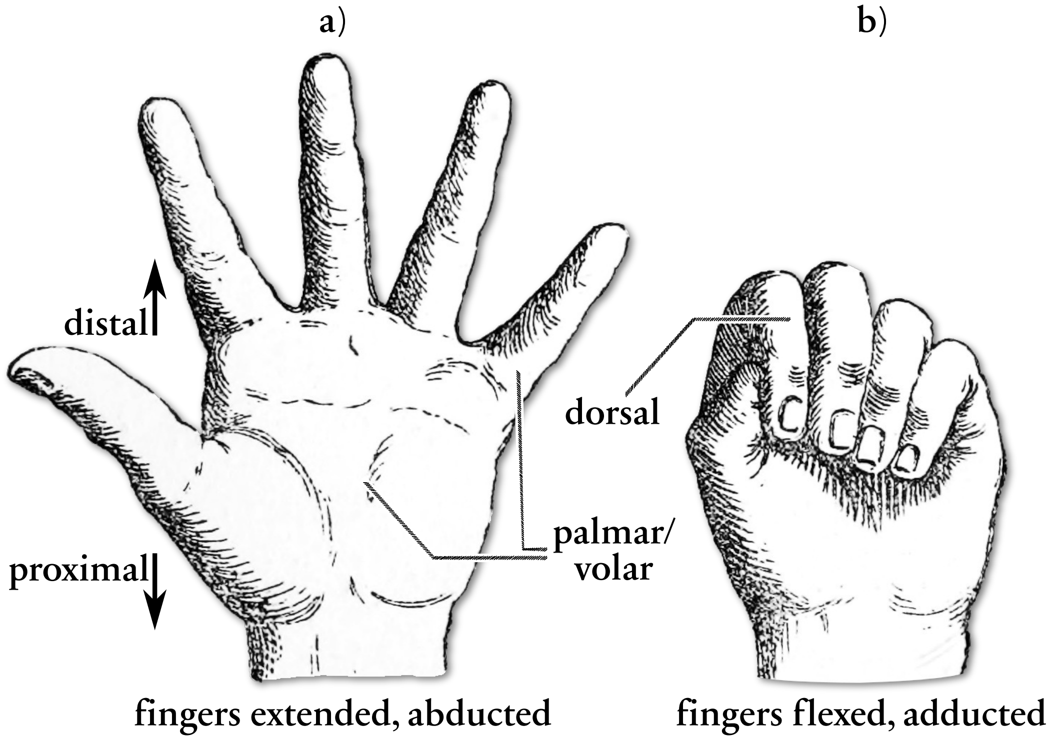


**Figure 2.2:** All fingers of the human hand can be extended (a) or flexed (b). Furthermore, fingers can be abducted (i.e. spread, a) or adducted (b). The terms palmar/volar, dorsal, distal, and proximal indicate directions on the hand. Illustrations adapted from Jackson (1865) (Public Domain).

MacKenzie and Iberall (1994 p.6) claim that grasping a tool enhances either the hand’s power or its precision. It can be argued that tools enhance the hand in even more ways. They also change the rigidity (hammer) and shape (screwdriver) of the contact area between human and object. Hand-operated controls are also used for steering automobiles, ships, and aircraft, combining a human’s perception and information processing with a machine’s power, precision, and speed. During the last century, hand-held tools also started to improve our cognitive capabilities. Pocket calculators, mobile phones, music players, digital cameras, or computer mice changed the way we process and create information.

## 2.2 What Does “Grasping” Mean?

> “Die Hand des Menschen ist durch ihre Vielseitigkeit außerordentlich verletzlich; sie braucht daher für die Ausübung der meisten Berufe eine Bewaffnung. Diese Bewaffnung liegt in der Vielheit der Werkzeuge, mit denen

> sie durch die Umspannung des Werkzeuges ein Ganzes bildet."

*Schlesinger (1919), p.322*

As the Latin roots of the word *definition* suggest, the goal of a definition is to include certain concepts and exclude others, precisely delimiting what falls within the definition and what not. However, in reality, numerous imprecise - often contradicting - definitions exist for most concepts. This circumstance may have different reasons. There might be disagreement between researchers whether certain border-cases should be included in a definition or not, resulting in competing definitions. Sometimes, different terms are used for describing the same concept. Often, incompatible definitions of the same concept emerge in different fields of science in parallel.

In the following, I discuss different definitions of "grasping" from related work and provide an own definition based on the functions of a grasp.

### 2.2.1 Grasping, Gripping, and Prehension

The substantives **grip** and **grasp** and their accompanying verbs are both used regularly to describe the same or similar states and actions[1].

The American Heritage Dictionary (2011) offers the following definitions for *grasp* and *grip*:

> **grasp**: "[a] firm hold or grip"

> **grip**: "[a] tight hold; a firm grasp"

This recursive (and slightly nonsensical) definition implies that both terms may be used in place of each other.

There seems to be a preference for employing the term *grasping* for the *dynamic* act of reaching for an object and closing the fingers around it , whereas *gripping* has more of a *static* connotation, describing only the act of holding an object.

*Grasp* seems to be more prevalent than *grip* in robotics research. In human-computer interaction, there seems to be no clear preference for either term. For example, Kry and Pai (2006a) and Taylor and Bove (2009) choose *grasp* whereas Veldhuis et al. (2004) and Kim et al. (2006) prefer *grip*. I have used *grip* in my first paper on this topic (Wimmer and Boring 2009) but switched to *grasp* in subsequent publications (Wimmer 2010a; Wimmer 2011a; Wimmer 2011b; Wimmer and Baudisch 2011).

[1] As of 25.01.2013, Google Scholar lists 668,000 hits for the search terms 'grasp hand' and 697,000 hits for 'grip hand'. This supports the assumption that both are used interchangeably.

**Prehension** is another term that is regularly used instead of *grasping* in the life sciences.

Again, the American Heritage Dictionary (2011) offers little help in determining semantical differences between both terms:

> **prehension**: "[t]he act of grasping or seizing."

Other dictionaries contain very similar definitions. The term *prehension* is rarely used in human-computer interaction and robotics.

Overall, there seem to be few commonly accepted semantic differences between *gripping*, *grasping*, and *prehension*. Researchers usually choose one of the three terms without justifying their choice, or use the terms interchangeably[2]. To my knowledge, the distinction between those terms has not been discussed in scientific literature. To avoid ambiguity, *grasp* and *grasping* are used throughout this dissertation.

### 2.2.2 Grasp Phases

Many researchers describe grasping as an action composed of three phases (Kang and Ikeuchi 1991; Jones and Lederman 2006, p.101):

- **pre-grasp or reaching phase**, wherein the hand is being moved towards the object and assumes a compliant posture
- **grasp phase**, wherein the fingers close around the object, and
- **manipulation phase**, wherein the grasped object is used or manipulated.

There has been some discussion on whether the reaching phase (often also called *reach-to-grasp movement*) and the grasp phase might be better described as one combined movement. Jeannerod (1986) argues that both are distinct, as experiments show that the characteristics of the reaching motion are determined only by the object's location while the grasping movement is determined only by the object's intrinsic properties. Whether both motions are really controlled separately has been a topic of discussion since (Smeets and Brenner 1999; Kamp and Zaal 2007). Unless explicitly mentioned, the research presented in the following pertains to grasp phase and manipulation phase.

[2] For example, Kroemer (1986) (who proposes a classification of grasps without referencing relevant earlier work) happily calls everything a *grip* - except for Napier's *"power grip"* which he calls power grasp. No reason is given for this change.

### 2.2.3 Existing Definitions for *Grasping*

Most publications discussing grasps do not propose or reference a definition of their subject. However, there are a few exceptions:

Cutkosky (1989) define a *grasp* as

> *"a set of contacts on the surface of the object"*

This or a similar definition can be found in various other robotics papers. In contrast to most other definitions, it does not require a human hand (or any hand at all) to be involved.

Despite naming their book "The Grasping Hand", MacKenzie and Iberall do not provide a definition of grasping. However, they propose an extremely broad definition of *prehension* (MacKenzie and Iberall 1994, p.6):

> *"the application of functionally effective forces by the hand to an object for a task, given numerous constraints"*

They note that prehensile behavior can also be accomplished by other body parts, such as tails, tongues or teeth (MacKenzie and Iberall 1994, p.6).

Feix et al. (2009a) define a *grasp* as:

> *"every static hand posture with which an object can be held securely with one hand, irrespective of the hand orientation."*

In his subsequent PhD thesis, Feix (2011) concedes that this definition excludes grasps with intrinsic hand movements, bi-manual grasps, and grasps where the object is held to the hand only due to gravity. Due to these restrictions, Feix states that *"[f]or the remainder of the thesis the term grasp will refer to all kind of grasps, not only ones that are in accordance with the grasp definition."* (Feix 2011, p.31)

Böhme (2011) offers a relatively precise definition of *grasping* in his dissertation on grasp affordances:

> *"To use one hand to establish physical control of an object so that it can be used in a subsequent action. At least two surfaces of the hand must exert opposing forces in order to hold the object."*

He notes that this definition is intentionally restricted to one hand and does not cover *form closure* grasps such as the *hook grasp*[3].

In Section 3.3 I analyze limitations of these definitions and provide a new, interaction-centric definition. Until then, any of the aforementioned definitions is sufficient. However, the following terms will be used throughout this dissertation to reduce ambiguity:

***specific grasp*** a unique grasp, i.e. a concrete grasp that can be observed (used for emphasizing the distinction to a grasp type).

***grasp type*** an abstraction of a grasp which comprises functionally or anatomically similar specific grasps.

***hand posture*** a certain configuration of finger joints - which is not necessarily a grasp as the hand might not be in contact with an object.

## 2.3 Grasp Taxonomies and Descriptions

Over the last 100 years, researchers have proposed a variety of approaches for describing and classifying grasps. These can roughly be divided in *taxonomies* - which arrange exemplary phenotypes of grasps in a hierarchical tree structure - and *descriptive models* which offer a method for describing arbitrary grasps based on their properties. With taxonomies, authors usually present a set of grasp types and argue that this set covers most or all grasps, and that all grasp types are distinct. Taxonomies and descriptions are usually invented for a specific purpose, such as identifying which types of grasps an artificial hand has to implement. Therefore, different areas of research prefer different taxonomies and descriptions. Both taxonomies and descriptions generally focus on describing static grasps. Reaching phase and manipulation phase are generally excluded.

Most of the taxonomies have also been incorporated into the comprehensive grasp taxonomy assembled by Feix et al. (2009b). It condenses 147 descriptions of grasps found in 17 publications into a set of 33 unique grasps. Grasps are classified by Napier's power/precision distinction, opposition type, thumb position, and involved fingers . The hook grasp is not included because the taxonomy requires at least two contact surfaces to apply opposing forces.

While the hand's anatomy has been subject of research for centuries[4], scientific interest into human grasping increased greatly during and after World War I, when surgeons and engineers had to develop prosthetic arms and hands for mutilated veterans. In order to define which actions a prosthetic hand should be able to perform, it was necessary to first describe existing grasps whose functions should be replicated.

[3] see Section 2.3.

[4] While Bell (1833) seems to be the first scientific publication on the anatomy of the human hand, it does not explicitly discuss *grasping*.

In a talk given in 1915, Bonnet distinguishes two functionally different types of grasps: plier grasps ("Zangengriff") involving the thumb and one or more other fingers, and hook grasps ("Hakengriff") which do not involve the thumb (Bonnet 1915, p.9).

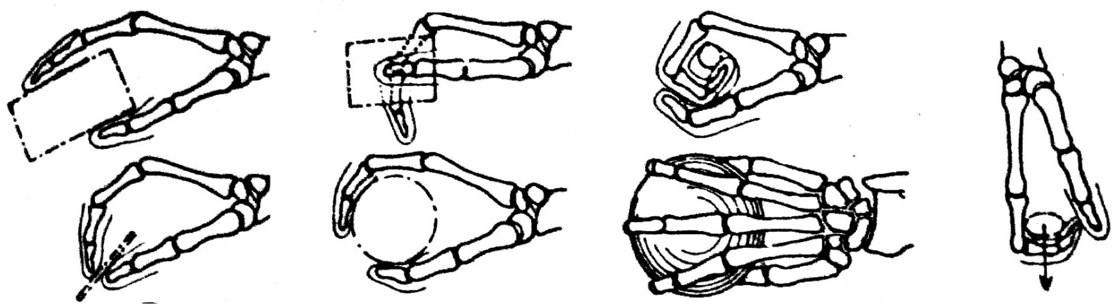

**Figure 2.3:** Schlesinger's list of unique grasps that an artificial hand should be able to conduct. Illustration slightly adapted from (Schlesinger 1919, 503)

Schlesinger (1919) - whose focus of research was the construction of artificial limbs - is generally seen as the first author to present a systematic list of functional grasps[5]. He identified twelve grasps which were often used in everyday tasks. Several of them are very similar. From these grasps Schlesinger selected seven grasps that an artificial hand should be able to conduct (Figure 2.3). He not only consolidated similar grasps but also eliminated grasps which he believed an artificial limb would not need to be able to conduct. This included grasps which could be conducted as well with the other, healthy hand. Another grasp - holding a file - was eliminated from the list of grasps because Schlesinger thought that an artificial hand would not be well-suited for this task. Although Schlesinger's list is sometimes described as a taxonomy of human grasps (Heumer et al. 2007; Saponas et al. 2009; Feix 2011), it is not a taxonomy, as it does not describe a hierarchy or grouping of grasps. It also does not pertain to human grasps but to selected grasps for artificial limbs.

Slocum and Pratt (1946) define three fundamental hand functions: *grasp* (involving the palm), *pinch* (involving the fingertips), and *hook* ("simply what the word implies").

Napier (1956) published the most widely cited scientific classification of grasps by distinguishing between *precision grips* which only involve the finger tips and *power grips*

[5] To my knowledge, my dissertation contains the first correct illustration of Schlesinger's list of grasps. Schlesinger's work was incorrectly summarized by Taylor and Schwarz (1955), who claim to show six basic grasps "as defined by Schlesinger" whereas Schlesinger proposes seven basic grasps (Schlesinger 1919, p.503). Additionally, two of Schlesinger's original grasps cannot be found in the list by Taylor and Schwarz. They also attribute a "tip grasp" to Schlesinger which is not actually in Schlesinger's list. Most authors refer to this summary when citing Schlesinger. Not all authors make clear that they follow Taylor and Schwarz, however.
It can be assumed that nearly nobody who references Schlesinger actually read Schlesinger's original report. Interestingly, MacKenzie and Iberall (1994) have apparently read the report. They describe and discuss several details of Schlesinger's work that are not found in other publications. Nevertheless, they do not mention the discrepancy between Schlesinger's grasp list and the rendition by Taylor and Schwarz.

which involve the palm of the hand. This basic distinction has found wide-spread acceptance[6]. As Böhme argues, Napier did not merely give new names to Slocum and Pratt's grasps. Instead, "Napier focused his classification on the question why certain prehensile actions are chosen instead of only asking which prehensile postures the human hand can adopt" (Böhme 2011, 10)

Napier also presents examples of specific grasps for both classes. Often, a screwdriver is used as an example for a tool that can be held in a tight power grasp for hard work but also in a precision grasp between fingertips for fine manipulation.

However, Landsmeer (1962) argues that the term *precision grip* is misleading. To Landsmeer, a *precision grip* is intended for manipulating the object being held. Therefore, Landsmeer argues, it is not sensible to distinguish between a static precision grip and a dynamic manipulation phase. He proposes *precision handling* as a better term. This re-definition has not been widely accepted, however.

Another approach for classifying grasps is to distinguish between *form closure* and *force closure*. Both Bicchi (1995) and Ponce et al. (1997) note that this distinction has been commonly accepted for some time, without providing a source. *Form closure* grasps are grasps that restrict an object's movement by obstructing all directions in which the object might move. No force has to be exerted at the contact points. An example of a form closure grasp is holding a small marble in the palm of the hand with all fingers closed around it. In contrast, *force closure* grasps depend on friction at the contact points. To effect this friction, the fingers have to exert force. An example of a force closure grasp is holding a small marble between thumb and index finger.

Arbib, Iberall, and Lyons (1985) propose a simplified, functional description of grasps using *virtual fingers*. They argue that force closure grasps can be modeled as just two virtual fingers , VF1 and VF2 which hold an object by exerting opposing forces on it. Sometimes, a third virtual finger, VF3, may be used for stabilizing the grasped object. A virtual finger may be either the thumb, one or more fingers, or the palm. For example, when holding a pen, thumb and index finger act as VF1 and VF2. The middle finger acts as VF3, stabilizing the pen.

Cutkosky and Wright (1986) argue that "when people use objects in everyday tasks, the choice of grasp is dictated less by the size and shape of objects than by the tasks they want to accomplish". Therefore, "grasps should be categorized according to function instead of appearance". They propose a taxonomy that organizes 16 functionally different types of grasps within a tree which is arranged along two axes: *power vs. precision* and *task-oriented vs. object-oriented*.

Cutkosky and Wright concede that their classification does not cover all possible grasp

[6] However, researchers sometimes redefine the meaning of these two categories to better fit their goal. For example, Mortensen and Bærentsen (2013) write: *"A precision grasp is defined as one or more fingers on the remote control buttons. A power grasp is defined as all fingers locked around the remote control."*. While these definitions are obviously inspired by Napier, he is neither mentioned nor referenced in the paper.

types. However, they argue, any other potential grasp can be replaced by a functionally equivalent grasp from their taxonomy. Therefore, they deem their list of grasps as sufficient for discussing new designs for artificial limbs or robotic grippers.

Kang and Ikeuchi (1991) argue that previous grasp classifications do not aim to describe actual grasps but rather tell which grasps might be used for a certain task or object. They present a method for describing arbitrary grasps using a *contact web*. The contact web is a data structure containing a *contact point* for each finger segment that is in contact with the grasped object. For each contact point, the 3D position is recorded. A specific grasp is represented as a set of contact points, the contact web. Building upon this concept, Kang and Ikeuchi also describe a method for recognizing certain grasp types based on their associated contact web. To this end, contact points are algorithmically clustered into virtual fingers. The arrangement of the virtual fingers is then translated into one of the grasp types proposed by Arbib, Iberall, and Lyons (1985). However, the contact web does not describe all aspects of a grasp. Neither size of the contact area nor grasp force are included in the model. In addition, the contact web does not distinguish between the sides of a finger segment. This means that a cigarette held between index and middle finger and a pen lying on the middle segments of both fingers would result in the same contact web. Overall, the contact web is the most precise formal description of grasps published so far[7].

To generate the aforementioned taxonomies and models, scientists merged and abstracted specific grasps that they assumed to be representative of human grasps in general. However, almost none of the existing taxonomies are based on a collection of specific grasps recorded 'in the wild'. For example, Schlesinger simply posits that the 13 specific grasps shown in Figure 309-321 of his work (Schlesinger 1919, 500) 'show the hand in its most important usage postures' ("Gebrauchsstellungen") (Schlesinger 1919, 501). Other researchers asked study participants to grasp predefined objects in laboratory settings (Kamakura et al. 1980). A notable exception are Cutkosky and Wright (1986) who show twelve grasps that they observed being employed by machinists during their work. These grasps are then organized in the presented taxonomy. However, Cutkosky and Wright do not mention whether these 12 grasps were indeed the only grasps they observed, or how prevalent they were.

In summary, there exists a diverse set of tools for describing and classifying grasps. In particular, the comprehensive grasp taxonomy by Feix et al., and the *contact web* model by Kang and Ikeuchi offer a precise language for characterizing grasps[8]. However, taxonomies only tell little about the importance, prevalence, and usage of different grasps.

---

[7] While not focused on grasps but on hand postures in general, the "Natural Human Hand Model" (NHHM) (Nierop et al. 2008) allows for meticulously describing the exact hand posture used for a grasp. The NHHM takes into account the constraints of each joint within a hand and also the hand's skin.

[8] See also the Columbia Grasp Database, `http://grasping.cs.columbia.edu/`, a database containing 3D models of various objects and optimal grasps for each object. The database was generated automatically using a physics simulation.

## 2.4 How Do People Grasp?

In order to design and implement interactive systems that employ grasping as an input modality, it is important to know how people actually grasp, instead of defining grasp types that trigger specific actions. This understanding is necessary for four reasons:

- First, grasp interaction should not accidentally overlap with everyday grasping. For example, simply picking up an object in order to look at it should not trigger any explicit reaction of the object in most cases.
- Second, there might be certain commonly employed grasp types which lend themselves to be used as intuitive triggers for certain actions. For example, holding a mobile phone in a landscape orientation with both hands might be a universally understood action for activating the phone's camera application.
- Third, grasps of a certain object by a single user may vary significantly depending on the user's physical and mental state, external conditions, and time. Training a grasp-classification algorithm using only data from a single training session can result in bad recognition performance in different settings.
- Fourth, grasps might significantly differ between users, even when conducting the same tasks. Designing grasp interaction based on the preferred grasps of a single user could result in grasps that a majority of users do not see as natural.

These issues lead to two basic questions that need to be answered:

a) Which grasps do people employ in everyday situations?
b) How similar are grasps of a single user or of a group of users across multiple interactions with an object?

The implications of these questions for grasp interaction are discussed in more detail in Chapter 9.

For many of the taxonomies presented in the previous chapter, the creators apparently did not conduct surveys but used their own imagination to determine 'common' grasp types. However, other researchers have investigated everyday human grasping and grasp variability in more detail. A small number of field studies have been conducted in order to analyze everyday grasping. In these cases, video recordings of manual actions were annotated with the observed grasp types. For determining how variably persons grasp specific objects, controlled laboratory experiments were conducted where participants' hands can be tracked or photographed more reliably.

This section gives an overview of existing research regarding how different people grasp different objects for different tasks. Later in this dissertation, Chapter 10 presents a study in which we investigated how users grasp mobile phones.

### 2.4.1 Grasping in Everyday Tasks

A common approach for classifying everyday grasps is to record a person doing manual tasks on film or video. This is then annotated by one or more researchers who mark duration and/or type of each grasp recognized in the video[9].

Sperling and Jacobson-Sollerman (1977) conducted a study investigating common grasps while serving food, eating, and drinking. 30 study participants (15 female, 5 left-handed) were filmed eating a "normal meal" - sliced meat, salad, a beverage, dessert, coffee, and cake. The films were manually annotated, resulting in 1277 coded grasps. Participants were instructed grasp each of the 15 different involved objects - including cutlery and porcelain but also e.g., a lump of sugar - at least three times during the "meal". Symmetrical objects were selected to avoid influencing which hand was used. The three most common grasps for each object were documented. Sperling and Jacobson-Sollerman devised their own classification scheme, distinguishing between three general grasp types. A "volar grip" involving the palm, a "finger grip" involving only the fingers, and a "web-of-thumb grip" involving the dorsal area between thumb and index finger. The first two grasp types are similar to Napier's power grasp and precision grasp. Furthermore, contact surface between hand segments and objects were documented.

While 77% of all grasps were conducted with the dominant hand, left-handed participants employed their left hand only in 68% of the grasps they performed. Sperling an Jacobson-Sollerman suggest that "it may be assumed that left-handed persons, through continuous social influence, will have learned to use their right hand to a greater extent than their hand dominance originally allowed".

Thumb and index finger were involved in almost all grasps, the middle finger in 95% of all grasps. The other two fingers were used rarely and mostly to provide additional support.

In an unpublished experiment, Feix and Jaworski (2009) investigated which types of grasps a (single) user employed while performing everyday tasks. These tasks included putting dishes into a dishwasher or using a garden hose. The captured video was manually annotated with information about each visible grasp[10]. In the analyzed video segment of 32 minutes length, 208 grasps were annotated (6-7 unique grasps per minute). Four grasp types accounted for 67% of all grasps: Large Diameter , Prismatic 4 Finger , Lateral , and Extension Type according to Feix' taxonomy. At least for objects weighing less than 1000 g, Feix and Jaworski could not find an obvious effect of object weight on the employed grasp type. However, they found that the *Large Diameter* and *Prismatic 4*

[9] There are also other approaches, as discussed in Chapter 4. For example, Ingram et al. (2008) recorded hand postures in everyday settings using data gloves. In personal communication (e-mail, 27.07.2010), Ingram was skeptical about extracting grasp information from the data. As they did not measure contact forces, he deemed it very difficult to determine whether a certain hand posture belonged to a grasp. Ingram declined to share raw data.

[10] Thomas Feix, personal communication, 29. July 2010.

*Finger* grasps were primarily employed for large objects (diameter > 50 mm) whereas *Lateral* and *Extension Type* grasps were employed for grasping small objects (diameter < 20 mm). Not surprisingly, object shape was found to also affect which grasp type was employed. Feix and Jaworski did not check for combined effects of (shape x size) or (shape x size x weight) on the employed grasp type, however. It seems probable that humans choose a grasp type based on the specific combination of objects' shape, weight, and size.

Bullock et al. (2013) recorded video of 2 housekeepers and 2 machinists performing miscellaneous tasks in their work environments. For each person, about eight hours of video were recorded over the course of several days. Together with the participants, the researchers determined tasks that were representative of the participant's everyday work. Only these were recorded using a head-mounted camera that the participant could control. This allowed them to exclude non-relevant or private actions from the recording. The recorded video was annotated with information about each visible grasp. Grasps were coded using a slightly modified Feix taxonomy. Grasps less than one second in length were not coded. Therefore, only more or less static grasps were coded. Dynamic manipulation involving separate grasps flowing into each other was excluded from the analysis. Bullock et al. report several interesting figures. Each participant conducted about 4700 grasps within the 8 hours of recorded action. This results in about 10 grasps per minute - significantly more than the 6-7 grasps found by Feix and Jaworski[11]. Nearly all grasp types from the Feix taxonomy were found in the recorded video. The 10 most prevalent grasp types made up 80% of all grasps. However, the housekeepers used fewer different grasps than the machinists. The 3 most prevalent grasp types were employed 50% of the time. Participants were grasping 60-90% of the time. Approximately 40% of grasps were power grasps, 40% precision grasps, and 20% intermediate grasps. Bullock et al. also identify several grasps that were primarily used for picking up objects or putting them down. They also document two general limitations of existing grasp taxonomies: These do not contain specific grasps for soft objects, such as towels. The researchers coding the grasps seen in the video often disagreed about the classification of such grasps. In addition, in several cases the grasping person was holding multiple objects in one hand, employing multiple grasps at the same time. Such grasp combinations are being completely ignored in existing taxonomies.

To summarize these studies:

a) people employ a multitude of different grasps during manual tasks,
b) however, a few prevalent grasp types account for the majority of grasps,
c) most grasps are employed only for a short time - even static grasps change every 5-10 seconds during manual tasks.
d) existing taxonomies do not capture important grasp types, such as grasping soft objects or multiple objects at the same time.

[11] The housekeepers and machinists in Bullock's study were professionals, whereas the grasping person in Feix' study was a student doing housework. This might explain the differences in speed.

### 2.4.2 Similarity of Grasps Across Users

In the "meal study" mentioned above, Sperling and Jacobson-Sollerman (1977) did not only classify different grasps employed during serving and eating a meal but also observed large differences between participants. For example, the three prevalent grasps for the knife were used in 48%, 28%, and 23% of all cases. 16% of participants alternated between two grasps for the knife. For the fork, the distribution was 80% - 10% - 10%; 33% of subjects alternated between grasps. For the dessert spoon, the most common grasp was employed in 92% of all cases, however. This means that grasp variability was strongly correlated to the object that was grasped. Overall, the more specific a task was, and the longer it took, the less variation in grasps was observed.

Sperling and Jacobson-Sollerman conclude that "[w]hich fingers an object is held with depends mainly on the shape of the object and the purpose of the grip". However, which grasp is used "also depends on personal habits and cultural factors".

Kamakura et al. (1980) asked seven participants (27-37 years) to grasp 98 different objects which were chosen based on a selection of nouns found in a dictionary. The grasping hand was photographed. In addition, the grasped object was covered in ink, staining the hand's skin at the contact areas between object and grasping hand. These contact areas were photographed, too. Grasps were considered identical if they "showed similarities in both the posture and the contact areas". Kamakura et al. derived 14 grasp types from the photos, organized in four categories: power grip, intermediate grip, precision grip, and a grip involving no thumb. They found that each subject employed all 14 different grasps throughout the study. 31 of the 98 objects were grasped in the same way by all participants. 86% of observed grasps could be categorized as belonging to one of the 14 grasp types, the rest were combined or intermediate grasps.

Kinoshita, Murase, and Bandou (1996) asked 26 study participants with an average age of 21 (female) respectively 24 (male) years to pick up cylindrical objects standing on a table. Participants were required to employ a *Precision Disk* grasp. The cylinders had three different diameters and three different weights. Participants had to use 2-, 3-, 4-, and 5-finger grasps. Kinoshita et al. manually measured finger positions on the cylinders. Grip force was recorded electronically. They found no effect of gender on force or position of fingers, a subtle effect of hand dimension and strength on force of fingers, and no or only subtle effects of cylinder size and weight on finger position. Overall, for such simple tasks, people seem to use very similar grasps.

An unpublished study on cylinder grasping, conducted by Maurice Sanner for his Diploma Thesis at the University of Munich, brought very similar results. However, as his study participants had larger hands than the participants in Kinoshita's study, absolute finger positions were slightly different.

Santello, Flanders, and Soechting (1998) had five participants (30-41 years) imagine to grasp 57 different objects - from "Apple" to "Zipper" - and assume a corresponding hand posture in mid-air. The angles of the finger joints were captured by a Cyber-

Glove worn by all participants. A discriminant analysis was used for quantifying the differences between hand postures. Santello et al. found no clusters of similar grasps, however, and did not find a correlation between expected grasp category (power vs. precision) and hand posture. As participants never actually touched an object, the study effectively documented hand postures shaped by the concept of an object, not by its actual surface properties.

Hinckley et al. (2014) observed how nine participants grasped a digital pen and tablet. Users employed two different grasps for stowing the pen while they interacted with the tablet's touch screen. In some cases, the users tucked the pen between e.g., middle and ring finger, in other cases they stowed it in their palm. Preferences depended on grasp context. Grasps varied significantly between users. For example, one user touched the screen using her ring finger, which seems to have resulted in unique grasps of the pen. Users also constantly switched between different grips. The authors suggest that these *regripping behaviors* are "motivated by comfort, fatigue, and functional considerations" (Hinckley et al. 2014). Users also often chose certain grasps to avoid erroneous touches on the tablet's touch screen.

Overall, these studies suggest that:

a) for certain objects and tasks, grasps by different persons are very similar
b) for other objects and tasks, there is a higher variability in employed grasps

### 2.4.3 Similarity of Grasps by a Single User

In conjunction with the meal study, a second study with the same group of participants was conducted (Jacobson-Sollerman and Sperling 1977). This study used the same test setup as the *Rancho Los Amigos Test* which is used for diagnosing limitations of human hand functions. The participants had to pick up and move different abstract objects lying on two shelves of different height. The 14 objects included cubes, balls, slabs, and tubes of different sizes. Participants had to pick up each of the objects once with the left and the right hand, resulting in 818 classified grasps. Jacobson-Sollerman and Sperling found that all cubes were always picked up using precision grasps. The smaller the cube, the less fingers were used. The layout of the objects on the shelves was not symmetrical, and objects sometimes were so close to each other that they obstructed the grasp. This resulted in a high degree of correlation between left/right hand and used grasp, i.e. participants employed significantly different grasps depending from which side the hand had to approach the object. As in the previous study, thumbs and index fingers were used in almost all grasps. The middle finger - which was used in 95% of all grasp in the meal study - was used only in 79% of all grasps in this study. Ring and little finger were used more often than in the meal study. Jacobson-Sollerman and Sperling note that the grasps used by participants of the study were quite different from the grasps that are defined in the Rancho Los Amigos Test for each object. Comparing this study to the meal study, they note that tests with abstract objects are "not directly applicable to integrated activities of daily living".

Within the *Secure Grip* project, Veldhuis et al. found evidence that people grasp a gun more consistently if they had often handled this type of gun previously. The most experienced participants conducted the most consistent grasps. However, while experienced participants employed very consistent grasps within a test session, grasps varied dramatically between sessions. Details are discussed in Section 4.4 alongside other findings from this project.

Overall, these studies suggest that:

a) the grasp employed is not only determined by task and object size, but also by the surrounding environment
b) the same person may grasp the same object for the same task in different ways over time
c) the more often a person grasps the same object for the same tasks, the more similar these grasps will be

### 2.4.4 Discussion

Looking at the combined results of all studies presented above, several issues become obvious:

In most cases, researchers manually annotated film or video recordings of people grasping objects. This is both a tedious process and leads to compulsive classifications, i.e. researchers assigning a certain grasp type to a specific grasp even though both barely match because of a lack of better alternatives. The main problem here is that existing grasp taxonomies do not cover many of the grasps actually occurring *in the wild*. While taxonomies aim to group grasps into clearly delimited grasp types, in reality specific grasps are extremely diverse and may not be put into mutually exclusive groups. Several researchers try to fit the observed grasps into these taxonomies but many report difficulties in deciding which grasp types should be assigned to specific grasps. None of the existing taxonomies supports classification of grasps which are a combination of other grasps or intermediate forms. Furthermore, for none of the presented studies do the researchers document how they determine the similarity of a grasp to each of the grasp types defined in the taxonomy they use.

While grasp taxonomies have several purposes, even the comprehensive taxonomy by Feix is very incomplete. Instead of trying to fit observed grasps into seemingly arbitrary taxonomies, it might be more worthwhile to instead group grasps by features that can be directly measured, as demonstrated for example by Kinoshita, Murase, and Bandou (1996) and Santello, Flanders, and Soechting (1998). To this end, unbiased, automatic recording of important features of a grasp is necessary.

Most of the studies presented in this chapter were conducted with only few participants. It seems that the more participants take part in a study, the more diverse the observed grasps are. As the studies investigate very diverse aspects of grasping, and as none of the studies has been replicated so far, their results are certainly not conclusive. Nevertheless, taken together these studies show a very large variety in grasps. These findings are also confirmed by our study on grasps for mobile phones - presented later in Chapter 10.

Many publications state that a specific grasp is determined not only by the shape of the object to be grasped, but also by the task that is to be performed. Only Jacobson-Sollerman and Sperling (1977) also mention that the surrounding environment may have an effect on how an object is grasped. Interestingly, none of the publications explicitly mentions that the anatomical features of the grasping persons might affect how they grasp an object. Only Kinoshita, Murase, and Bandou (1996) report measuring hand size in their study.

## 2.5 Summary

In this chapter I have given an overview of research on human grasping. Since about a century, scientists have observed how human hands grasp objects. While early research was driven by rehabilitation medicine and prosthetics, other disciplines such as neuroscience and robotics have also significantly contributed to our understanding of human grasping since then.

Several definitions, classifications, and taxonomies of grasping exist, shaped by the specific needs of different fields. Multiple studies show that people employ a wide variety of different grasps that evades simple classification schemes, and that object properties, task, and surrounding determine the specific grasp that is used.

Chapter 3

# Grasp Sensing:

## Definitions and Workflow

*Interpreting a user's grasp as input for human-computer interaction has only recently received broader attention. This chapter gives a concise overview of grasp sensing for human computer interaction. It provides the theoretical background for the following chapter 'Grasp Sensing Techniques' that discusses specific implementations. As none of the previous definitions for 'grasping' accurately describes human grasping, I propose a new definition that is unambiguous and matches intuitive understanding. Grasping is "the act of bringing body parts - usually parts of the hand - into physical contact with an object, thereby weakly or strongly binding the object's movement to the movement of the body parts in at least two opposite directions through friction or obstruction of the object's movement, independent of gravity or other forces acting on the object". To ease implementation, it is helpful to capture and process only static grasps, not dynamic manipulation. This limitation is necessary for practical reasons and may be acceptable because dynamic manipulation can be modeled as a series of static grasps. In order to reduce ambiguity, I also propose definitions for grasp interaction, grasp sensing, grasp information, and related terms. Grasp sensing can be divided into three parts: capturing a grasp signature using a grasp-sensitive surface or other sensors, identifying the grasp that is employed, e.g. by means of machine-learning classifiers, and finally interpreting the meaning of the grasp, i.e. determining an appropriate reaction to the grasp.*

**Attribution:** This chapter is based in part on concepts presented in my papers "HandSense: Discriminating Different Ways of Grasping and Holding a Tangible User Interface" (Wimmer and Boring 2009) and "Grasp Sensing for Human-Computer Interaction" (Wimmer 2011a) but has been significantly expanded.

## 3.1 Introduction

While human grasping has been subject of intensive research for over a century, only recently has grasp sensing and its application to human-computer interaction been explored.

In Section 1.2 I have argued that sensing how users grasp objects opens a yet mostly unexplored input channel for interacting with physical and digital artifacts.

While understanding human grasping is essential for *designing* grasp interaction, reliably capturing grasps is essential for *implementing* grasp interaction. Furthermore, as mentioned in the previous chapter, grasp sensing also allows us to investigate grasping more efficiently and effectively.

This chapter provides the conceptual background for all following chapters. First, I explain the difference between static and dynamic grasps and argue that focusing on static grasps is acceptable. Then, I propose a new, interaction-centric definition of grasping. Afterwards, I present further definitions for various terms related to grasp interaction. Finally, I describe a generic *Grasp sensing workflow*, consisting of three steps: capture, identification, and interpretation.

## 3.2 Limiting Grasp Sensing to Static Grasps

As discussed in Section 2.2.2, related work distinguishes between three phases of grasping:

- reaching phase,
- grasping phase,
- manipulation phase.

However, Landsmeer (1962) points out that grasping and manipulation are inherently coupled. The hand often seamlessly transitions between multiple different grasps, e.g., when picking up a mobile phone and putting it into a pocket. Therefore, ideally grasp sensing would capture and analyze the complete chain of motions involved in a prehensile movement.

However, for practical reasons, I focus on static, discrete grasps in this dissertation. This explicit limitation is valid for three reasons:

- **Lack of basic research on dynamic grasps.** As described in the previous and the following chapter, most research on grasping focuses on discrete, static grasps.

Therefore, little is known about variability and other properties of transitions between grasps. While grasp sensing may be able contribute to a better understanding of these properties, currently there is not enough reliable knowledge of grasp transitions to base grasp sensing or grasp interaction on it.

- **Adequate Effectiveness.** Prehensile manipulations of grasped objects can be discretized into a series of stable grasps. As the transitional hand movements between individual grasps depend on the previous and the following grasp, they contain only very little additional information. Therefore, incorporating grasp transitions into grasp interaction would probably not greatly enhance recognition or interaction. On the contrary, grasp transitions are probably highly variable and not wilfully chosen. Inferring information about the user's grasp or intentions from them seems difficult.
- **Higher Efficiency.** Multidimensional time-series data - such as a chain of hand movements - is much harder to analyze and classify than discrete snapshots. Limiting oneself to discrete, static grasps, significantly simplifies implementation of grasp-sensitive artifacts.

Of course, many stable grasps allow the user to move individual fingers and perform gestures with them (Wolf et al. 2012). These non-grasping fingers are not part of the grasp. However, such micro-interactions may be facilitated by certain grasps or require them. Given the high complexity of gestural interaction, this topic is out of the scope of this thesis.

In summary, it seems necessary, sufficient, and beneficial to focus on static grasps for grasp sensing and grasp interaction.

# 3.3 An Interaction-Centric Definition of Grasping

The definitions for "grasping" presented in Section 2.2 describe various aspects of grasps. However, they describe grasping at a very low level, focusing on hand anatomy, object shapes, and force equilibria.

The research presented in this thesis investigates how humans might interact with objects by grasping them (see e.g., Figure 3.1). Therefore, it is helpful to consider not only these functional aspects but look at grasping from the user's perspective. For a user, grasping an object is effectively just a way to control or feel it. A definition that excludes certain actions that users would describe as *grasping* may unnecessarily limit the design space for grasp interaction. Therefore, an intuitive definition of human grasping should be broad, embracing all hand movements that people might call *grasping*.

Existing definitions (except for the one by MacKenzie and Iberall (1994)) do not cover various hand configurations that might be called grasps, such as grasping a pen with one finger (implicitly: Arbib, Iberall, and Lyons (1985)), holding a ball with both hands (Feix et al. 2009b), or pressing a phone to the ear with only one finger (Böhme 2011). Usually such grasps are silently ignored without giving reasons for their exclusion. The definition of prehension by MacKenzie and Iberall (1994) is quite abstract and overly broad. According to their definition, prehension also includes touch-typing, punching someone in the face, or snipping a marble. Additionally, all of the aforementioned definitions focus on grasps as means of controlling an object, ignoring that we also grasp objects in order to feel their properties - such as temperature or vibration - even if we do not want to interact with them. For example, grasping an object also facilitates haptic feedback (Fukumoto and Sugimura 2001).

Therefore, I propose a new definition[1] that is more inclusive and more congruent with everyday meaning of the term "grasping".

**Grasping** is:

(a) the act of bringing
(b) body parts - usually parts of the hand -
(c) into physical contact with an object,
(d) thereby weakly or strongly binding the object's movement to the movement of the body parts in at least two opposite directions
(e) through friction or obstruction of the object's movement,
(f) independent of gravity or other forces acting on the object.

A **grasp** is accordingly defined as:

(a) the state of
(b) body parts - usually parts of the hand -
(c) being in physical contact with an object,
(d) thereby weakly or strongly binding the object's movement to the movement of the body parts in at least two opposite directions
(e) through friction or obstruction of the object's movement,
(f) independent of gravity or other forces acting on the object.

This definition defines grasps based on contact between the hand (or other body parts) and the object. It will be used in the remainder of this dissertation. While the hand is still the focus of this definition, other body parts may be used to support or replace the hand. My definition addresses the previously mentioned limitations of some existing definitions:

[1] As argued by Munroe (2011), introducing yet another definition poses dangers. However, my definition pertains to a novel perspective on grasping, and is therefore not competing with but augmenting other definitions.

- It is agnostic with regard to hand anatomy, object shapes, or exerted forces. This allows it to encompass a variety of grasps that might not fall within traditional definitions of *grasps* but which are generally accepted as grasps[2]. For example, my definition also includes fixating a folded newspaper between upper arm and torso.
- The definition intentionally does not require a specific motivation (e.g., manipulation, taking control of, etc.) for a grasp. A *grasp* is purely defined by its observable properties. This *duck typing* approach[3] reduces ambiguity.
- It is made clear that *grasping* is a process that leads to a *grasp*. Therefore, reaching movements are clearly a part of *grasping* but not of a *grasp*.
- The *hook grasp* - which is not covered by most of the aforementioned definitions - is clearly covered by this definition (Figure 3.1b).
- By requiring that a grasp binds an object's movement to the movement of the hand in *at least two* opposite directions, the definition excludes several non-grasp actions, such as pressing a button. While the button can be pressed by the finger, it cannot be pulled by moving the finger in the opposite direction (Figure 3.1d).
- By requiring that a grasp needs to control an object independent of the direction of gravity, the definition excludes obvious non-grasps, such as a brick lying on the back of the hand.

## 3.4 Definitions

As presented in this and the following chapter, human grasping has been investigated by researchers from several different fields - such as prosthetics, robotics, neuroscience, computer science, and human-computer interaction. These independent investigations have both widened and deepened our understanding of human grasping. However, this also caused researchers from different fields to use different names for the same concepts or the same name for different concepts.

For example, the terms *grasp classification*, *grasp recognition*, and *grasp sensing* are ill-defined and may describe the same concept for some researchers and completely different concepts for other researchers.

Ekvall and Kragic (2005) provide an example of interchanging use of terms within one paper. Under the heading "Grasp Recognition: Three Methods", they "present three methods for grasp classification".

In addition, many common concepts have been used implicitly but have not been defined formally. This makes it hard to discuss and compare them.

[2] In order to determine how clear and precise my definition is in practice, one might conduct a poll. By asking participants which of a set of body movements they would classify as "grasping", congruency of my definition with users' mental models of grasping could be quantified. Such a study has not yet been conducted, however.

[3] Wikipedia: `http://en.wikipedia.org/wiki/Duck_typing`

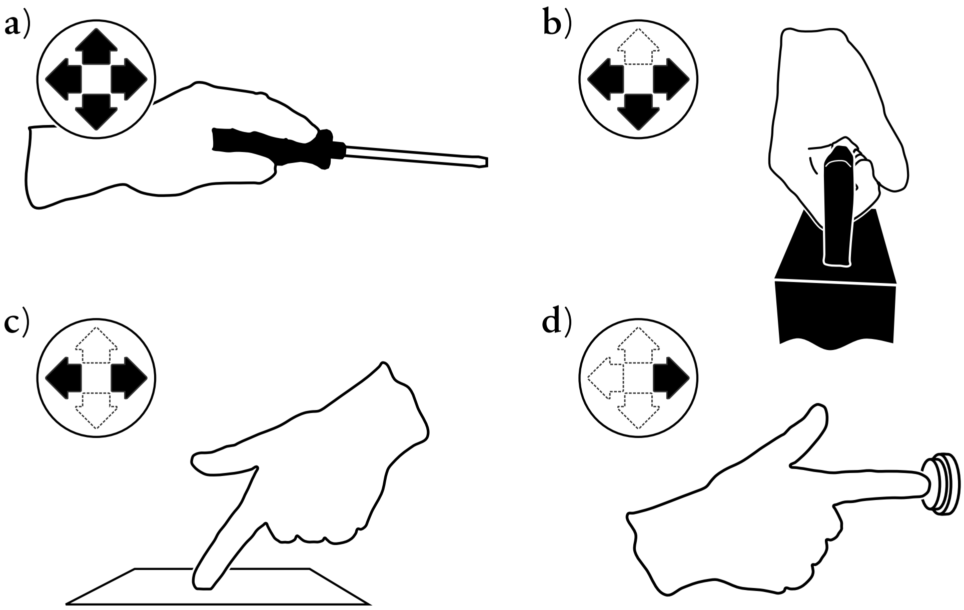


**Figure 3.1:** "Grasping" and "touching" are generally considered semantically different actions. A definition of "grasping" should therefore not include simple touches. The definition of "grasping" proposed in this thesis requires an object to be coupled to the movement of the hand (or other body parts) in at least two opposite directions. This includes, for example, (a) holding a screwdriver in a classic power grasp, (b) grasping the handle of a suitcase in a hook grasp, or (c) sliding an object across a table with one finger. Not included in this definition is, e.g., (d) pushing a button, as the button moves only in one direction.

In order to thoroughly consolidate and compare research from different fields, it is helpful to use a common, consistent language.

Therefore I propose a hierarchy of definitions for *grasp sensing* and related terms. I tried to avoid redefining terms that have been used in previous publications. Thus, I introduce a few new, virgin terms. Some of these have been previously defined in Wimmer (2011a). These definitions are refined and extended in the following.

A **grasp signature** is the (digital) representation of a specific grasp, as recorded by one or more sensors. It consists, for example, of contact points or digit positions associated with a grasp, including information such as force, temperature or finger texture at each contact point. It does not necessarily completely describe a grasp, but only the effect of a grasp as visible to the sensors.

In robotics, Mason et al. (2012) define *grasp signature* quite similarly as "the time history of the entire grasp process as perceived by the hand's own sensors". This definition includes temporal aspects of a grasp - which I explicitly exclude - but only pertains to the sensors in the robotic hand.

**Grasp information** is any information that can be used to partially or completely describe a concrete grasp. This includes the sensor readings combined in the *grasp signature*, current sensor readings that are not caused by the grasp (e.g. accelerometer data describing the object's orientation), and existing background knowledge (e.g. about the user who is interacting with the object). Each individual sensor reading in a grasp signature is a piece of grasp information.

A **grasp-sensitive surface** is the surface of a graspable object that is able to capture *grasp signatures* using one or more embedded touch sensors. Not all areas of a grasp-sensitive surface need to be touch-sensitive, however. For example, the handle of a power drill is also an instance of a grasp-sensitive surface, as changes in grasp force of the index finger are sensed by the mechanical pushbutton and used to trigger rotary motion of the drill.

The term "grasp-sensitive" was first used on a web page describing the Bar of Soap prototype[4] but is not to be found in Taylor's Master's Thesis or the CHI paper (Taylor and Bove 2009). I preferred this counterpart to the established term "touch-sensitive" over alternatives and used it first in the title of my FlyEye paper (Wimmer 2010a). The term has since been adopted for example by (Sato, Poupyrev, and Harrison 2012).

**Grasp sensing** is the process of capturing *grasp information*, usually a *grasp signature*, (partially) identifying the specific grasp or type of grasp that corresponds to the captured information, and interpreting its meaning, using additional context information.

I prefer the term *grasp sensing* over alternatives, such as *grasp recognition* or *grasp classification*. As mentioned above, these terms have previously been used in different contexts and with changing meaning. In addition, *recognition* and *classification* might also be seen as pertaining only to the act of comparing a captured grasp signature to a pre-recorded signature. Furthermore, the term *grasp classification* may refer to an input, the process, or the result of grasp sensing. Therefore, *grasp sensing* seems more precise than the alternative terms found in related work.

**Grasp interaction** is the concept of utilizing grasp information for facilitating, enhancing, or enriching explicit and implicit interaction between a user and graspable objects.

In the remainder of this thesis, I employ the terms defined above when describing own research and related work, even if the authors used different terms in their publications.

## 3.5 Grasp Sensing Workflow

The technical foundation of grasp interaction is grasp sensing. As defined above, grasp sensing refers to the act of digitizing and interpreting human grasps. Grasp sensing can not only be done to support grasp interaction, but is also used in robotics, neuroscience, rehabilitation medicine, and other fields.

[4] see `http://web.media.mit.edu/~jeevan/pages/bos.html`.

The workflow presented in the following is implicitly used in all grasp-sensitive prototypes presented in the next chapter. It does not describe a new approach but documents and formalizes existing practice.

As shown in Figure 3.2, grasp sensing comprises three independent steps with clear interfaces (Wimmer 2011a):

- **capture** a *grasp signature* by combining available *grasp information*
- **identify** the *specific grasp* or *grasp type* based on the *grasp signature*
- **interpret** the *meaning* of the *grasp (type)*, depending on the *context* of the grasp.

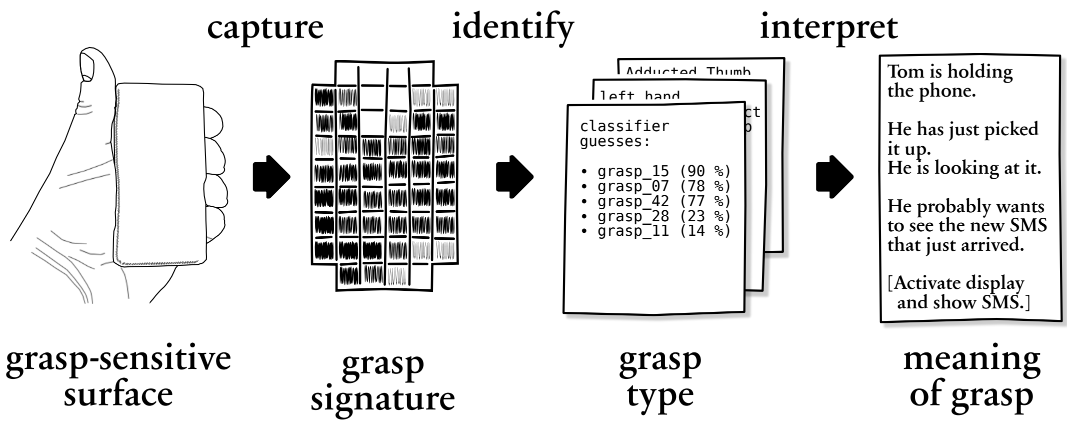


**Figure 3.2:** An exemplary workflow for grasp sensing: A grasp-sensitive surface *captures* a grasp signature. This signature is then used - together with additional information - to *identify* the specific grasp or grasp type. Finally, the meaning of this grasp (type) is *interpreted* using further available context information. The meaningful information generated by such a grasp-sensing system can then be used in grasp interaction.

### 3.5.1 Capturing Grasps

The first step in grasp sensing is to capture[5] information about how the user grasps the object. This can be done in various ways. For example, the grasping person or an observer might write down the positions of finger tips, joints, and palm relative to the grasped object. In practical applications, capturing is done automatically by means of sensors. These sensors can be installed in the environment, attached to the grasping

[5] Both in my original paper and in this thesis I prefer the term *capturing* over *sensing* for three reasons: *grasp sensing* already has another meaning, therefore it would be confusing to call an individual step the same as the whole process; furthermore, the term *capturing* is also used for getting raw pixel data from a video screen, a process which is quite similar to *grasp capturing*; finally, *capturing* encompasses more than just reading sensor data - e.g., collecting context information.

person, or embedded into the grasped object. Sensing techniques suitable for grasp sensing are discussed in great detail in Chapter 4. In the grasp capturing step, available sensors are read out. Optionally, some or all of the sensor data might get preprocessed, e.g. using low-pass filters. The output of this step is a *grasp signature*, a combination of all sensor data pertaining to this grasp. For example, the grasp signature may contain:

- a bitmap of pressure/shear/temperature/etc. readings from a grasp-sensitive surface
- photos of the grasping hand, as captured by cameras installed above the scene
- locations of individual fingers, as noted by an experimenter
- angles of finger joints, as measured by sensors embedded into a glove
- information about previously analyzed grasps
- information about the grasped object, e.g. its shape and orientation
- information about the context of the grasp, e.g., the identity of the grasping person

### 3.5.2 Identifying Grasps

The captured *grasp signature* is then used to identify the grasp that has been employed. Depending on the application, either a *specific grasp* or a *grasp type* is identified. In many cases it may not be possible to exactly determine the grasp. Therefore, the identification step may also output a set of grasps that might have resulted in the recorded *grasp signature*. Each of these grasps may also have a probability associated. This approach is regularly called “classification”. Instead of returning the names of discrete grasps, the identification step may also represent a grasp by one or more continuous values. For example, a grasp might be characterized by the number of involved fingers and the angle between index finger and thumb.

Identifying a grasp is generally done using heuristics or machine-learning approaches.

With heuristics, a set of rules needs to be established that define how to weight and interpret the sensor data in the *grasp signature*. For example, for distinguishing power from precision grasps, a system might calculate the size of the contact area between hand and object, using data from the grasp signature. If the size exceeds an experimentally determined threshold, a power grasp is identified. Otherwise, a precision grasp is identified.

With machine learning, a *classifier* algorithm is trained with numerous grasp signature for each grasp that should be recognized. The classifier automatically determines weightings of the individual features within the grasp signature that allow it to classify new grasp signatures as belonging to one of the known grasps.

While machine learning often offers more robust classification than simple heuristics, it requires the developer to supply sample grasp signatures for all grasp types which should be identified.

Additionally, the quality of machine learning classification hugely depends on the features that are used for training and classification. For example, common classifiers are not able to correctly classify grasps that have been slightly rotated with regard to grasps from the training session. Therefore, sensor data needs to be transformed into a rotationally invariant representation for training and classification (Pai et al. 2005).

The selection of features and necessary transformations are determined by the number and type of grasps to be identified. For example, if the system should distinguish between power and precision grasp, the size of the whole contact area would be an expressive feature. On the other hand, the size of the contact area would not be suitable for determining handedness, as left-handed and right-handed grasps often have similar, mirrored contact areas.

Determining the 'quality' of a biometric classification algorithm or system is not straightforward. The performance of such a system depends on the training data, the test data, and parameters to the algorithm. In general, a binary classification algorithm can make two kinds of errors:

- *Type I errors (false positives)* occur when the system accepts a sample that should be rejected (e.g., allows an unauthorized person to enter the building)
- *Type II errors (false negatives)* occur when the system rejects a sample that should be accepted (e.g., prohibiting an authorized person to enter the building).

For given dataset, three measures are commonly reported for such systems: The *False Acceptance Rate (FAR)* describes the percentage of Type I errors, i.e., test cases that are incorrectly classified as matching (false positives). The *False Rejection Rate (FRR)* describes the percentage of Type II errors, i.e., test cases that are incorrectly classified as non-matching (false negatives).

Both measures depend on the choice of parameters for the matching algorithm. Therefore, they are often plotted against a parameter that determines the 'strictness' of the algorithm. FAR and FRR are negatively correlated. A strict matching algorithm results in a low FAR and a high FRR, while a relaxed matching algorithm results in a high FAR and a low FRR.

Therefore, when reporting the FAR of a matching algorithm, one also has to report the corresponding FRR for the given set of parameters. As FAR and FRR are negatively correlated and between 0.0 and 1.0, there is always a set of parameters where FAR = FRR. This point where the same number of false positives and false negatives occur is called *Equal Error Rate (EER)*.

When there are more than two classes (e.g., a given grasp pattern may be caused by $n$ different grasps), the classifier's results are often presented in a confusion matrix. Approaches for identifying grasps are discussed in Chapter 4 alongside sensing techniques.

### 3.5.3 Interpreting Grasps

Finally, a grasp sensing system needs to determine the *meaning* of a recognized grasp. *Meaning* in this case means the answer to the question "What caused this specific grasp to be employed?". The user's goal is a major part of the meaning of a grasp. Most current approaches to grasp sensing concentrate on inferring the intention of the user from the grasp. For example, users might commonly employ a certain grasp type when they want to take a photo with their mobile phone. Therefore, this grasp type indicates their intention to take a photo. When identifying it, a grasp-sensitive mobile phone might automatically start the camera application. Most existing grasp-sensing systems generally use hard-coded associations between certain grasp types and assumed intentions of the user (see Chapter 4). However, it might also be possible to build systems that automatically learn which actions to associate with certain grasps. In addition, some grasp-sensing systems do not try to infer the user's intention but the users' identity (Veldhuis et al. 2004) or other information. The *interpretation* step has gotten only little attention by researchers so far. I discuss this challenge in Part III.

### 3.5.4 Discussion

The workflow described above represents an archetypical model of grasp sensing. While the concept seems straightforward, the distinction between capturing, identifying, and interpreting a grasp has not yet been acknowledged explicitly or implicitly in previous publications. Clearly defining boundaries between components of a grasp sensing system may be useful in at least three regards:

**Defining Responsibilities.** Sensor design (*capture*), development of machine-learning classifiers (*identify*), and interaction design (*interpret*) require significantly different skills. Explicitly acknowledging these boundaries allows researchers and implementors to focus on one of these areas and eases collaboration.

**Defining Architectures.** As each of the steps is mostly independent of each other, the model may also be used for defining abstract and concrete interfaces between them. Well-defined interfaces would allow application developers to independently select sensors, classifiers, and interpreters depending on their use case.

**Comparing Systems.** As the review of existing research on grasp sensing in Chapter 4 shows, the proposed workflow describes existing research very well. Therefore, distinguishing these three steps helps in structuring and discussing research on grasp sensing.

Chapter 4

# Grasp Sensing Techniques

*There are three general approaches for placing grasp sensors: instrumenting the user, instrumenting the environment, and instrumenting the objects to be grasped. In this chapter, the current research in these three areas is presented, and strengths and weaknesses of the approaches are discussed. I argue that instrumenting objects with grasp-sensitive surfaces is better suited for interactive artifacts than the other two approaches. However, existing solutions for grasp-sensitive surfaces are not suitable for rapid prototyping and further research. This chapter presents the related work for the following chapters.*

**Attribution:** This chapter is based in part on my papers "Grasp Sensing for Human-Computer Interaction" (Wimmer 2011a) and "Grasp Interaction Using Physiological Sensor Data" (Wimmer 2011b). It has been significantly expanded, however. Some figures have been taken or adapted from these papers. I have asked T. Scott Saponas, Stephen Mascaro, and Raymond Veldhuis - whose research is featured prominently in this section - to verify whether I correctly described their research. Minor changes they suggested are incorporated in this chapter.

## 4.1 Introduction

Capturing grasp signatures - information about the grasp(s) employed when interacting with an object - is the first challenge any system for grasp interaction needs to solve. The type, quality and quantity of information in a grasp signature depends on the technique used for capturing it. Grasp sensing techniques can be grouped into three categories, depending on where the grasp sensors are located: in the *environment*, on the user's *body*, or within the *object* to be grasped. This distinction is certainly not the only way to group grasp sensing techniques. One might also group techniques by sensor type (optical, capacitive, resistive, inductive, etc.) or expressiveness (a potentially hard to define measure of how much information about a grasp is captured by the sensor). However, grouping sensing techniques by their location seems most helpful for discussing their application areas. The choice of sensing technique determines reliability, expressiveness, and usability of a grasp-sensing system. Each of the three sensor locations - environment, body, and object - offers unique strengths and challenges, as described in the following sections and discussed in the final section of this chapter.

## 4.2 Instrumenting the Environment

Setting up an external tracking system for capturing grasps can be both the easiest and most precise approach for capturing grasps.

External tracking systems are systems that require a semi-permanently installed tracking infrastructure surrounding hand and grasped object. They can be divided into marker-based and marker-less tracking systems. Most such systems do not allow for capturing grasp intrinsics besides hand posture. However, this limitation is acceptable for many applications. Furthermore, some approaches also allow capturing additional grasp intrinsics, as detailed below.

Common marker-based tracking systems employ optical markers or electromagnetic sensors which are affixed to the user's finger joints. For example, OptiTrack[1] and ART[2] systems employ small reflective globes mounted to the objects to be tracked. Multiple synchronized cameras, each equipped with an infrared light emitter, track the reflections from the globes. The tracking system then calculates the 3D position of each globe using these images. It needs to be calibrated after initial setup, so that the relative positions of all cameras are known. Additional calibration and/or post-processing is necessary for distinguishing between the (very similarly looking) reflections from multiple passive markers. Active markers, employing pulsed LEDs, allow omitting some of the calibration steps and offer more robust tracking but are usually larger and more expensive than passive markers.[3]

Wang and Popović (2009) developed a fast recognition system for hand configurations that uses only a single camera and a glove with a distinctive color pattern on it. Depending on the hand configuration, only a subset of the colored areas on the glove are visible to the camera. The color patterns for all hand configurations that shall be recognized are captured in a training pass. During operation, the system compares the current camera image to this database of color patterns and retrieves the recorded hand configuration

---

1 see `http://www.optitrack.com`

2 see `http://www.ar-tracking.com`

3 In 2008 I supervised the project thesis of Iliana Dimitrova at ART, who investigated ways to semi-automatically map marker reflections to finger joints, based on their unique movement trajectories during a short calibration phase. The thesis title was "Passives Fingertracking und Gestenerkennung".

that is most similar to the camera image. This approach can be seen as a hybrid between marker-based and marker-less tracking. It allows only differentiating coarse hand configurations, not tracking of individual fingers or contact points. Additionally, the system would need to be trained to recognize new grasp types and objects.

Electromagnetic tracking systems, manufactured e.g. by Polhemus, consist of an electromagnetic field generator and multiple sensors that are connected to a measuring device via cables. The field generator generates an electromagnetic field consisting of orthogonally aligned fields for the X, Y, Z axes. These fields are either time-multiplexed (pulsed DC) or frequency-multiplexed (AC). Small, orthogonal coils in each sensor capture orientation and strength of the electric field. From this information, the measuring device calculates position and orientation of each sensor with regard to the field generator (Polhemus 2006). Electromagnetic tracking systems offer high spatial and temporal resolution and low latency. Occlusion is no problem. However, conductive objects nearby can affect the tracking accuracy. A major limitation is that each sensor needs to be connected to the measuring device by a cable. Sensors and cables do not necessarily hamper or otherwise affect grasping as they are usually attached to the back of the hand. However, capturing grasping movements in a confined space, such as a pocket, is either impossible or affected by the need to protect the sensors from being ripped off.

Marker-less tracking systems do not require markers to be attached to an object but track distinctive optical features of the object. Generally, marker-less systems consist of several cameras capturing the same scene from different angles, or one or more 3D cameras using stereo vision[4] and/or structured light.[5]

As part of the GRASP project, Oikonomidis et al. have demonstrated reliable grasp capturing using the Microsoft Kinect (Iason Oikonomidis, Kyriazis, and Argyros 2011) and a multi-camera setup (Iasonas Oikonomidis, Kyriazis, and Argyros 2011). Marker-less tracking systems require significantly more image processing and calculation of potential hand configurations than marker-based systems. Therefore, they are rarely used in actual grasp research.

Kang and Ikeuchi (1994) report on an experiment where hand posture and object position were tracked using a combination of magnetic tracking, an instrumented glove, and optical range finders. This allowed them to identify different grasp phases and the grasps involved. Despite using a sophisticated multi-pass recognizer, the more complex one of two tasks in the experiment resulted in a slightly incorrect classification.

The aforementioned tracking systems do not allow for capturing grasp intrinsics besides hand configuration.

However, cameras can also be used for capturing grasp force, as shown by Y. Sun, Hollerbach, and Mascaro (2006a) and Marshall et al. (2008).

[4] e.g., the Point Grey *Bumblebee* series, `http://www.ptgrey.com/products/stereo.asp`.

[5] e.g., the Microsoft *Kinect*, `http://en.wikipedia.org/wiki/Kinect`, or the *Leap Motion*, `http://www.leapmotion.com/`.

With increasing grasp force, blood is pressed out of the fingertip's vessels. This changes the coloration across the nail bed: The lower half of the tissue which is visible under the fingernail turns redder. At the same time, the tip turns whiter.
Sun et al. employ the approach by Mascaro and Asada (2001), described in the following section, but use a camera mounted above the finger to capture images of the fingernail and surrounding skin. This system is able to capture perpendicular force and shear forces of a single finger. In the presented setup it is not suited for real-life settings, as the finger needs to be placed directly under the camera. Sun et al. published several other papers on this topic (Y. Sun, Hollerbach, and Mascaro 2006b; Sun, Hollerbach, and Mascaro 2007; Sun, Hollerbach, and Mascaro 2008; Sun, Hollerbach, and Mascaro 2009).

Marshall et al. (2008) determine the location of the fingertips within the camera image by applying standard computer vision approaches. Their algorithm then determines the variance in coloration (hue) for each fingertip. This approach is more resilient to lighting changes than using absolute brightness or hue changes. The more force a user applies with her or his finger, the greater the variance of the fingertip's coloration. As it takes some time for the blood vessels to fill again, there is a delay of about 100 ms after each press until the fingertip has regained its original color. The presented approach can only distinguish between two states - 'not pressing' and 'pressing' - and can not be employed if the user is wearing nail varnish or gloves. It is also susceptible to occlusion[6].

In summary, external tracking allows for robust, flexible, and relatively cheap capturing of hand configurations, and therefore of grasps. Both optical and electromagnetic tracking allows capturing the position of every finger joint and digit. However, the dependency on external infrastructure limits their usefulness for real-world applications. Additionally, optical tracking methods are subject to occlusion - which is often inevitable when the tracked hand grasps an object. Finally, external tracking allow only for extremely limited capture of grasp intrinsics such as grasp force or contact areas.

For these reasons, external tracking is used primarily in areas where high fidelity and flexibility are more important than usability, such as neurophysiology research, robotics research, and for prototyping novel interaction techniques and interactive systems.

A comparison of environmental grasp sensing to body-worn grasp sensing and grasp-sensitive surfaces can be found in Section 4.5.

## 4.3 Instrumenting Users

Instrumenting the user with grasp-sensing hardware has advantages over instrumenting the environment. As all sensors are attached to the user, grasps can be captured

[6] Incidentally, it seems that Marshall et al. were not aware of the research published several years earlier by Mascaro and Sun. As evidenced throughout this dissertation, there seems to be very little mutual awareness and exchange between researchers in robotics, neuroscience, prosthetics, and human-computer interaction.

in real-life interactions, e.g. at home, in the workplace, in mobile settings, or in shops. Thus, user-worn grasp sensors allow both for collecting realistic usage data, and for actually enhancing real-life interaction. Additionally, these sensors can capture different types of grasp information, as detailed below, allowing for richer grasp interaction.

An obvious approach is to attach one of the aforementioned external tracking systems to the user. However, optical tracking systems are prone to occlusion and provide only a small tracking volume (Mistry and Maes 2009). Electromagnetic tracking systems require the user to carry a field generator and wear sensors on all finger joints. While such systems would probably not be accepted for everyday use by most people, they can be employed for collecting grasp data in real-life settings. For example, as mentioned in Section 2.4, Feix conducted an unpublished experiment using video annotation in 2009. He investigated which types of grasps a (single) user employed while performing everyday tasks. These tasks included putting dishes into a dishwasher or using a garden hose while wearing a head-mounted camera. The captured video was manually annotated with information about each visible grasp[7].

As I described previously (Wimmer 2011b), there are also several body-worn sensor technologies that capture physiological data which can be used for grasp sensing. The following examples are ordered by position of the sensor on the user's arm, starting at the fingertips and going up to the upper arm (Figure 4.1).

While Holz et al. (2012) investigated sensors implanted into the user's body, most sensing systems for capturing real-time physiological data are attached removably to the user's body and could also be integrated into smart clothing.

Kry and Pai (2006b) present a prototype that combines force sensors attached to the fingertips and an external marker-based tracking system. This combination allows for capturing grasps in more detail than with each technique on its own. As presented in the paper, the setup can be used to capture more realistic grasping movements to be used in animating virtual characters. Force sensors that are optimized for placement on the user's fingertips are commercially available and used for ergonomics research.[8]

Embedding multiple pressure sensors into a glove might be useful for capturing grasp forces and the distribution of such forces across the individual fingers. As the pressure sensors completely cover the user's fingertips, they may lower or increase the friction between fingertip and object. Additionally, covering the fingertips reduces tactile feedback, potentially causing the user to fasten her or his grip (Shih et al. 2001). Both factors may affect how the user grasps an object. Therefore, such approaches should be only used for grasp sensing in scenarios where the user would already wear gloves.

Mascaro and Asada (2001) avoided covering the fingertips by attaching optical sensors to the fingernails. This approach predated their camera setup presented in the previous

---

[7] Thomas Feix, personal communication, 29. July 2010. See also his taxonomy (Feix et al. 2009b).

[8] see e.g. `http://www.pressureprofile.com/finger-tps`

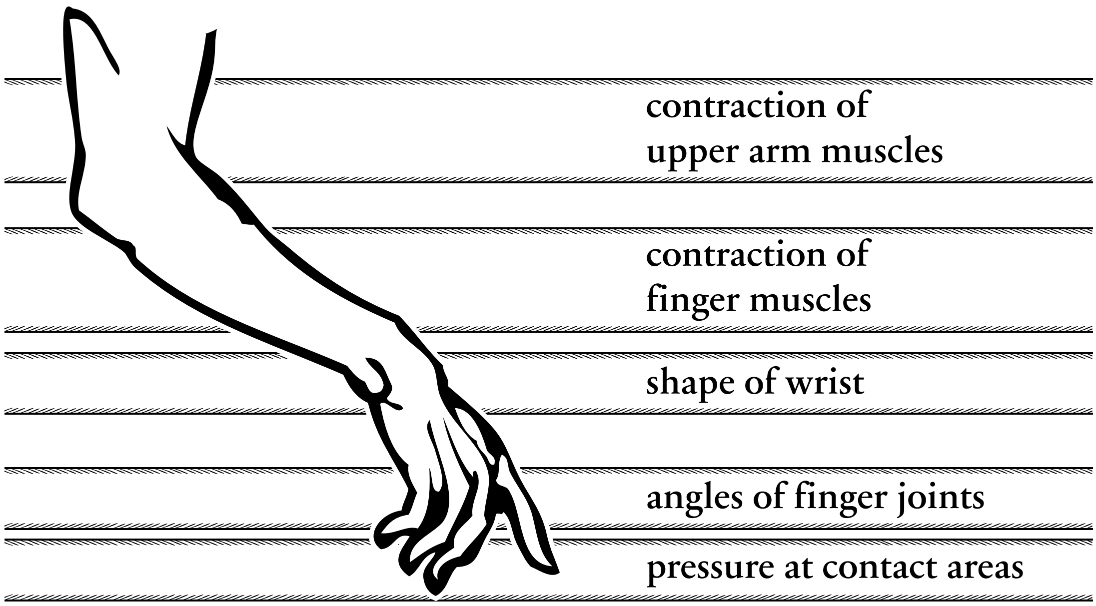


**Figure 4.1:** Applying sensors to a user's arm can provide different types of grasp information.

section. From the changes in the fingernail's coloration, Mascaro et al. derive perpendicular force and shear force for each instrumented fingertip. While the spatial resolution of the optical sensors is lower than that of an external camera, fingernail-mounted sensors are not subject to occlusions, limited field-of-view, or changes in environmental lighting[9]. However, the system requires routing cables from the fingernails to a (potentially body-worn) computer. Mascaro published several papers on this topic (Mascaro, Chang, and Asada 1999; Mascaro and Asada 2000; Mascaro and Asada 2001; Mascaro and Asada 2004).

For measuring the angles between individual digits, data gloves (Zimmerman et al. 1987) equipped with flex sensors can be used. These flex sensors are long, thin strips attached to the back of the glove, each one covering the back of a finger. Bending such a flex sensor decreases its internal electrical resistance, which can be measured at two electrodes on one end of the sensor. For example, Ingram et al. (2008) captured finger movement during everyday tasks using data gloves.

Another approach employs accelerometers attached to the user's finger joints. Fukumoto and Suenaga (1994) use finger-mounted accelerometers for distinguishing between tapping with different fingers. Lobo, Trindade, and Dias (2011) propose an approach for combining multiple sensors in a glove and within graspable objects in order to reconstruct grasp types from Feix' taxonomy. Hrabia, Wolf, and Wilhelm (2013) re-

[9] Stephen Mascaro, personal communication, 24. August 2012.

duce the number of sensors required for reconstructing hand postures to eight.

Rekimoto (2001) presents a wristband with embedded capacitive sensors which is capable of distinguishing different hand postures. The sensors measure the distance between wristband and wrist at several locations with high precision. As the extrinsic finger muscles, located in the forearm, contribute greatly to the hand posture, many changes in hand posture also cause changes in the shape of the wrist's circumference. Rekimoto demonstrates that at least some hand gestures can be associated with distinct wrist shapes. However, the paper does not state how many different hand postures can be distinguished with this technique, and how reliably it works.

Several researchers in the area of prosthetics and biomedical engineering (Reddy and Gupta 2007; Smith et al. 2008; Shrirao, Reddy, and Kosuri 2009) have investigated the use of surface electromyography (sEMG) for measuring the tension of extrinsic finger muscles in the forearm, and estimating the angles of the corresponding finger joints[10]. Surface electromyography captures the electrical activity of skeletal muscles using electrodes placed externally on the user's limbs. T. Scott Saponas and co-authors (Saponas et al. 2008, Saponas et al. (2009), Saponas et al. (2010)) have explored uses of sEMG for human-computer interaction, namely recognizing certain hand postures and gestures.

Determining hand posture using sEMG has a number of limitations that limit the use of this technique for grasp sensing:

- sEMG senses muscle activity. Therefore, small and slow movement of the finger joints is hard to detect.
- Each user is required to complete a training session lasting at least a few minutes[11].
- Without applying electrodes to the palm and the back of the hand, only activity of extrinsic finger muscles can be measured. The activity of intrinsic finger muscles, most importantly the *thenar* muscles controlling the thumb, can not be determined. Therefore, only a limited subset of hand postures can be captured using sEMG.
- Recognition accuracy for hand postures is not high enough yet. Both Ju, Kaelbling, and Singer (2000) and Saponas et al. (2009) report a recognition accuracy of about 80% for just four different, simple hand postures. Given that the human hand is capable of at least 17 distinctly different grasps (Feix et al. 2009b), such an accuracy is probably not sufficient for most use cases.

At the upper arm, measuring biceps muscle tension might also convey information for grasp sensing. While not telling anything about the grasp itself, biceps tension should correlate to the weight of the grasped object. I am not aware of any actual use of this approach for grasp sensing.

[10] (Smith et al. 2008) and (Shrirao, Reddy, and Kosuri 2009) are mentioned here just for reference. For these publications only the abstract was available to me.

[11] T. Scott Saponas, personal communication, 23. August 2012.

In theory, and in the distant future, brain-computer interfaces (BCIs) might allow capturing all information about grasping movements which is available in the brain. In practice, capturing and analyzing brain activity is still a great challenge. Significant research is being conducted on using BCIs for controlling prosthetic limbs. Invasive BCIs that require implanting electrodes into the user's skull provide better resolution and fidelity than non-invasive techniques but are still far away from widespread adoption. Therefore, they are mostly confined to laboratory experiments on human grasping. For example, Fifer et al. (2011) used an invasive method, electrocorticography (ECoG) for determining grasp aperture during reach-to-grasp movements, i.e. before actually grasping the object.

A more appropriate technique for capturing brain signals for grasp sensing would be electroencephalography (EEG), a non-invasive, real-time sensing technique using electrodes attached to the users head by suction cups or a harness. Agashe and Contreras-Vidal (2011) were able to derive the trajectories of several finger joints during grasping movements using EEG. I am not aware of any research that uses BCIs for capturing static grasps.

While the aforementioned techniques allow capturing information about a user's grasp, none of these is capable of detecting which object the user is actually grasping. However, getting information about the grasped object is essential for grasp interaction, as the meaning of a grasp frequently depends on the type of object that is being grasped.

Recognizing a grasped object can be done in various ways, such as using computer vision and a database with images of graspable objects. A more robust and efficient approach is to embed a unique ID into each object that might get grasped. This ID can be retrieved by a reader device mounted on the user's arm. Fishkin, Philipose, and Rea (2005) demonstrate this approach using an RFID (radio frequency ID) reader attached to the user's wrist or back of the hand. RFID tags are attached to or embedded in the objects. Berlin et al. (2010) improve on this concept by adding inertial sensors to the RFID-sensing bracelet. This allows them to not only identify objects but also simple gestures conducted while holding an object. Obviously, such an approach can only be used for a pre-defined group of objects with attached or embedded RFID tags. Additionally, these objects need to be large enough for attaching or embedding an RFID tag.

In summary, body-worn sensors can provide ubiquitous grasp information of different kinds and with varying degrees of precision. Almost every parameter of a grasp can be captured with high fidelity using body-worn sensors.

The major drawback of body-worn sensors is that they may make the wearer feel uncomfortable or impeded. Smart clothing has not yet become widely adopted, and research has just started on using sensors implanted into the body for interaction (Holz et al. 2012).

Additionally, the best place for such sensors - where most information about a grasp can be captured - is directly on the back of the hand, digits, palm and fingertips. However, attaching sensors to the hand inevitably impedes the hand's normal operation and is not widely accepted as fashionable. The most acceptable solution - both socially and from a technical standpoint - might therefore be thin gloves with embedded sensors. As mentioned above, even extremely thin gloves affect how the wearer grasps and manipulates objects.

In summary, instrumenting the user can be helpful for investigating how people grasp in everyday scenarios. Until body-worn sensors become socially accepted and interfere less with everyday hand movements, they are not suitable for practical applications, however.

A comparison of body-worn grasp sensing to environmental grasp sensing and grasp-sensitive surfaces can be found in Section 4.5.

## 4.4 Instrumenting Objects

A third approach - besides instrumenting environment or user - is to incorporate grasp sensing into the objects the user is interacting with. Generally this means making the whole surface of an object or a part of it grasp-sensitive. This third approach is the focus of my dissertation. Therefore the following discussion of related work aims to be comprehensive and contains more detail than the previous two sections.

Grasp-sensitive surfaces have been subject of academic and commercial research for about ten years. Most research has focused on techniques for implementing grasp-sensitive surfaces, with only little interest in applications that are enabled by grasp-sensing objects.

The structure of each subsection dealing with a prototype follows the three stages of a grasp-sensing workflow as discussed in Section 3.5: (a) sensor technology used for capturing a grasp signature, (b) data processing used for identifying a grasp, and (c) applications based on the grasp type.

### 4.4.1 Technologies for Grasp-Sensitive Surfaces

Several technologies allow for sensing human touch.

The four most relevant touch sensing technologies discussed in this chapter are *capacitive proximity sensing*, *resistive or capacitive force sensing*, various *optical sensing* methods, and *impedance sensing*. Table 4.1 contains a comparison of these technologies with regard to characteristic properties of grasp-sensitive surfaces.

**Capacitive proximity sensing** (Cremer 1907; Beck and Stumpe 1973; Lee, Buxton, and Smith 1985; Smith 1999; Baxter 2000; Barrett and Omote 2010) is widely employed in touch screens for mobile devices. Several sensing modes with different capabilities exist (Smith 1999). In all of them, a touching body part increases capacitive coupling between an electrode and ground or between electrodes. This change in the electrode's capacitance is measured and converted into a proximity value. Capacitive sensing allows for detecting and measuring proximity and touch through non-conductive materials. However, only certain conductive objects, such as fingers, can be detected at all[12]. Close-by conductive objects severely reduce sensitivity and may be mis-identified as close-by or touching fingers. Surfaces are covered either using many individual electrodes or a matrix of orthogonal wires. Optically transparent electrodes can be made of Indium Tin Oxide (ITO).

| | capacitive | impedance | optical | force |
|---|---|---|---|---|
| only sensitive to human tissue | | + | | |
| sensing through opaque plastics | + | | | |
| contactless | + | | + | |
| proximity sensing | + | | + | |
| sensing thickness of touching tissue | + | ? | ? | |
| low ambiguity of sensor readings | | + | + | + |

Table 4.1: Capabilities of the four predominant sensing technologies for grasp-sensitive surfaces (slightly adapted from (Wimmer 2010a)).

**Resistive or capacitive force sensing** measures contact force and location at one or more points on the surface. Discrete resistive/capacitive force sensors measure the resistance/capacitance between two electrodes separated by a compressible spacer. Traditional resistive touchscreens measure a voltage gradient between a rigid and a deformable conductive surface separated by spacers. They only allow for locating a single touch or the centroid of all touches. However, sophisticated measuring approaches allow for also extracting contact area size, which is proportional to the force exerted by all touches. Unlike capacitive proximity sensors, force sensors can not detect proximity. However, they may be operated by both conductive and non-conductive objects.

**Optical touch and proximity sensing** technologies exist in various forms. For touch sensing, *Frustrated Total Internal Reflection* (FTIR) (Han 2005) and *Diffuse Illumination* (DI) (Schöning et al. 2008) are the most common approaches. Only the latter allows for sensing proximity in addition to touch. Both approaches use infrared light. Optical ap-

[12] Actually, it is a little bit more complicated and depends on the sensing mode. For instance, a coin can be easily detected using transfer mode but not using shunt mode capacitive sensing.

proaches are sensitive to conductive and non-conductive objects and may be negatively affected by environmental light sources.

**Impedance sensing** (Mäntyjärvi et al. 2004) is rarely used in practice. It is technically similar to measuring the galvanic skin response. The mode of operation is described later when discussing the prototype by Mäntyijärvi et al. Impedance sensing allows for distinguishing between touching body parts and other conductive objects. Unlike all other technologies described before, it requires the user to directly touch an electrode. Impedance sensing shares some properties with swept-spectrum capacitive sensing (Sato, Poupyrev, and Harrison 2012) which is also discussed in more detail later.

### 4.4.2 Relationship to Touch-Sensitive Surfaces

Touch-sensitive surfaces have been subject of scientific research for decades. However, one cannot directly translate the findings from this research to grasp-sensitive surfaces. At least three significant differences to touchscreens need to be taken into account when designing grasp-sensitive surfaces:

**Surface** Touchscreens are preferably flat or only slightly bent. As documented for example in our Curve study (Wimmer et al. 2010) and by Roudaut et al. (Roudaut, Pohl, and Baudisch 2011), non-planar surfaces pose a number of challenges for touch interaction. Grasp-sensitive surfaces need to be non-planar in order to be graspable. While a grasp-sensitive surface may be composed only of planar surfaces (e.g., a box), ergonomic and design requirements often require graspable objects to have curved surfaces. Therefore, grasp-sensing techniques need to support curved surfaces. Additionally, the sensor hardware needs to be small enough to be fitted into a graspable object.

**Sensor Resolution** Touch input has traditionally been used for pointing tasks. Therefore, the shape of the contact area between finger and touch-sensitive surface is less interesting than the touch position. For pointing tasks, only a single touch position is needed. Grasp sensing requires different touch information. Instead of touch points, contact areas need to be captured. Some current touch sensor technologies allow capturing a high-resolution bitmap of contact areas and pressure values.

**Data Model** In order to be useful, grasp signatures captured by a grasp-sensitive surface need to be analyzed by a grasp recognition system or other software. For this, an interface between both is necessary. While all major operating systems offer standardized single- and multi-touch interfaces, none of these interfaces support grasps. For example, all OS APIs are based on the notion of single touches/pointers that may have additional properties like *pressure*. However, there is no standardized way for a sensor driver to report the *outline* or *pressure map* of a touch to the operating system or user space. Only in April 2011, TUIO 2.0[13], the de-facto protocol standard for platform-independent multi-touch interaction, has added attributes for describing contour and raw bitmap of a touch event. Additionally, grasp-sensitive surfaces require a suitable coordinate system. As the surface wraps around the object, it has to be either projected onto a planar coordinate system, described as a 3D mesh, or represented in an application-specific way. Section 3.5 describes some approaches to this challenge.

### 4.4.3 Early Research on Grasp Sensing

One of the first instances of an object that detects how it is being held by the user is presented by Harrison et al. (1998). A palmtop computer is equipped with two pressure sensors attached to the left and right side (and other sensors). This allows the device to detect whether the user is holding it with the left or right hand. Assuming that the user therefore uses the other hand for interaction on the touchscreen, the computer would optimize the graphical user interface (GUI) for right-handed or left-handed use.

Harrison et al. also report qualitative findings from a user study:

> "While the user explicit interactions were quickly understood, the passive interaction (handedness) was perceived as "magical". Since no explicit commands or manipulations were needed, users seemed amazed that the device recognized and optimized for handedness. They were unable to tell how this was accomplished without us explaining it. This suggests not only that passive kinemes can be powerful, but, when well integrated with the device form factor, they greatly impact the users' interaction experience. We clearly need to explore more passive manipulations to see if this is a general property for passive kinemes." (Harrison et al. 1998).

Hinckley and Sinclair (1999) present the *Scrolling TouchMouse*, a computer mouse with a few embedded binary capacitive touch sensors. It detects when a user grabs or releases the mouse and adjusts the graphical user interface accordingly. Hinckley and Sinclair suggest that an application's tool bars might be only shown while the user is grasping the mouse, increasing the available screen size when not using the mouse.

Hinckley et al. (2000) present a palm-sized computer with a large capacitive touch sensor covering the back of the device. It is used for detecting whether the user is holding the device. (A second touch sensor is located near the screen bezel and used for (explicitly) scrolling screen contents.) Hinckley et al. suggest that such a sensor can be used to only activate certain actions while the user is holding the device. They also suggest that the computer should not automatically power off while being held by the user.

[13] `http://www.tuio.org/?tuio20`

**Discussion** These three systems only allow for sensing whether someone is grasping the device, and on which side(s) the user is holding the device (Harrison et al. 1998). The latter information might be used for inferring the hand the user is using for interacting with an embedded touchscreen or keyboard. As pointed out by Harrison et al. (1998), implicit interaction ("passive kinemes") supported by grasp sensors may significantly enhance the user experience.

### 4.4.4 Secure Grip Project

From 2003 to 2008, Raymond Veldhuis' group at the University of Twente published several papers on grasp-sensitive gun butts employing an embedded piezo-resistive force sensing matrix. The focus of their research was to reliably identify the person using the gun. For this they compared the pressure pattern captured by the sensor matrix to a database of previously recorded patterns. The stated goal was "to design a weapon that can only be used by the rightful owner" (Kauffman et al. 2003). To this end, Veldhuis et al. investigated several preprocessing and pattern recognition algorithms. Most of the research was conducted within the ''Smart Grip" project[14] sponsored by the STW technology foundation.

According to Raymond S. Veldhuis, research on this topic ended in 2008 because their very ambitious goal (see below) was found to be mostly unattainable. Additionally, Veldhuis wanted to concentrate on other topics[15].

**Sensor Technology** Veldhuis et al. used a piezoresistive pressure sensor made by Tekscan Inc for all research. It has a size of approximately 8 by 8 cm[16]. It is made of a flexible substrate and can therefore be bent along a single arbitrary axis. The sensing matrix consists of 44 x 44 silver traces with piezoresistive ink between traces at crossings. Each crossing represents a sensor element (sensel). Each sensel can capture a pressure range between 0 and 30 PSI (0 - 207 kPa) with 8 bit resolution. The sensor is connected to a host PC via a serial connection (RS-232) (Kauffman et al. 2003). According to Raymond S. Veldhuis[17], the Tekscan sensor was the best sensor hardware identified in a thorough survey of pressure sensors in 2002.

**Data Processing** For determining who is grasping the gun's butt, Veldhuis et al. tried out a variety of algorithms and presented them in several publications.

[14] `http://web.archive.org/web/20061009210426/http://www.sas.el.utwente.nl/home/SecureGrip/`

[15] Raymond S. Veldhuis, personal communication, 16. September 2012.

[16] Raymond S. Veldhuis, personal communication, 16. September 2012. The actual dimensions were never mentioned in any publication.

[17] Raymond S. Veldhuis, personal communication, 16. September 2012.

A first approach (Kauffman et al. 2003) used a likelihood-ratio classifier. Each of the 44 x 44 (= 1936) sensor values represented one feature. Using samples with so many features would require a lot of training data and computational resources in order to achieve reliable and fast recognition. Therefore, the number of features to be used was reduced by first performing a principal component analysis (PCA) for determining the features that were most indicative of a grasp. This was followed by a linear discriminant analysis (LDA) which identified features that varied most between users. Kauffman et al. (2003) also present a preliminary evaluation of their approach with 30 participants which indicates that a false acceptance rate (FAR) of 10% would correspond to a false rejection rate (FRR) of 10% and decreasing either one slightly would exponentially increase the other one.[18] This also means that the equal error rate (EER) was 10%.

Veldhuis et al. (2004) present an improved version of this classifier and an evaluation with 26 users. They found an average EER of 1.8% which they deemed good. However, the achievable FRR was deemed to high, as it significantly exceeded the rate of gun malfunctions allowed in the Netherlands - which according to the authors is 0.0001 % (or 1 in 10,000 cases of gun use).

Shang investigated several preprocessing and classification algorithms in her PhD thesis (Shang 2008) and further publications. She investigated the use of support vector machines (SVMs) - which are a popular class of machine learning classifiers. She found that these worked well for noisy sensor data but lost their superiority when the data is preprocessed or features have been reduced.

Shang and Veldhuis (2007) present a preprocessing approach called Local Absolute Binary Patterns (LABP) that normalizes the sensor data between different recordings: They found that the absolute pressure values generated by the same person during different sessions changed massively. This made then difficult to compare using the existing classifier. Shang et al. solved this by applying a filter kernel to the raw values that calculates the variance of pressure values for each 3 x 3 sensel area. Therefore, areas where no pressure is being applied stay at a very low value. Areas with high pressure values (which also contain high absolute variance) get high values. The filter also fills small gaps in the sensor matrix caused by non-responsive sensels. For a given set of samples, the LABP filter reduced the average EER from 7.2% to 3.6%.

Finally, Shang and Veldhuis (2008) present two additional classification algorithms called Maximum Pairwise Comparison and Mean Template Comparison. Both achieve an EER of 3% - 4%. At an FRR of 10ˆ-4 (the required legal maximum for gun malfunction), the Maximum Pairwise Comparison algorithm achieves an FAR of 45%.

**Applications** As noted above, the main goal of the research conducted by Veldhuis et al. is to implement a smart gun butt that can identify the user grasping it. Only

[18] This interpretation is based on Figure 11 (right) from the paper. No accurate textual description is given in the paper.

authorized users should be able to fire the gun. However, while their research lead to EERs of only a few percent, the requirement for a FRR of less than 0.01 % lead to a FAR of over 40% (Shang and Veldhuis 2008) - which was not deemed acceptable. As Shang (2008) noted at the end of his PhD thesis: "We conclude, therefore, that based on the approaches and methods investigated grip-pattern recognition is not suitable to be used for a police gun."

Nevertheless, this research provides strong evidence that grasp patterns are suitable for many biometric applications.

Based on this research, Buhan et al. (2006) used grasp patterns as a shared key for establishing a secure communication channel between two grasp-sensitive devices. In order to couple two devices, the user would grasp both devices consecutively with the same hand. As the grasp patterns are inevitably noisy, they are never exactly the same for both devices. Therefore, Buhan et al. employed a fuzzy cryptographic algorithm that is resilient to small differences in the grasp patterns. The device which initiates communication encrypts a known message with the grasp pattern. Based on data collected previously, the decoding algorithm on the receiving side determines the bits in the grasp patterns that are most likely to vary between grasps of the same person. The decoding algorithm then flip each of these bits in its own copy of the grasp pattern one at a time and checks if the resulting key is valid, that is if it decrypts the message. As this brute-force cryptanalysis is done only once when establishing the communication channel, the impact on performance is low. Buhan et al. report an FAR < 0.1% and an FRR 1.7% if the decoding algorithm is allowed to flip up to three bits (i.e. try out eight different keys).

While this approach seems promising, the authors do not discuss the impact of different object shapes on authentication performance. Grasp patterns greatly differ for different form factors. Therefore, coupling two devices with different shapes - such as mobile phones from different manufacturers - might prove difficult.

**General Findings** Some further findings which are relevant for grasp sensing were reported in the aforementioned publications.

Kauffman et al. (2003) noted that "[i]t appeared that more experienced subjects (who had handled the gun more often and over a longer period) showed better results than first-time subjects. The three most experienced subjects even showed an FAR and FRR of 0% with this limited number of tests."

Shang (2008) confirmed this finding, noting that the grip patterns of police officers were found more stable than those of casual users.

Buhan et al. (2006) also used the grasp pattern data collected from the police officers. When matching grasp patterns recorded within a single session, they found an EER of 1%. However, matching grasp patterns against the patterns recorded in a previous session resulted in an EER of 15%. Veldhuis confirmed that this cross-session variability was the greatest challenge encountered in the research project[19].

**Discussion** Overall, the research conducted by Veldhuis et al. contributes valuable insight into approaches and limitations for processing grasp patterns. The very high requirements regarding the recognition accuracy - a lot higher than that of all other published research on grasp recognition - lead to the development of sophisticated algorithms for preprocessing and pattern recognition. As they used a single sensor for all of their research, it is not clear how much the success was limited by the choice of sensor technology.

Hardware limitations of the pressure-sensing matrix include:

- the sensor only reacts to force, not light touch or proximity
- the substrate can only be bent along one axis, limiting its use for more complex object geometries
- Damaged cables in the prototype caused missing lines in the captured grasp patterns. Shang et al. (2006) developed a reconstruction algorithm to mitigate these errors.

The findings from these research projects document general limitations of grasp sensing, independent of sensor technology:

- grasp patterns of different users are significantly different which makes them suitable for biometric purposes
- within a single session, expert users of a graspable object consistently grasp the object in the same way, while novice users employ slightly different grasps each time.
- over time, however, even expert users slightly change their hand posture when grasping, resulting in different grasp patterns.

### 4.4.5 Touch Detection System for Mobile Terminals

In 2004, Mäntyjärvi et al. (2004) presented a technique for detecting when the user holds her or his mobile phone.

[19] Raymond S. Veldhuis, personal communication, 16. September 2012.

**Sensor Technology** The prototype measures skin impedance to distinguish whether a hand or the fabric of a pocket or carrying pouch is touching the mobile phone.

Skin impedance is the electrical resistivity of human skin as a function of the frequency of an AC signal that is applied. Mäntyjärvi et al. chose measuring skin impedance instead of skin conductivity because the former approach is better able to distinguish between skin and metal objects (such as keys) touching the sensor electrodes.

In the prototype presented in the paper, two sensor electrodes are embedded in the side of a mobile phone. A sinusoidal signal with voltage of 600 mV is applied to the electrodes. During a sensing sweep, the frequency is increased from 1 kHz to 100 kHz in 11 steps. For each frequency, the resistance between the electrodes is measured.

**Data Processing** Measuring the impedance of a conductive object results in a characteristic curve, whereby for each step within the sensing sweep the frequency of the applied signal is marked on the X axis and the corresponding resistivity marked on the Y axis.

Mäntyjärvi et al. compared two approaches for feature extraction:

1) take the center point of the impedance curve, i.e., the average resistance of all points of the curve, and

2) calculate the coefficients of the impedance curve and use these as features.

Two basic classifiers were compared, too: Linear Discriminant Analysis (LDA) and Quadratic Discriminant Analysis (QDA).

In a small user study, skin impedance of eleven participants was recorded. Additionally, the impedance of pieces of fabric and leather was recorded to simulate the mobile phone being in a pocket or carrying pouch.

Cross-validation of these measurements showed that both LDA (100% accuracy) and QDA (97% - 100%) could reliably distinguish between skin and fabric/leather when using the center points of the impedance curves as features. When using the function coefficients as features, QDA (95% - 99%) showed better results than LDA (72% - 98%).

**Applications** The presented research was purely limited to detecting whether a mobile phone is being grasped by the user. No further information about the grasp was inferred.

**General Findings** None.

**Discussion** As demonstrated by Mäntyjärvi et al., measuring skin impedance has a major advantage compared to capacitive touch sensors: even a thin layer of fabric between skin and sensor electrodes significantly alters the measured impedance curve. This might significantly reduce false recognition of a grasp, e.g. when the device is in a pocket. On the other hand, impedance measurements require electrically conductive electrodes on the outside of the device, impacting the design of the device's shell. It would be interesting to explore whether measuring skin impedance can be used for authentication or determining other properties of the user grasping the device.

### 4.4.6 Tango

In 2005, Pai et al. (2005) presented *Tango*, a grasp-sensitive sphere employing capacitive force sensors. This initial presentation of the hardware was followed by a paper discussing a grasp recognition approach for spherical objects (Kry and Pai 2006a). As mentioned in the previous section, Kry and Pai also explored instrumenting the environment and the user in another paper published in the same year (Kry and Pai 2006b).

**Sensor Technology** The Tango (Pai et al. 2005) is a spherical input device with an embedded 3-axis accelerometer and a grasp-sensitive surface covering the majority of the sphere except for the "poles". Its size is about 70 mm in diameter[20].

The grasp-sensitive surface is implemented as a matrix of conductive strips. These strips are laid out in two layers of 32 "meridians" and eight perpendicular "circles of latitude". Both layers are separated by a thin layer of non-conductive rubber foam. Similar to projected-capacitive touch screens (Barrett and Omote 2010), each crossing of these strips forms a capacitive sensor, resulting in 256 sensor elements covering the surface. Unlike projected-capacitive touch screens - which operate in *shunt mode* (Smith 1999) - the surface of the Tango is not touch-sensitive, however, but pressure-sensitive. Pressing on the surface, compresses the rubber foam separator, reducing the distance between outer and inner conductive strip. This increases the capacitance between both. The sensing hardware sequentially measures the capacitance between each horizontal and vertical strip and converts it to an 8-bit value for each of the 256 sensor elements, achieving an update rate of 100 Hz. All measurements are sent to a host computer via USB or Bluetooth.

As depressing a strip on the outer layer also depresses adjacent strips on the inner layer, the boundaries between pressed and non-pressed areas are blurred. Therefore all raw measurements are deconvolved using a transfer function derived from calibration data.

[20] Estimated based on photos in (Pai et al. 2005). No information about the size can be found in the papers.

**Data Processing** The initial grasp-sensing algorithm (Pai et al. 2005) uses a simplified hand model with 11 degrees of freedom, simulating only three-finger precision grasps. In a first step, three pressure clusters, corresponding to thumb, index finger, and middle finger are identified by simple thresholding and clustering algorithms. A naive heuristic is used to assign thumb, index finger, and middle finger to the three pressure clusters: The thumb is farther away from index and middle finger. The order of the fingers (i.e. handedness) is hard-coded. An extended Kalman filter is used for deriving the hand posture. This approach assumes that the Tango is always grasped from a pre-defined direction, i.e., that all fingers are positioned on or near the Tango's 'equator', as defined by the sensor layout.

Kry and Pai (2006a) present an improved approach for grasp-sensing on the Tango that allows for an arbitrary orientation of the sphere. A more sophisticated clustering algorithm is used that is also robust against small gaps in the pressure clusters, caused e.g. by faulty sensor elements.

For conducting rotationally invariant comparisons of captured pressure distributions to stored patterns, Kry and Pai employ *spherical harmonics*. Similar to the effect of a Fourier transform, the pressure pattern in X-Y coordinates gets represented as a feature vector of ten frequency components. This mapping is locality-preserving, that means it ensures that similar grasps generate similar feature vectors, independent of the orientation of the sphere with regard to the hand. In a further step, each feature vector is projected into a truncated PCA space with only 6 dimensions/features, reducing noise and memory requirements.

In a training phase, these "pressure hashes" (Kry and Pai 2006a) are matched to actual hand postures by simultaneously capturing pressure patterns using the Tango and hand postures using optical marker-based tracking.

For grasp recognition, the hand posture whose associated pressure hash most closely resembles the currently captured pressure hash is retrieved. An additional "rest pose" is retrieved when no pressure is observed. Kry and Pai also present some performance improvements for faster lookup. No formal evaluation of recognition accuracy was reported.

**Applications** Pai et al. (2005) suggest using the Tango for navigation and interaction within a 3D scene. Two-finger and three-finger grasps may be distinguished and assigned different actions or modes. The raw pressure profile can also be used for manipulation, e.g. virtual modeling clay.

**General Findings** While Pai et al. (2005) propose assigning actions to different pressure distributions, e.g. generating a mouse click event when the "index finger is pressing harder than the middle finger", Kry and Pai (2006a) report that using single-finger clicks does not work as intended, as the other fingers have to exert a higher force, too. This makes it "difficult to control the pressure of one finger independently of the others".

**Discussion** As mentioned by Kry and Pai (2006a), grasp recognition on the Tango is limited to precision grasps. However, they expect their approach to also be usable for conforming and palmar grasps.

Tango introduced a unique capacitive pressure sensing matrix that has not been replicated so far. Kry et al. employed *spherical harmonics* in order to elegantly implement rotationally invariant grasp recognition. This approach, too, has not been re-used so far. Unfortunately, both publications on Tango lack a detailed evaluation. Therefore, it is hard to judge how well Tango actually works. The earlier research by Veldhuis et al. is not referenced.

### 4.4.7 Samsung Prototype

In 2006, researchers at Samsung Advanced Institute of Technology (Kim et al. 2006) presented a prototype of a grasp-sensitive mobile phone that uses 64 capacitive touch sensors on a flexible circuit board. Kim et al. argue that "there are natural grip patterns [. . . ] when using hand-held tools"[21] which are indicative of how the user wants to interact with these tools.

These "natural grip patterns" are determined by the affordances of the tool - intrinsic properties of a user interface that offer obvious ways of interacting with it.

They are also relevant when using mobile phones for different tasks. For example, Kim et al. suggest, people usually hold a mobile phone with both hands when writing text messages. The goal of the presented research was "finding the most natural grip patterns for the pre-defined set of mobile applications".

Kim et al. describe grasp recognition as a process consisting of four stages (Figure 2 in the paper):

(a) the user grasps the device,
(b) the device senses "touch region information",
(c) it pre-processes and classifies this information (called "Recognition"),
(d) and executes appropriate applications on mobile device.

This approach partially inspired my own characterization of grasp sensing as described in Section 3.5.

[21] While the authors exclusively refer to "grips" in their paper, I use the term "grasps" for consistency.

**Sensor Technology** The prototype mobile phone mock-up (no concrete size information given) contains 64 capacitive touch sensor electrodes, etched into a flexible PCB and attached to the inside of the 1.8 mm thick plastic case. The electrode size of 8 x 8 mm was figured out by "trial and error". These sensor electrodes are located on the back, the left and right sides, and on the top of the device. The front of the device houses a touch-screen whose sensing technology is not described in the paper and which is not used for grasp sensing. The bottom side houses a PC connector.

The sensor electrodes are connected to eight ESSD SS01 8-channel capacitive sensor ICs. Each channel offers a resolution of 8 bit and an update rate of 30 Hz. All data analysis is done on-board by an ARM CPU. Due to the limited processing power of this chip and memory limitations, only a subset of applicable machine-learning approaches is investigated. An embedded accelerometer is used for detecting when device is held still. However, for no apparent reason, the accelerometer is not used for determining the orientation of the device.

Kim et al. distinguish eight distinct grasp patterns which map to five applications: accepting a call, composing a text message (single-handed and two-handed operation), taking a picture (one grasp pattern for horizontal orientation, two different grasp patterns for vertical orientation), watching a video, and playing a game on the mobile phone. These applications were chosen based on a field study conducted beforehand. For each application, one or more prevalent grasp patterns were chosen based on interviews with study participants.

**Data Processing** Kim et al. employed a naive Bayes classifier and support vector machines (SVM) for classifying grasp patterns. As the on-board CPU had limited processing power, the classifiers were trained off-line, i.e. on a desktop computer. The models' parameters were extracted and supplied to the on-device classifiers. Each sample consisted of the sensor values of all 64 sensor channels, resulting in 64 features. No feature-reduction was applied. As mentioned above, the accelerometer data was not included in the samples. Training data was collected from 50 participants. Each participant conducted five iterations of each of the eight grip patterns, resulting in 250 samples per grip pattern to be recognized. In order to capture only static grasps, the capturing algorithm analyzed the changes in accelerometer data and waited until the participant held the device in a stable orientation. Then 15 samples were captured and averaged. Kim et al. found that ideally more samples would be needed in order to improve the recognition rate. However, "the cost (both time and effort) of acquiring the hand grip training data from subjects was prohibitive", so they stayed with their initial set of training data.

No additional user study was conducted. Instead, the classifiers' accuracy was determined 'in vitro' using cross-validation. The Naive Bayes classifier reached a relatively low accuracy of 79%, while the SVMs offered an accuracy of 93% - 100%, depending on SVM kernel. For on-device recognition, only the linear kernel with 93%+ accuracy was suitable due to memory limitations.

Kim et al. mentioned the need to distinguish between actual, static grasps and "unintended user hand grip", i.e. manipulative hand movements as a major challenge. They tried using the likelihood of correct classification - as reported by the classifier – to detect static grasps. However, this approach did not work reliably. Future work on this issue was proposed but not published so far.

**Applications** The prototype presented by Kim et al. distinguishes the aforementioned eight different ways it is being held and changes its screen contents accordingly. For example, when the device is held vertically with both hands at its lower end, it switches to 'test message' mode, displaying a keyboard on the lower half of the screen and a text entry field on the upper half. The authors argue that this kind of automated mode switching allows reacting to the user's intentions more quickly. A video demonstrating concept and prototype was presented at CHI'09 (Lee et al. 2009)

**Discussion** The research published by Kim et al. consists both of an impressive technical contribution - a functional prototype with high sensing resolution - and an evaluation that is rather limited in comparison. It can be assumed that involving accelerometer data, and reducing the number of features in each sample might have improved the - nevertheless quite acceptable - recognition accuracy. As no study of real-world phone usage has been conducted, the reported results only tell little about recognition accuracy and potential issues during actual use.

### 4.4.8 Graspables

In 2009, Taylor and Bove presented the *Bar of Soap* and the *Ball of Soap*, two grasp-sensitive objects using capacitive touch sensor matrices[22]. The goal of their research was to find out "how measuring the way people hold and manipulate objects can be used as a user interface" (Taylor and Bove 2009).

To this end, Taylor and Bove built the "Bar of Soap", a multi-function device that "implicitly understands users' intentions" and changes its personality accordingly. Similar to the prototype presented by Kim et al. (2006), the *Bar of Soap* shows an interface mock-up on the LCD which corresponds to the operation mode the user implicitly chooses. Taylor and Bove implemented five modes: camera, game pad, phone, personal data assistant (PDA), and remote control. A second prototype, the "Ball of Soap", was built to explore implementation and usage of a spherical grasp-sensitive object. It did not switch between different operating modes.

[22] Some of the information presented in this section could not be found in the conference paper but only in Taylor's Master's Thesis (Taylor 2008) which apparently acted as the basis for the paper.

**Sensor Technology** The Bar of Soap is a rectangular box of 115 * 76 * 33 mm containing a 3-axis accelerometer and 72 discrete binary capacitive sensor electrodes, connected to three QProx QT60248 chips. An Atmel micro-controller reads sensor values from the QProx chips and sends them to a host computer via Bluetooth. Two LCD screens on the largest sides can display arbitrary information.

Sensor electrodes on the four smaller sides of the box are etched into copper-coated circuit boards. The LCD screens are covered by electrodes made out of transparent plastic strips coated with indium tin oxide (ITO) to make them conductive. To improve coupling between the user and the capacitive sensors, the edges of the box were lined with a copper strip connected to the ground pins of the QProx chips[23].

The second prototype, the *Ball of Soap*, is a 62-sided sphere with a diameter that "approaches three inches" (approx. 75 mm). One side contains a power button and a programming connector while each of the remaining 61 sides contains a sensor electrode made from self-adhesive copper foil. As with the Bar of Soap, three QProx QT60248 chips are used for sensing touch. To improve capacitive coupling between user and *Ball of Soap*, a grounding bracelet worn by the user can be connected to the ball.

**Data Processing** The grasp pattern captured by the *Bar of Soap* is transmitted to a computer via Bluetooth. Only static grasps are analyzed. Sampling and pattern classifications takes place every three seconds.

For feature reduction, the surface is split into eight areas: one for each of the four smaller sides and two areas on each of the large sides incorporating the displays. During grasp recognition, the number of touched electrodes within each area is counted. Therefore, a sample consists of eight features with approximately 72/8 = 9 different possible values (~ 3.17 bit). Taylor and Bove found this primitive approach to produce better results than Principal Component Analysis and Fischer Linear Discriminants. As the box has a symmetric shape, there are four orientations of the device that look identical to the user. The internal orientation of the device - and the internal order of the touch-sensitive areas - are not obvious to the user. Instead of training the classifier to recognize each grasp in four different orientations, the *Bar of Soap* internally reorients the order of sensor areas according to accelerometer readings[24].

The evaluation was split in a single-user evaluation and a multi-user evaluation. In the single-user evaluation, for each mode 39 sample grasps were recorded by a single user[25].

[23] Actually, Taylor and Bove (2009) only report that "[r]esponse was further improved by lining the edges of the device with a grounded copper strip." and that for the Ball of Soap, "[a] grounding bracelet can be attached to improve sensor response". As both devices are wireless, it is probable that "ground" is not earth but the GND pins of the respective circuits.

[24] This might for example mean that the first feature always refers to the surface that is on the top. No detailed description is given in the paper.

[25] The paper reports: "For the single user part we collected 39 sample grasps in each functionality mode for a total of 395 grasps." This ("395" instead of "195") is obviously a typo.

Various classification algorithms were compared in this study: Templates, Neural Networks, Bayesian Classification, k-Nearest Neighbors, Parzen Windows, and General Linear Discriminants. In a 10-fold cross-validation, each classifier was repeatedly trained with 29 samples for each operation mode and tested on the remaining 10 samples.

For multi-user testing, each of 13 participants provided three grasp samples per operation mode. Participants were told the name of a mode and asked to hold the device in a way they personally would prefer for this mode. The classifiers were trained the complete single-user data and tested on the multi-user data.

Parzen Windows were found to offer the best recognition accuracy for the single-user data but the worst recognition accuracy for the multi-user data. Therefore, a Bayes classifier - which scored best overall - was used in the prototype.

Depending on the algorithm, 82 - 95% of grasps were correctly matched to the corresponding operation mode in single-user mode. In the multi-user test, recognition accuracy was 72% - 79%.

The *Ball of Soap* uses a different feature reduction approach, resulting in 12 groups/features. Each sensor electrode contributes (with different weightings) to the value of multiple adjacent groups. Each group's value ranges from 0 to 31 (5 bit). Grasp recognition starts when a throwing motion is sensed. No evaluation has been conducted for the *Ball of Soap*.

**Applications** Taylor and Bove propose several applications for grasp-sensitive devices:

Mode switching for mobile phones comprises the majority of the paper. Using 72 small binary sensor electrodes instead of larger electrodes with higher resolution allows for tracking the location of a single touch on the device's surface. Taylor and Bove demonstrate this advantage with a Rubik's Cube simulation (shown on an external display) where the sides of the cube can be rotated by sliding a finger along the corresponding edge of the *Bar of Soap*. Hidden Markov Models are used for detecting these sliding gestures. The *Ball of Soap* could be used in training and for controlling baseball video games. Grasp sensing could also be used as a safety mechanism to ensure that the user correctly holds a power device.

**General Findings** Taylor and Bove seem to have been surprised by the fact that users employed different grasp patterns:

> "functionality modes were selected based on the assumption that they would have distinct grasps associated with them. A large source of error in the multi-user study was when this assumption failed" (Taylor and Bove 2009).

They also found a significant effect of hand size on recognition accuracy. No explanation for this effect is given, and the authors suggest training with more users.

**Discussion** The research presented by Taylor and Bove is very similar to the research by Kim et al. (2006). In the paper, Taylor and Bove mention that both projects were conducted in parallel. They argue that their research has a wider scope. Taylor and Bove did cite most of the relevant related work - with the notable exception of the research conducted by Veldhuis' team. As a result, Taylor and Bove stumbled across two issues that were already identified by previous research:

- variation in users' anatomy causes different grasp patterns
- for the same task, different users may employ different grasps

As Taylor and Bove did not conduct a long-term study, they also did not identify differences in grasp patterns across multiple sessions.

Nevertheless, the paper offers a comparison of the accuracy of several algorithms for pattern classification as well as some novel application concepts.

### 4.4.9 GraspZoom

Miyaki and Rekimoto (2009) augment a mobile phone with unidimensional force sensing perpendicular to the touchscreen. This allows for one-handed switching between modes and continuous zooming by squeezing the phone.

**Sensor Technology** A single force-sensitive resistor is mounted between the back of an iPhone and an additional backplate. It is read out by a micro-controller which transmits the value to the phone via a serial connection. Neither update rate nor resolution are reported.

**Data Processing** The system distinguishes between three states: *out-of-range*, *touched*, and *pressed*. In the *pressed* state, the force reported by the sensor is also supplied to the application. An hysteresis is implemented to avoid rapid switching between *touched* and *pressed* states.

**Applications** Miyaki and Rekimoto propose two applications: continuous one-handed zooming and scrolling.

**General Findings** The authors did not conduct an evaluation. They suggest a few other ways for measuring touch force.

**Discussion** This paper presents a very simple example of grasp interaction. It is mainly included because of its title.

### 4.4.10 Touché

Sato, Poupyrev, and Harrison (2012) present a *swept frequency capacitive sensing* technique that significantly expands the information that can be gained from a single capacitive sensor.

**Sensor Technology** Instead of letting the capacitive sensor operate at a specific excitation frequency, the frequency is changed from 1 kHz to 3.5 MHz in 200 steps. At each step, the sensor's value is recorded. Graphing the sensor values as a function of the sensor's excitation frequency gives a "capacitive profile" (Sato, Poupyrev, and Harrison 2012) that is specific to the object touching the sensor electrode.

**Applications** This sensing technique allows for several novel applications. One application presented in the paper is detecting how a metallic doorknob is being grasped. Sato et al. distinguish five different grasp states that correspond to distinct contact areas between doorknob and hand:

(1) not touched,
(2) touched with a single fingertip,
(3) held between two fingertips,
(4) grasped between thumb and index finger,
(5) grasped with the whole hand.

**Data Processing** For the aforementioned application, an SVM classifier was trained with samples from 12 users. Each of the users provided 30 samples for each of the five grasps. In order to capture a greater variety of grasps, the users were also asked to adjust their grip slightly during sampling.

Each sample consists not only of all 200 raw values obtained by the frequency sweep, but also of several derivatives of the capacitive profile, and the locations of the minima in the "capacitive profile". The latter features are added as they make the classifier more robust against absolute variations in the raw data. It is not discussed or evaluated whether a subset of these features would be sufficient.

After the recording session for each user, the classifier was trained with only the samples from this user ('per-user classifier'). Afterwards, each user repeated each of the five grasps five times. The SVM classifier managed to recognize on average 96.7% of theses grasps correctly.

Additionally, a twelve-fold cross-validation of the captured data was conducted. In this case, the classifier achieved an average accuracy of 76.8%. Removing grasp #4 from the dataset increased the average accuracy to 95.8%.

**General Findings** Sato et al. found no correlation between accuracy of the classifier and the users' height, weight, body-mass index (BMI), or gender.

**Discussion** *Swept frequency capacitive sensing* looks like a promising sensing technique for grasp interaction. However, so far no research has been conducted on using multiple sensing electrodes. Therefore, hand shape and other properties of a grasp can not yet be determined using this technique. As a minor note: Although the sensing technique is very similar to the skin impedance sensing approach presented by Mäntyjärvi et al. (2004), Sato et al. do not cite this previous publication. Sato et al. did not conduct long-term studies, too.

### 4.4.11 Apple Patents

Hardware and software manufacturer Apple has applied for several patents concerning touch and grasp sensing on mobile devices. These patents do not describe novel sensing technologies but propose sensor layouts and applications.

Kerr, Hotelling, and Huppi (2006) suggest identifying users based on their grasp patterns, determining whether a device is held in the left or right hand, and adjusting the device's user interface accordingly. In the patent, Fig. 16 and 17A-D show a box with a fine matrix of 20 x 22 sensor nodes on the back of the device, and 20 x 8 resp. 22 x 8 sensor nodes on the sides. This concept is similar to the *Bar of Soap*. Fig. 18 shows the same box with (presumably) 3 large touch sensors on each of the sides and 9 large touch sensors on the back of the device.

In 2012 the USPTO granted Apple Inc. a patent (originally applied for in 2007) on "Flexible Touch Sensing Circuits" (Hotelling and Westerman 2012). The patent describes "a multi-touch skin placed along three-dimensions of an object for enabling multi-touch inputs during the operation of the object". Interestingly, the patent's claims and description cover only "an array of capacitance sensing nodes", limiting the patent to a specific sensor layout and technology.

The patent explicitly mentions grasp sensing as an application supported by this concept:

> "In the case of a tennis racket or a golf club, for example, the coordinates of the player's grasp of the handle can be detected. The coordinates can then be analyzed to determine if the player has correctly grasped the handle for a forehand stroke or a drive. The analysis may include obtaining other information, such as the orientation of the racket face or club head in relation to the grasp, to determine whether the player has correctly positioned his hand or hands on the handle. [...] Based on the analysis, an action can be performed." (Hotelling and Westerman 2012)

### 4.4.12 Driver Verification Based on Handgrip Recognition on Steering Wheel

R. Chen, She, et al. (2011a) investigated driver recognition using a grasp-sensitive steering wheel. Another paper by the same authors presents a classifier for dynamic grasps using essentially the same hardware setup and sensing workflow (R. Chen, She, et al. 2011b).

**Sensor Technology** The authors used a *Tactilus* pressure-sensitive mat made by Sensor Products Inc. It has 32 by 32 sensels - each with a size of 11 mm. While the overall size of the mat is not reported, it is presumably about 35 by 35 cm large. Each sensel can detect up to 388 mmHg (52 kPa) of pressure. Sensor values are reported with a resolution of 8 bit/sample at a sampling rate of 1-10 Hz. It is not reported why and how the sampling rate was not constant. The mat's properties do not match any off-the-shelf sensor offered by Sensor Products Inc. It is not reported how and in which orientation the mat was attached to steering wheel. The rather low spatial resolution does not allow for reliably distinguishing individual fingers, which is also indicated in Figure 1 of the paper.

**Data Processing** The classification workflow consists of feature extraction using a quadtree, dimensionality reduction using PCA, and classification using a likelihood-ratio classifier. First, a 24 x 24 px region of interest is extracted from each 32 x 32 px frame by (manual?) clipping[26]. From this image fine and coarse features are extracted using a quadtree. This means that the image is divided into 2 x 2 sub-regions. For each sub-region/node the average sensor value is calculated and added to the feature vector. Each of the sub-regions is recursively divided further in the same way until a 2 x 2 px block remains. This allows for the feature vector to effectively contain pressure data sampled at different resolutions. Somehow the quadtree generation results in 64 2x2 px

[26] Strangely, in the patterns shown in Figure 1 of the paper, all *raw* touch locations are centered on a 12 x 12 grid instead of a 24 x 24 grid.

nodes - i.e., more than half of all pixels are unaccounted for. The paper does not make clear why and how this happened. An intermediary, 85-dimensional feature vector[27] (presumably one first level node, four second level nodes, 16 third level nodes, and 64 fourth level nodes) is then reduced to 12 dimensions using PCA, retaining 97.5 % of the original data's energy. Finally, this feature vector is then matched against pre-recorded templates by a likelihood-ratio classifier, similar to those used by Veldhuis et al.

**Applications** The system described above was used for identifying car drivers. A study with 21 participants was conducted. Eight of the participants had no previous driving experience. Each participant recorded their grasp patterns by grasping the steering wheel with their left hand at least ten times for 5-6 seconds. Altogether 455 recordings of grasp patterns were collected. It is not made clear how these are distributed across the 21 participants. The authors report that they conducted a 4-fold cross-validation. This was somehow combined with two trials where the grasp patterns of three participants picked at random were compared to their own and other grasp patterns. The authors report similar recognition rates for both trials, resulting in a FRR of about 20% and a FAR of about 8%.

**General Findings** The authors encountered similar challenges as Veldhuis et al. (who are referenced several times in the paper). For example, multiple grasps by the same participant were very similar within a session but changed significantly between sessions. Participants with driving experience grasped more consistently than participants without driving experience.

Furthermore, the authors report that the participants' thumbs did not always show up in the sensor image. This certainly reduces the amount of information available for grasp recognition.

**Discussion** The use of a quadtree for representing both fine and coarse features in the data is interesting. However, I am not certain whether this step is necessary in general, as coarse features should also be extracted from within the finer features by the PCA. As mentioned throughout the descriptions of sensor hardware and processing workflow, the paper gets quite confusing at times. Unfortunately, none of the authors replied to my requests about the paper.

### 4.4.13 Grips and Gestures on a Multi-Touch Pen

Song et al. (2011) present a multi-touch pen that is able to detect how it is being grasped, and which finger gestures are performed on its surface. In a pre-study, they observed

[27] Later on the authors mention a 86-dimensional vector, which is probably a typographical error.

that artists hold a digital pen in different ways, depending on the task they want to accomplish with it. Mostly a standard *tripod grip* was used, but some other similar grips were employed, too. The majority of the hand surface touching the pen is used for stabilizing it with only few fingers left which can be used for pressing buttons on the pen or conducting gestures. With current digital pens, the barrel button - which is used e.g. for switching modes - can only be operated when holding the pen in a certain way. This makes it inconvenient or impossible to hold it in the way that is most apt for the task. Song et al. explored - among other interaction techniques - how well a grasp-sensitive pen could detect in which way it is being grasped.

**Sensor Technology** The prototype is based on a Wacom Intuos pen which is tracked by a graphics tablet. A tube, covered by a custom-built capacitive sensor matrix is slid over the pen's barrel, thereby covering the pen's integrated button. The sensor matrix (100 mm x 50 mm) consists of 20 x 10 orthogonal electrodes printed on a flexible substrate using conductive ink. The substrate is then glued to the tube, covering it completely. The sensing electronics are located on a circuit board at the end of the pen. They are connected to a host computer via USB. Sensors are read out at an update rate of 100 Hz. The resolution per sample is not documented in the paper.

**Data Processing** As Song et al. also explore explicit input techniques (tapping, gestures) on the multi-touch surface of the pen, they need to distinguish between static components (grasp) and dynamic components (gestures) in the grasp signature. This is done "based on their temporal characteristics", i.e. by identifying moving contact points in the grasp signature. The type of grasp is determined solely from the static components. Gesture recognition is only applied to the dynamic components. Picking up, putting down, and repositioning the pen in the hand also generates dynamic contact points, These should not be analyzed for gestures. Therefore, such non-gesture changes in the grasp signature are identified by the magnitude of the changes: When conducting a gesture, the bounding box of the dynamic part is relatively small. In contrast, adjusting the grasp causes changes in all parts of the grasp signature - which results in a rather large bounding box for the dynamic part.

Stable grasps are matched to grasp database using a Naive Bayes classifier. When holding a pen, its orientation is usually not relevant to the user. In order to achieve rotationally invariant matching, only rotationally invariant features are used by Song et al.: number, size, orientation, and eccentricity of contact points.

The prototype is able to distinguish four different grasp types, called *tripod*, *relaxed tripod*, *sketch* (called "Index Finger Extension" by Feix et al. (2009b)), and *wrap* ("Small Diameter").

Grasp recognition was evaluated with 10 users, each of whom provided 2000 samples per grasp type. Cross-validation within a single user's grasp data shows an average recognition rate of 87%. The 'warp' grasp type generated the majority of recognition errors. Removing "warp" grasp data from the cross-validation lead to 94% recognition rate. No cross-validation using samples of multiple users is reported.

A further user study compared mode switching using two different finger gestures, *double tap* and *swipe*, to mode switching by pressing the barrel button. No grasp recognition was involved in this evaluation. Gestures were found to be comparable in speed to the button but had a significantly higher error rate, mostly caused by false negatives.

**Applications** In addition to mode switching, Song et al. suggest several further uses of grasp sensing for enhancing pen interaction. A grasp-sensitive pen might be used for detecting when a user is switching between pen and mouse for input. The computer could automatically adjust the input focus on the screen, depending on which tool is being used. Gestures could also be used for invoking actions or adjusting parameters of a tool.

**General Findings** None.

**Discussion** Grasp sensing plays only a minor role in this publication.

Song et al. present an interesting method for differentiating between explicit gestures on the grasp-sensitive surface and implicit manipulations while changing between stable grasps.

Like Taylor and Bove (2009) and Sato, Poupyrev, and Harrison (2012), Song et al. omit significant previous research in their paper, too. In this case, grasp recognition on spherical objects had already been 'solved' by Kry and Pai (2006a). Unaware of this research[28], Song et al. implemented and described their own, less sophisticated method for grasp recognition on a cylindrical object.

### 4.4.14 GripSense

*GripSense* (Goel, Wobbrock, and Patel 2012) utilizes the built-in touchscreen, accelerometer, gyroscope and vibration motor of a smartphone for limited grasp sensing. As no grasp-sensitive surfaces are used, this research is only presented briefly. Goel et al. argue that detecting whether the user operates the touchscreen of a mobile device with the left thumb, right thumb, or an index finger can be helpful for adapting the user interface. The way the mobile device is being grasped partially determines which finger is used for interaction. GripSense uses data from internal sensors to estimate how the

[28] Hyunyoung Song, personal conversation after the conference talk at CHI '11.

user is grasping the device, and with which finger the user is operating the touchscreen. GripSense implements a quasi-passive sensing method, requiring action by the user in order to successively determine which finger is most probably being used for interaction. After capturing five interaction steps, GripSense is able to determine the finger with 80-90% accuracy. The system is also able to detect whether the user is holding the device but not interacting, and whether the device is lying on a flat surface. Goel et al. suggest that their approach can be also used for silencing a ringing phone by squeezing it.

### 4.4.15 iRotate Grasp

L.-P. Cheng et al. (2012a) presented a demo at UIST 2012, showing a mobile device that automatically adjusts its screen's orientation according to the way the user holds the device[29]. Auto-rotation algorithms for mobile devices rotate the screen to landscape or portrait orientation based on accelerometer readings. If users are lying horizontally on a couch or bed, with their head sideways, the automatically rotated screen is always oriented in the wrong direction. In a previous survey (L.-P. Cheng et al. 2012b) 91% of all participants reported having experienced inappropriate auto-rotation, with 42% regularly experiencing this annoyance.

The prototype developed by Cheng et al. analyzes the users' grasp pattern for determining whether they are holding the device in landscape or horizontal orientation. An evaluation shows that this approach correctly chooses the appropriate screen orientation in about 90% of all cases.

**Sensor Technology** An iPod Touch is embedded into a slightly larger iPhone 4S case which houses 32 PGM5526 photoresistors - ten sensors in two columns on the back of the device, seven on each long side, and four on each short side. The sensors on the sides are spaced 1.5 cm apart. The sensors are connected to an Arduino Pro Mini 328 through four CD4067B analog multiplexers. They are sampled at an update rate of 30 Hz and 10 bit resolution using an Arduino Pro Mini 328 which is connected to the iPod's serial interface. The prototype weighs 150 g, i.e. 10 g more than a iPhone 4S. As it is sensitive to changes in ambient light levels, studies were only conducted in a laboratory environment.

**Data Processing** Sensor data is downsampled to 8 bit per sample on the Arduino and transmitted to the iPod for further processing. The 32-feature vector is directly fed

[29] The published two-page demo paper is a shortened version of a paper that was originally submitted to UIST 2012 but got rejected. It has not been published elsewhere since. As the demo paper omits important details due to size constraints, I have incorporated information from the unpublished full paper and from personal communication with Lung-Pan Cheng, 29.12.2012.

into an SVM classifier which was trained using samples from six participants. Each participant conducted 108 grasps:

- 2 repetitions *of*
- 3 actions (scroll, pinch-to-zoom, type) *employing*
- 3 grasp types (left, right, both hands) *in*
- 2 user orientations (sitting, lying down on side) *and*
- 3 device orientations (portrait, landscape left, landscape right)

For each grasp, 10 seconds of sensor data were recorded at 30 Hz sampling rate, resulting in 194400 grasp signatures. Cross-validation showed a classifier accuracy of about 90% on average.

The classifier operates at an update rate of 25 Hz.

If the device's accelerometer reports rotation, grasp recognition is triggered. In order to achieve reliable classification, the most common classification result of five consecutive samples is taken. This results in a delay of 0.2 seconds.

**Applications** As described above, the prototype is able to automatically rotate the screen contents of a mobile phone depending on the grasp employed. A user study with 15 participants was conducted in order to evaluate real-life performance of the prototype. As the prototype is susceptible to changes in lighting conditions, the study was conducted in the same laboratory setting and at the same time as the training session. Participants conducted the same 54 tasks as used for training the classifier. They were allowed to grasp the prototype in any way but had to switch to the next task every 15 seconds.

While the default accelerometer-based auto-rotation feature selected the correct orientation in 60% of all cases, the grasp-sensing prototype managed to select the correct orientation in 87% of all cases (93% lying, 82% sitting).

**General Findings** In personal communication Cheng mentioned that users employed many different grasps. For instance, a participant grasped the device in an unusual way when holding the device in both hands: "He used his two palms without fingers to grip on devices".

Before building the current prototype, the authors also tried using the capacitive touch sensor foil from a Microsoft Touch Mouse. However, it reported incorrect touch events after bending it.

Cheng et al. note that the lock screens of common smartphones are always shown in portrait orientation, possibly compelling users to also hold the phone in portrait orientation for unlocking it. They suggest that this information may be used to automatically train a classifier to recognize grasps in portrait orientation.

**Discussion** This paper focuses on a clearly defined use case for grasp sensing. The authors identified important parameters and challenges in a pre-study, verified the accuracy of their classifier using cross-correlation, and evaluated the prototype in a user study. Cheng et al. are aware of existing related work and build successfully on it. It is very unfortunate that this research has not been fully published - further evidence of major flaws in pre-publication peer review.

### 4.4.16 iGrasp

Cheng et al. continued research on grasp-sensitive mobile devices with iGrasp (Cheng et al. 2013), focusing on interaction with on-screen keyboards.

In a pre-study with 64 participants the authors compared three grasp positions for a tablet:

- no grasp, tablet lies on a table,
- one hand holds the tablet, the other one types
- both hands hold the tablet, both thumbs are used for typing

Participants tried out four different keyboard modes - merged and split keyboard layouts in both a docked position on the lower edge and an undocked position more to the center of the screen. 98% of participants preferred different keyboard layouts and positions depending on how they were holding the device. For each of the three grasp conditions, a different layout/position combination was preferred by a large majority of participants (> 70%).

An iPad 2 was augmented with capacitive touch sensors. It distinguishes between different grasps and automatically adjusts the on-screen keyboard. One application - iGraspSwitch - automatically adjusts the layout of the on-screen keyboard to different one-handed and two-handed grasps. The other application - iGraspPosition - positions the on-screen keyboard always under the users' thumbs, independent of at which height they grasp the tablet.

**Sensor Technology** The backplate of an iPad 2 was outfitted with 46 binary capacitive sensors, 23 on each edge of the longer sides. The sensor electrodes consist of 40 mm x 8 mm strips of copper foil which are spaced at intervals of 10 mm, leaving a gap of 2 mm between electrodes. The electrodes are connected to four Freescale MPR121 binary touch sensor controllers. An Arduino - affixed to the back of the iPad 2 - reads out the touch state at a sampling rate of 60 Hz and sends the data to the iPad 2 over a serial connection. Outline and weight of the iPad 2 were not significantly changed by this augmentation, however the Arduino on the back changed the overall shape of the device.

**Data Processing** Binary grasp information from the capacitive sensors is handled differently for the two applications.

For iGraspSwitch, the 46 electrodes are grouped into four sensing areas: upper left, upper right, lower left, lower right. The individual sensor readings for each group are combined using an OR operator. Therefore, a sensor group would report a touch if one or more of its electrodes are touched. Heuristics are used to distinguish between no grasps, one-handed grasps, and two-handed grasps. If only sensor groups on one side of the tablet report touch, a one-handed grasp is recognized. If sensor groups on both sides of the tablet report touch, a two-handed grasp is recognized. However, during the pre-study the authors observed some users holding the tablet by grasping the one side on the upper half from the back and supporting the opposite corner of the tablet with their forearm. Therefore simultaneous contact in the upper-left and lower-right sensor group and vice versa is classified as one-handed grasp.

For iGraspPosition, the electrodes are grouped into 23 rows on each side. Grasp position along the long side of the tablet is calculated as the average position of all touched sensor electrodes.

**Applications** iGraspSwitch automatically selects the preferred layout for the on-screen keyboard, depending on how the tablet is being held. A user study with 18 participants showed that iGraspSwitch allows users to begin typing on average 1.49 seconds after the on-screen keyboard is shown - compared to 2.57 second for a standard, non-grasp-sensing keyboard. Users also preferred it to a manually adjustable keyboard.

iGraspPosition automatically adjusts the vertical position of the on-screen keyboard so that it appears closer to the typing fingers. A second user study showed that iGraspPosition allows users to begin typing earlier than with a fixed-position on-screen keyboard. However, it does not perform significantly better than iGraspSwitch.

**General Findings** One participant in the pre-study was wearing shorts and rested the lower edge of the tablet on his bare legs. This contact between body and sensor electrodes was incorrectly recognized as a grasp on the lower edge of the tablet. The authors therefore removed the lowest electrodes and changed the electrode layout.

The authors also mention limitations of capacitive sensors for grasp sensing (wet hands, gloves) and discuss implications.

**Discussion** This research project shares many properties with iRotate Grasp. Insights from the pre-study allowed the authors to refine their sensor layout and algorithms. This underlines the value of studying how users grasp an object before augmenting it with sensors. As iGraspSwitch shows, knowing how users grasp the device allows using very few sensors for reliable grasp sensing.

### 4.4.17 FlexAura

FlexAura (Liu and Guimbretière 2012) is a prototype of a flexible PCB covered with a high density of SMD IR range sensors. This flexible material allows for sensing grasp and proximity with high spatial resolution.

**Sensor Technology** The prototype consists of flexible PCBs holding a 16 x 24 array of sensor elements, each consisting of an SMD IR LED and an SMD phototransistor. Light emitted by the LED gets reflected back towards the phototransistors by objects in close proximity (< 30 mm) of the surface. Measuring the light intensity captured by each phototransistor allows for estimating the distance between surface and object. By using SMD components, Liu and Guimbretière (2012) achieved a spatial resolution of 10 dpi. However, resolution quickly declines with distance (no equation is given) as the LEDs have a wide emission angle of 140 degree.
The sensors are sampled at an update rate of 50 Hz using a micro-controller. While the ADC supports 10 bit resolution, all sensor data is scaled to 8 bit resolution due to noise issues[30].

Current draw is relatively high at 300 mA. This makes the current design unsuitable for most mobile applications. The authors suggest adding collimating lenses on top of the LEDs in order to increase range and decrease power requirements.

**Data Processing** Raw sensor data is preprocessed to linearize the sensor values. The authors discovered that the raw values are proportional to the inverse quartic function of the distance for greater distances, and proportional to the inverse quadratic function for closer distances. (This is to be expected as the light intensity of both emitted pulse and reflection each suffers from quadratic decay. At close distances, light from adjacent LEDs hits the reflecting object, too. Therefore, the inverse quadratic falloff applies only to the reflected light). Therefore, the quartic root of each raw value was used instead of the raw values. The authors did not attempt classifying individual grasp types but present several visualizations of the sensor data.

**Applications** Two FlexAura PCBs were wrapped around a pen and used for capturing various grasps. The paper contains several visualizations of the captured patterns. As the sensors are able to sense finger positions not only on the pen but also in close proximity, the authors suggest that FlexAura could also capture finger gestures.

**Relevant Findings** Figure 4 in the paper shows that a spatial resolution of at least 10 dpi is sufficient for clearly distinguishing individual fingers touching the surface.

[30] Shinwei Liu, personal communication, 03. February 2013

**Discussion** Overall, FlexAura is an incremental improvement of an established optical sensing technique. The operating principle is not new (Butler, Izadi, and Hodges 2008). However, Liu and Guimbretière (2012) present an approach that seems suitable for implementing grasp-sensing prototypes. As the power consumption is quite high, FlexAura is not suited for most battery-powered applications. While the authors call FlexAura a "flexible" sensor, it can only be bent along one axis. The authors do not cite Wimmer (2010a).

Strangely, the authors state in the paper that "different grips can result in a similar set of contact points" and "some of the most important information about grip comes from the shape of the inside of the hand" without citing any sources or providing examples.

## 4.4.18 Grip Force Authentication (NTT Docomo)

Iso et al. (2012) investigated how reliably grasp patterns can be used for identifying the user of a mobile phone. To this end, they instrumented a mobile phone with arrays of force sensors on its long sides. The authors extensively discuss their approach for user recognition but do provide little information about the grasp-sensitive surface. Unfortunately, the paper lacks important information about the implementation. Personal communication with the first author filled in some details but could not resolve all questions[31]. I do not completely understand what they were doing.

**Sensor Technology** The sensor arrangement is only described in little detail. A smartphone is equipped with four pressure sensor arrays, each consisting of 226 cells. Mode of operation and sensor layout are not reported. Figures 2 and 3 in the paper show only 133 discrete cells altogether. Each cell has a size of 2.5 mm x 2.5 mm. The cells are sampled at 100 Hz with 12 bit resolution. No further information about the implementation was available.

**Data Processing** The system authenticates users based on their grasp signature, using a "likelihood distribution of changes in grip force". It uses changes in sensor readings instead of absolute sensor readings as input values due to difficulties with stable calibration.

Iso et al. compare three different formats of sensor data: A (sensors independent), B (sensors form a single vector), and C (each of the four sensor arrays forms an individual vector).

Verifying the identity of a user is done in three steps: * preprocessing, * predicting changes in sensor data using a Kalman filter, * comparing this prediction to observed sensor data,

[31] Toshiki Iso, personal communication, 27.12.2012.

In short, the system tries to predict changes in a grasp signature based on the changes recorded during a training session. If the prediction matches the observed data for a certain span of time, the system assumes that training data and observed data are generated by the same person.

**Applications** The only application presented in the paper is identifying users. To quantify the performance of this approach, a user study was conducted. Data collection seems to have been conducted for about a month. In the user study, ten 10 participants were asked to browse emails for 60 seconds on the grasp-sensitive phone. During each task, 500 grasp signatures were captured (10 Hz x 50 s). Each participant generated three or eleven sets of pressure data (some participants did not have enough time to participate in the whole study). Cross-validation showed that data format A did not allow for reliable identification. Formats B and C both offered an EER of 10%. The authors calculate that samples from 30 s of usage are already sufficient for achieving an EER of 10%. However, it is not discussed whether this means capturing 300 grasp signatures or recording grasp-signatures for 30 s.

**General Findings** None.

**Discussion** The paper lacks important information about implementation of the prototype, data processing, and evaluation. For example, only results for a single study participant (ID 02) are reported in Figures 4 and 5 of the paper. The difference between data formats is not made clear. With the exception of (Kim et al. 2006), none of the existing research is referenced.

However, the authors present an interesting approach for classifying grasps using a Kalman filter for predicting sensor data based on training data, and then comparing the observed sensor data to predictions for each grasp type captured in the training session.

### 4.4.19 Grip UI (NTT Docomo)

Based on the research by Iso et al. (2012), two of the authors built a prototype of a grasp-sensing mobile phone that was presented at CEATEC Japan 2012[32]. Additional details and general thoughts on grasp-sensitive user interfaces were published in a technical report (Tsukamoto, Yuta, and Okada 2014).

The goal of this project was to support one-handed operation of mobile phones, for instance when "holding a child's hand" (Tsukamoto, Yuta, and Okada 2014). Sensor hardware is not described. However, the prototype is apparently force-sensitive along both

[32] A video of the demonstration was published at `http://www.diginfo.tv/v/12-0177-r-en.php`

long sides. The authors mention that the grasp-sensitive phone is only slightly wider and as thick as a regular phone. Sensor readings are made available to applications via Android's SensorManager framework.

Only one use case is presented in the demonstration and the paper: squeezing the phone in order to activate an application on the mobile phone. Depending on the vertical position of the squeezing fingers, different applications may be started.

Three relevant "grip features", i.e. characteristic properties of a grasp are defined: pressure intensity, pressure distribution, and pressure time transition. The authors argue that looking at temporal change in pressure allows for distinguishing between intentional and accidental changes in grasp pressure. This assumption is not supported in any way, however.

The authors also mention a trade-off between spatial resolution and sensor depth: smaller sensors allow for higher spatial resolution but offer lower resolution of grasp force.

The most interesting contribution of the technical report is a list of ten requirements for a grasp-sensitive user interface:

It must:

1) … be independent of user attributes such as age, gender, physique
2) … not change the shape of the object
3) … not conflict with other input methods
4) … work with existing applications without modifications
5) … support external applications
6) … allow the user to employ their preferred grasps
7) … be easy to train the system
8) … provide feedback to the user
9) … offer a response time lower or equal to other input methods
10) … must not negatively affect device operation

These requirements are sensible but not sufficient in my opinion, as discussed in Chapter 11 and Section 12.5.

### 4.4.20 28 Frames Later

Noor et al. (2014) use grasp-sensing to predict time and location of touch input on a mobile phone. Their key insight is that operating a mobile phone's touch screen with the thumb effects small shifts in the grasp employed to hold the phone. Training a machine learning classifier on these changes allows for predicting an approximate touch location before the fingertip actually makes contact with the touchscreen. This information may be used for reducing perceived latency.

**Sensor Technology** A Nokia N9 is augmented with flexible PCB that does not significantly alter its shape and weight. The PCB is wrapped around the left side, the back, and the right side of the phone. It contains 24 electrodes - 14 on the back, 4 on the right side, and 6 on the left side. These electrodes are connected to two AD7147 capacitive touch controllers which are sampled by the phone at 50 Hz via I2C. The raw data is offset-corrected and scaled from 16 bit to 8 bit per sample.

**Data Processing** For training the system, touch locations on the screen and their accompanying grasp signatures are recorded. Each captured grasp signature is treated as a 24-dimensional feature vector. In a first step, dimensionality is reduced to two dimensions via Principal Component Analysis (PCA).

Canonical Correlation Analysis (CCA) is used to find linear correlations between the two-dimensional grasp signature and the two-dimensional touch location.

For predicting touch locations based on grasp signatures, Gaussian Process Regression is employed. It is trained with the grasp signatures captured directly before the thumb made contact with the touch screen, combined with the later touch locations.

In order to predict not only touch location but also time of contact, the first-order derivative of the grasp signature is included in the feature vector.

**Applications** The authors suggest using their system for reducing perceived latency by triggering a system response before the user actually touches an on-screen button. The predictor was trained with grasp signatures from 20 users who contributed 250 samples for each hand. It is able to predict touch position with a precision of 18 mm (root-mean-square error, RMSE) 200 ms before contact. The closer to actual touch the prediction is made, the more precise it is. Time of contact can be estimated with 'reasonable' precision up to 500 ms before contact.

**General Findings** The authors suggest that the grasp dynamics for other tasks, e.g. typing, might be different, requiring per-task training data.

**Discussion** This research shows how implicitly provided grasp information can enhance interaction with mobile devices. While it is not clear how well the predictor would perform in real-world settings, initial results look promising.

Predicting the time of contact is very important for reducing latency. While the authors report how precise their prediction for time-of-contact is, the more interesting - and ultimately relevant - information would be how often the system correctly predicts the time of contact. Furthermore, the prediction should probably be tuned to rather be late than early. Reacting to a touch before the user has actually physically touched the screen might feel awkward, like a dialogue partner answering questions you did not yet ask. Ideally, the estimate for the time of contact could be continuously updated, e.g. by using a Kalman filter.

### 4.4.21 Sensing Techniques for Tablet+Stylus Interaction

Hinckley et al. (2014) present an observational study investigating how users grasp a digital pen while interacting with a tablet computer, a pen and a tablet equipped with grasp-sensitive surfaces, and several applications that benefit from the sensor data. The results of the observational study are described in Section 2.4.

**Sensor Technology** The prototype consists of a digital pen and a Windows 8 tablet. Both are equipped with custom-built grasp-sensitive surfaces.

A custom-printed 7 x 30 sensel capacitive sensor matrix is wrapped around the pen's barrel (190 x 14 mm) and read out by a Cypress CY8CTMA463-60BUI touchscreen controller. While the sensor resolution is not documented, it seems to be 8 bit/sensel. The sensor has an update rate of 30 Hz. The pen also contains a gyroscope and an accelerometer. Sensor data is transmitted wirelessly.

The tablet's back and sides are covered by 44 x 26 capacitive sensors. The update rate is 25 Hz. Sensor technology and resolution are not reported. The tablet also contains an accelerometer, a gyroscope, and a magnetometer.

**Data Processing** Similar to the earlier prototype by this group (Song et al. 2011), the sensor data from the cylindrical sensor surface is not transformed into a rotation-invariant representation. Instead, the grasp signature is normalized by translating and scaling the matrix of raw values. Classification features are stylus yaw and pitch, normalized grasp signature and histograms, the number of covered sensels, and the sum of all capacitive sensor values.

The pen prototype is able to distinguish four different grasp types, called *writing*, *tuck*, *palm*, and *no grip*. Sensor data is classified by an SVM which was trained with 32400 samples from nine right-handed users. While the authors emphasize that the users were asked to also conduct transitional movements between different grasp types, it is not reported how these were annotated or whether there was any filtering - such as an hysteresis - to stabilize the classifier's output.

A user study with nine additional users showed a recognition accuracy of 93% each for *writing* and *tuck* grasps but only 77% for *palm* grasps. The lower performance of the latter is explained as a result of the users not grasping the pen strong enough in this state.

Sensor data from the tablet is not used for grasp sensing but for palm rejection and similar tasks.

**Applications** The paper encompasses a large number of interaction techniques that use the recognized grasp types. These include a virtual pen barrel button, detection of unintentional touches on the tablet's screen, or switching between different virtual tools. In most cases, data from multiple sensors is used to accurately guess the user's actions and intentions.

**General Findings** The authors describe several interesting observations and insights from a user study. These are described in Section 2.4.

**Discussion** This paper is one of very few[33] that encompass observations, novel sensor hardware, and useful interaction techniques. While some important information on the grasp sensing approach is missing from the paper, and while the authors do not discuss rotationally-invariant grasp signatures, this paper is very well rounded. Compared to the Song et al. (2011), the hardware was slightly improved.

## 4.5 Summary

In this chapter I presented a comprehensive overview of grasp sensing techniques, with a strong focus on grasp-sensitive surfaces, i.e. sensors that can be embedded in tangible objects.

Comparing and analyzing related work leads to three main conclusions that are discussed in the remainder of this chapter. These are still as valid in 2014 as they were when I started my research in 2008:

(1) Grasp-sensitive surfaces extend or enable interaction with tangible objects in a multitude of ways. For Human-Computer Interaction, it is preferable to instrument objects with grasp-sensing capabilities instead of users or environments.
(2) Researchers are often unaware of existing research on grasping and grasp-sensitive surfaces. Therefore, researchers sometimes repeat the same errors and re-invent the same wheels.
(3) Research on grasp-sensing objects involved and involves significant effort in hardware development and signal analysis. Current applications for grasp-sensitive

[33] Actually the only one as of April 2015.

surfaces are mostly simple proof-of-concept implementations; little thought is directed towards integration with other modalities or real-world complexity. This is also caused by a lack of good prototyping tools for grasp-sensitive surfaces.

### 4.5.1 Benefits and Applications of Grasp-Sensitive Surfaces

As the diverse projects presented in this chapter show, grasp sensing may enhance existing and enable novel interaction techniques.

Grasp-sensing artifacts may be used as 3D input devices (Pai et al. 2005), switch between modes and trigger actions upon being grasped in certain way (Kim et al. 2006; Taylor and Bove 2009; Tsukamoto, Yuta, and Okada 2014), implicitly support interaction without the user necessarily noticing that the object senses how they grasp (Harrison et al. 1998; Noor et al. 2014), allow for implicit authentication (Veldhuis et al. 2004; Iso et al. 2012), and extend the vocabulary of existing input techniques (Song et al. 2011).

For research in human-computer interaction and most practical applications, it is useful - and more prevalent - to instrument the object with grasp-sensing capabilities than to instrument environment or user.

Grasp-sensitive surfaces have unique functional, technical and commercial advantages and limitations compared to the other two approaches. The main advantages include:

- Grasp-sensitive surfaces are the best way to capture the contact areas between hand and object. This allows sensing the function of the grasp, instead of its form, which is desirable for grasp interaction.
- Sensors embedded into a device are less obtrusive than sensors attached to the user or the environment.
- Oftentimes grasp sensing is used to enhance interaction with electronic gadgets or tools, such as mobile phones, input devices, or power tools. Integrating grasp sensing, processing and reaction into one device is arguably the most robust and efficient approach.
- Integrating grasp sensing into an interactive object allows the manufacturer complete control over the user experience. It also overcomes the chicken-egg problem where lack of external sensing hardware prevents developers from creating applications for that hardware - and vice versa.

Limitations of embedding grasp sensing into objects - compared to grasp sensors attached to users or the environment - include:

- Grasp-sensitive surfaces can only capture grasp signatures, i.e. contact points/areas and occasionally proximity. Information about hand posture during pre-grasp and grasp is not available. In most cases, grasp signatures also do not provide information about the user.

- Grasp-sensing hardware increases the device's footprint. Therefore, it is hard and expensive to make tiny objects grasp-sensitive.
- Every single grasp-sensitive object needs to be equipped with sensors. This leads to duplication of functionality, increases the manufacturing costs, and may make grasp-sensing economically inviable for cheap objects.

Overall, it seems sensible to instrument users and environments for researching human grasping and to instrument objects for researching grasp interaction. As shown by Holz and Baudisch (2011), high-quality data about touch input, captured by specialized equipment, can help in improving our understanding of touch input. Grasp information captured by body-worn or environmental sensors may also be used to provide a baseline for accuracy and precision of grasp-sensitive surfaces.

In summary, instrumenting the user is necessary only in a few cases, like in an analysis of the grasp process. Grasp-sensitive surfaces can only capture the result of a grasp process, and therefore provide less information. However, the information that a grasp-sensitive surface delivers can be of equal or better quality. As most users might arguably not like being instrumented, grasp-sensing objects that work without any external instrumentation will be commercially more successful in the immediate future.

### 4.5.2 Lack of Knowledge of Previous Research

As already mentioned in the previous chapters, research on human grasping is conducted in several research areas. However, oftentimes researchers are apparently not familiar with relevant related work from neighboring disciplines. This is also true for HCI research on grasp-sensing user interfaces. Rarely researchers reference relevant related work from other research disciplines. This leads to researchers ignoring important aspects, re-inventing the wheel, or devising worse solutions for previously-solved challenges:

For example, Veldhuis et al. have conducted extensive research on grasp sensing and have first documented variability of grasps by the same users. Despite this, nearly no longtime studies have been conducted with grasp-sensitive objects. Veldhuis et al.'s findings have been ignored by most HCI researchers - probably because they have not been published or cited in HCI journals or conference proceedings. Time and again, HCI researchers are surprised by the diversity of grasps employed by study participants (see e.g., Taylor and Bove (2009)), despite early research on human grasping predicting this.

Sometimes research is unnecessarily duplicated because the authors are not aware of previous research. Marshall et al. (2008) were not aware of earlier, extensive research by Mascaro, Chang, and Asada (1999). Taylor and Bove (2009) essentially replicated research published more than two years earlier by Kim et al. (2006) without building on their experiences. As mentioned in Section 2.4, a student and I conducted an unpublished study on grasp variability that was accidentally nearly identical to a study conducted 14 years earlier. Unfortunately, we were not aware of this study until one day before the submission deadline of our paper (which we then decided not to submit).

Finally, ingenious solutions are forgotten and replaced by worse solutions. For example, Song et al. (2011) were not aware of previous work on rotationally invariant grasp classification (Kry and Pai 2006a) and therefore developed their own - less versatile - algorithm.

Such a lack of knowledge about relevant research in other disciplines is definitively not unique to grasp interaction or HCI in general. However, as grasp interaction needs to draw upon knowledge from a multitude of research areas - anatomy, sensor design, signal processing, machine learning, and UI design - gaps in knowledge are especially harmful.

### 4.5.3 Prototyping is Still Too Hard

Most prototypes presented in this chapter use custom-built grasp-sensitive surfaces. Only Veldhuis' Smart Gun Grip and the steering wheel employ off-the-shelf sensor mats. Goel, Wobbrock, and Patel (2012) use the internal sensors of a mobile phone.

Five prototypes use custom flexible PCBs (Kim et al. 2006; Song et al. 2011; Liu and Guimbretière 2012; Tsukamoto, Yuta, and Okada 2014; Noor et al. 2014).
Four prototypes use electrodes made of custom-cut copper foil (Pai et al. 2005; Taylor and Bove 2009; Cheng et al. 2013; Iso et al. 2012).
Seven prototypes use custom sensor circuits ranging from simple force-sensitive resistors to complex capacitive sensing systems (Harrison et al. 1998; Hinckley and Sinclair 1999; Hinckley et al. 2000; Mäntyjärvi et al. 2004; Miyaki and Rekimoto 2009; Sato, Poupyrev, and Harrison 2012; L.-P. Cheng et al. 2012a).

In recent years, advances in rapid manufacturing technologies have made it much easier and cheaper to order custom flexible and rigid PCBs online or print them at home. However, designing and assembling working sensor hardware is still out of reach for most researchers.

---

Overall, this review of related work shows that grasp sensing may enhance existing input techniques and enable novel interaction techniques. However, researchers need to be more aware of existing research in adjacent disciplines. Furthermore, building grasp-sensing objects requires significant effort and knowledge in electrical engineering.

In order to facilitate development and discussion of grasp-sensitive user interfaces, I developed three techniques for prototyping grasp-sensitive surfaces which are presented in the following chapters.

# II

# Novel Techniques for Prototyping Grasp-Sensitive Surfaces

As discussed in the previous chapter, most grasp-sensitive surfaces used in research prototypes are custom-built. Design and implementation of these sensors requires in-depth knowledge of analog electronic circuits and access to specialized manufacturing tools or materials, such as flexible circuit boards (Kim et al. 2006) or ITS substrates (Taylor and Bove 2009). In addition, many sensors have custom shapes, making it hard to adapt them to other form factors. This makes iterative prototyping of grasp-sensing objects tedious and impedes exploratory design. For researchers, the time and effort spent on custom grasp-sensing circuitry reduces the available time for research on human factors and user interfaces.

In the following chapters, I present three novel approaches for implementing grasp-sensitive surfaces which I developed. These have some unique properties which distinguish them from most other sensing techniques. Most notably, these techniques are suitable for prototyping grasp-sensitive artifacts with only minimal knowledge of electronics. They have been developed from 2008 to 2011 and informed my further research on grasp sensing.

**CapToolKit and HandSense** Traditional (capacitive or resistive) sensor substrates are not well suited for prototyping, as they are hard to modify incrementally. Employing a small number of very sensitive sensors instead of large, less sensitive sensor substrates dramatically eases prototyping and allows for reconfiguration and reuse of the sensors. As no commercially available or well-documented toolkit for capacitive sensing existed, I developed *CapToolKit* in 2006. CapToolKit consists of client software and a custom controller board which can read out up to eight custom capacitive sensors. The high resolution of these sensors allows for robust grasp sensing even with only few sensors.

*HandSense*, a grasp-sensing prototype with four capacitive sensors allows distinguishing several different grasp types. As CapToolKit is open-source, and was commercially available for some time, HandSense makes prototyping grasp-sensitive objects simple and affordable. In addition, HandSense does not only sense touch but also proximity and thickness of touching tissue. Grasp recognition is done with heuristics; a hybrid machine-learning approach was also implemented. In a user study, we demonstrated that a few high-resolution sensors and appropriate heuristics can be used to reliably distinguish a number of different grasp types with an accuracy comparable to much more sophisticated systems.

**FlyEye** When prototyping grasp-sensitive objects, it is often not known in the beginning, how users will actually grasp the object. Therefore, one also does not know where sensors should be placed. The most flexible approach would be to augment the whole surface of the object with high-resolution grasp sensing.

*FlyEye* enables designers to make arbitrary surfaces grasp-sensitive without the need for custom sensors or soldering. To this end, *FlyEye* uses a camera sensor for capturing contact on the surface of an object. The camera is placed inside the object. Many individual optical fibers are embedded into the surface of the object and connected to the camera. A calibration algorithm associates pixels in the camera's image with positions on the surface. While FlyEye allows grasp-sensing on arbitrary shaped surfaces, without knowledge about electrical engineering, its usefulness is limited in practice. Augmenting large surfaces requires manual placement of hundreds or thousands of fibers. Additionally, fibers and camera occupy space inside the object, making it impossible to make solid objects grasp-sensitive.

**TDRtouch** The optimal prototyping tool for grasp-sensitive surfaces would be a touch-sensitive structure that is capable of covering arbitrary concave and convex shapes while requiring minimal circuitry. These requirements are met by some kind of string or cable that one can wrap around an object.

*TDRtouch*, the third technique presented in this dissertation, employs Time Domain Reflectometry (TDR) for tracking multiple touches on a cable. A TDR device sends short pulses into a cable. These are partially reflected back wherever a finger touches the cable. The runtime of the reflections can be used to calculate the position of each touch. Arbitrary objects can be made grasp-sensitive by wrapping a cable around them and connecting it to a TDR device.

In the following chapters, I describe principles, properties, and applications of these three techniques, and compare them to each other and other sensing techniques.

Chapter 5

# CapToolKit and HandSense:

## Grasp Sensing Using Only Few Capacitive Sensors


**Figure 5.1:** The HandSense prototype employs four capacitive proximity sensors - two on each long side. Using heuristics for grasp classification, it allows distinguishing between six different left-handed and right-handed grasps with an average accuracy of 81%.

*HandSense is a research prototype that distinguishes between different one-handed grasp types using heuristics. Four high-resolution capacitive sensors are embedded into a box of about the size of a mobile phone. HandSense is also able to distinguish between left-handed and right-handed grasps. A user study showed an average recognition rate of 81 % for six grasp types, whereby the recognition rate for some users was above 90%. This level of accuracy is similar to those of prototypes employing machine learning and high-density sensor grids.*

**Attribution:** This chapter is based primarily on the paper "HandSense - Discriminating Different Ways of Grasping and Holding a Tangible User Interface" (Wimmer and Boring 2009). For this paper, my co-author Sebastian Boring was involved in planning the research project, analyzing data from the user study, and writing the paper. I was solely responsible for concept, implementation, and for conducting the user study. This chapter also documents further research conducted after publishing the paper. This research into using machine learning for grasp classification was rather informal and has not yet been published elsewhere. For this chapter, I have completely re-analyzed the logs of the original study.

## 5.1 Motivation

Previous and parallel research by other authors (see Chapter 4.4) focused on grasp-sensitive surfaces with a high sensor density but low resolution of each individual sensor (see also Table 5.3).

While this approach allows for accurate recognition of grasps, having a sensing grid with dozens or hundreds of sensors seems excessive for recognizing a small number of basic grasps. For example, Taylor and Bove (Taylor and Bove 2009) recognize five different grasps using 72 individual sensors. Kim et al. (Kim et al. 2006) recognize eight different grasps using 64 individual sensors.

Covering the entire surface of an object with sensors has several drawbacks:

- Less surface area remains available for displays, buttons, or connectors.
- Covering the whole surface with conductive traces can impede wireless signal transmission and reception, which is especially problematic for mobile devices.
- Sensor grids need to be adapted to the shape and curvature of the surface. This requirement increases engineering time and cost. It also makes iterative prototyping tedious.

With HandSense (Figure 5.1), we wanted to find out how well grasp recognition with only a few sensors could work. In addition, we decided to design heuristics for distinguishing between different grasp types. Most other grasp-sensitive prototypes employ machine-learning classifiers for recognizing grasps. As we knew little about machine learning, we wanted to find out how well heuristics fare compared to machine-learning approaches.

We chose *handling a mobile phone* as an exemplary use case that might be enhanced by grasp sensing. Mobile phones are ubiquitous user interfaces that are grasped often and in several different ways. Additionally, their box-shape allows straightforward reasoning about recognizable grasps and eases implementing a prototype. We built a prototype with four capacitive sensors and defined nine different grasp types to be recognized. These 'grasp types' also include states where no hand touches the object: "lying on table" and "in pocket". The sensor readings are analyzed in real time using a set of heuristics. In order to quantify the recognition accuracy of our prototype, we conducted a user study. Furthermore, I conducted some additional, informal research into using a machine learning classifier for distinguishing different grasps.

The remainder of this chapter is organized as follows:

- Section 5.2 gives a short overview of *CapToolKit*, the capacitive sensing hardware employed in HandSense.
- Section 5.3 introduces the hardware of the HandSense prototype.
- Section 5.4 presents a use case for grasp-sensing mobile devices and the grasp types to be distinguished by HandSense.
- Section 5.5 presents the heuristic classifier employed in HandSense and a user study that evaluated its performance.
- Section 5.6 shortly discusses initial research into a machine-learning algorithm for HandSense that has not been continued.
- Section 5.7 contains a comparison of HandSense and other grasp-sensing research prototypes.
- Section 5.8 concludes this chapter with a critical discussion of the approach, the prototype, and the user study.

## 5.2 CapToolKit

Most grasp-sensing prototypes employ custom-designed, proprietary sensor grids. These have custom shapes, custom controllers, and custom wiring. This makes iterative prototyping of grasp-sensing artifacts unnecessarily tedious. Additionally, custom hardware impedes independent replication of study results and improvements to filter and classification algorithms.

Therefore, we built upon CapToolKit (Wimmer et al. 2007; Wimmer 2011c) for capturing grasps, a capacitive sensing toolkit that I initially developed in my *Diplom* thesis and improved during my PhD research. This open-source toolkit consists of small sensors, a controller board, firmware, and visualization software (Figure 5.2).

a) b)

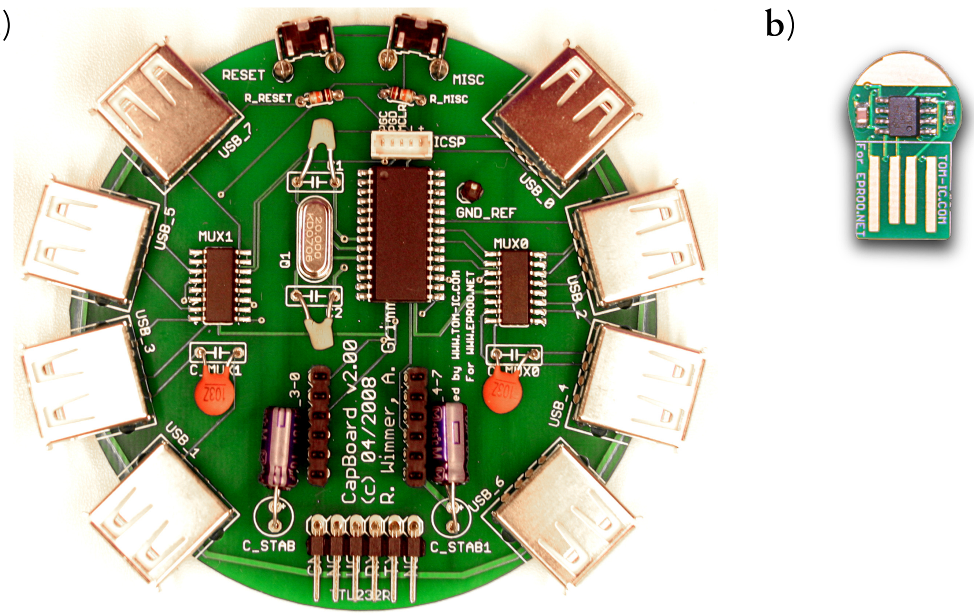

**Figure 5.2:** CapToolKit consists of hardware, firmware, and software that allows rapid prototyping of capacitive sensing applications. The CapBoard (a) manages up to eight simple but precise capacitive sensors (b), filters sensor readings, and transmits them to a host computer via USB.

Capacitive sensing (Baxter 1997) describes a group of sensing techniques that detect changes in the environment by measuring capacitance changes. It can be used for measuring fuel level in tanks, microscopic displacements in MEMS sensors, or touch and proximity detection.

Generally, a capacitive sensor measures the capacitance between a sensor antenna (sometimes called *sensor electrode*) and its environment. Conductive objects close to the antenna increase the capacitance measured by the sensor.[1] The closer the object is to the sensor, the higher the measured capacitance.

Capacitive sensing is a technique very well suited for touch and grasp sensing, as it is very sensitive to human touch but not to non-conductive materials. Therefore, capacitive sensors can be embedded inside plastic casings or behind screens. However, other conductive objects, such as metal casings, or electronic components, also cause an increase in capacitance. This means that capacitive sensors can not distinguish between a finger and a piece of metal touching the antenna. Additionally, such conductive objects also decrease sensor resolution, as they partially saturate the sensor. In order to shield the antenna from close-by electronics and other objects that could taint measurements,

[1] This is a strong simplification for conciseness' sake. In-depth descriptions of the properties and working principles of capacitive sensing can be found in the literature referenced in this chapter.

some capacitive sensors (such as CapToolKit) employ a *guard electrode*, active shielding that partially covers the antenna and decreases sensitivity in all directions that are covered by it.

A good overview of capacitive sensing techniques for human-computer interaction can be found in Joshua R. Smith's PhD thesis (1999) and several publications he co-authored (Zimmerman et al. 1995; Smith, White, and Dodge 1998).

Unlike most other capacitive sensor designs, CapToolKit employs simple sensors that convert capacitance into a frequency-modulated electrical square wave signal with a frequency of 1 - 2 MHz. These sensors are attached directly to the antenna and connected to the controller board (*CapBoard*) via common and cheap USB cables. As the conversion happens close to the antenna, sensors may be placed up to several meters away from the controller board without signal degradation.

CapToolKit has not been developed further since 2010. However, recently, development on OpenCapSense (Große-Puppendahl et al. 2013) has started, an new open-source toolkit for capacitive sensing whose design is inspired in large parts by CapToolKit.

## 5.3 The HandSense Prototype

To maximize the number of different grasps to be recognized, sensors have to be placed at locations that offer as much information about each grasp as possible; each of the grasps needs to cover a unique combination of sensors.

For HandSense, we decided to employ four sensors, as this number allows for symmetrically distributing sensors within the object and supports recognizing a reasonable number of different grasps. Symmetrical arrangement of sensors allows for recognizing both left-handed and right-handed grasps. There are two symmetrical layouts of four sensors on a rectangular area: axially symmetrical (Figure 5.3a) and axially/rotationally symmetrical (Figure 5.3b). We chose the axially symmetrical sensor layout (Figure 5.3a) because it places the sensors at the long sides - which are often grasped during interaction with a mobile phone.

In theory, four binary touch sensors allow distinguishing between 15 different grasp patterns and a non-grasping state ($2^4 = 16$). In practice, eight of these grasps patterns are not functionally effective, that is, they are not suitable for actually holding the object (Figure 5.4). The other symmetrical sensor layout (Figure 5.3b) is limited to the same number of functionally effective different grasps. Therefore, it does not have an advantage over the layout used for HandSense.

Of the remaining eight recognizable patterns, two are rarely used for grasping mobile phones (Figure 5.4). However, using sensors with more than one bit resolution allows for distinguishing different grasps that share the same combination of sensors. This allows us to extend the number of recognizable grasps.

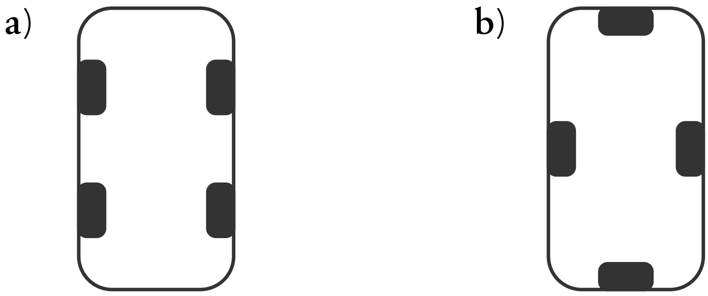


**Figure 5.3:** Four sensors can be distributed on sides of a box either in an axially symmetrical (a) or rotationally symmetrical layout (b).

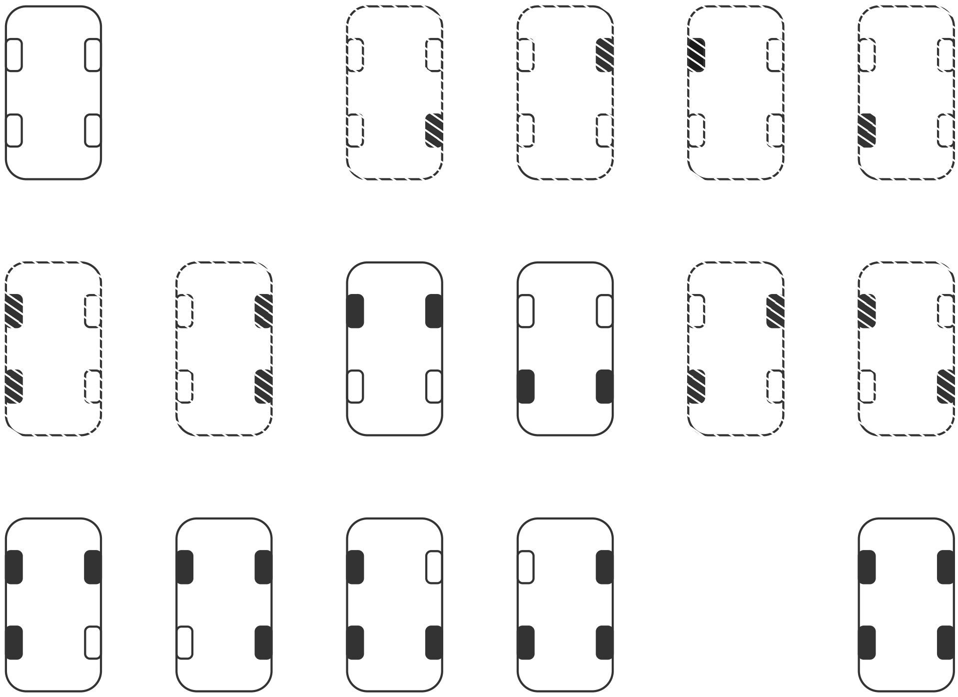

**Figure 5.4:** In theory, it is possible to distinguish up to 16 different grasp patterns using only four binary sensors. However, only eight of these (highlighted) are functionally effective and allow securely holding a box-shaped object. Other shapes and sensor distributions are limited to the same number.

The HandSense prototype (Figure 5.1) is a box about the size of a mobile phone (100 mm x 50 mm x 25 mm). It is equipped with four CapToolKit sensors, each 30 mm x 15 mm in size. These sensors are located in the top left, top right, bottom left, and bottom right of the long sides.

These are connected to the CapBoard which is in turn connected to a host computer. The CapBoard transmits measurements from all four sensors to the computer at an update rate of 25 Hz. As the four sensors operate in time-multiplex mode, no two sensors are active at the same time. This also means that the measurements used for recognizing a grasp are taken at different points within the 40 ms time frame. As HandSense is intended to only capture static grasps, this slight time offset between samples is not an issue in practice.

All sensor data is normalized to a range from 0.0 to 1.0. To this end, minimum sensor readings are recorded when no hand touches the prototype. Maximum sensor readings are recorded by very tightly holding the prototype between both hands. Due to inherent sensor drift and changing environmental properties, this calibration procedure should be performed once per session.

## 5.4 Use Cases and Grasp Types

While related research focused on biometric authentication (Veldhuis et al. 2004), direct manipulation (Kry and Pai 2006a), or switching between different modes/applications (Kim et al. 2006; Taylor and Bove 2009), we were interested in basic grasp primitives for interaction with a mobile device. Within the restrictions of the sensor layout (Figure 5.4), we identified a minimum set of basic states and actions which are essential for everyday mobile phone use. They are described in the following scenario. For the sake of simplicity, we focused on single-handed grasps.

> Max is carrying his mobile phone *in his pocket*. When it starts ringing, he *pulls it out*, *holds it in his right hand*, and answers the call. Afterwards, Max puts the phone down onto his desk. While the phone *lies on the desk*, it starts ringing again. Max *picks it up* with his left hand and takes this call.

These states, *in pocket*, *pull out*, *hold in hand*, *on table*, and *pick up* presumably cover a significant percentage of everyday grasp interaction with mobile phones.

Subsequently, we mapped these actions to grasp types, as shown in Figure 5.5.

For the pick-up and holding grasps, we wanted to recognize both left-handed and right-handed grasps. As a mobile phone can be in a pocket either top-up or top-down, we also decided to map the pull-out action to *grasp top* and *grasp bottom* grasps. For the sake of

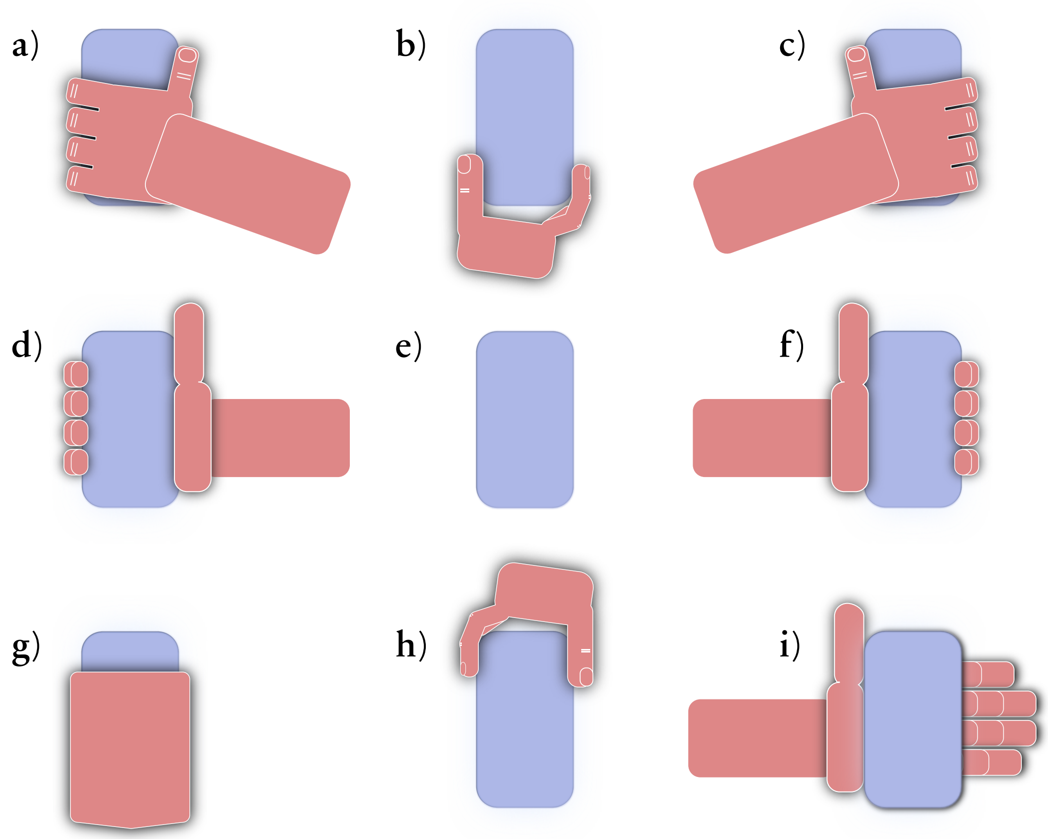


**Figure 5.5:** Grasp types to be distinguished using the HandSense prototype. In addition to traditional grasp types, HandSense also distinguishes *on table* (e), *in pocket* (g), and *on hand* (i).

simplicity, two states - *in pocket* and *on table* - that are not actually grasps are also referred to as *grasp types*.

The grasp types to be distinguished are therefore (Figure 5.5):

a) *Grasp Left* - device grasped with fingers of left hand, e.g. for picking it up
b) *Grasp Bottom* - device grasped on bottom end, e.g. for pulling it out of a pocket
c) *Grasp Right* - device grasped with fingers of right hand, e.g. for picking it up
d) *Hold Left* - device held in left hand between palm and fingers
e) *On Table* - device lying on table, not touched
f) *Hold Right* - device held in right hand between palm and fingers
g) *In Pocket* - device in pocket, not touched
h) *Grasp Top* - device grasped on top end, e.g. for pulling it out of a pocket
i) *On Hand* - device on hand, not grasped

It should be noted that these grasp types were not mapped to certain modes or actions. Instead, the different grasp types are context information that can be used together with other sensor data for determining a user's interaction context.

However, the scenario offers some suggestions on how grasp sensing could enhance interaction with a mobile phone. For example, a mobile phone that knows that it is in a pocket, i.e., touching the user, can switch to silent mode and only vibrate on calls. Whereas, when the phone detects that it is lying on the table, it might activate the ring tone again, as the user might not recognize it vibrating. Additionally, the phone, could stop ringing once the user touches
Touching the phone in the pocket might also be used for interacting with it, e.g. by knocking certain patterns (Hudson et al. 2010) or drawing commands on the device's surface (Saponas, Harrison, and Benko 2011).

# 5.5 Grasp Recognition Using Heuristics

The sensor data captured by the HandSense prototype was used for classifying different types of grasps. The first classifier that was implemented is based on a set of heuristics. These rules were developed and refined iteratively to maximize recognition rate.

## 5.5.1 Mapping Raw Data to Grasp Types

We first defined five different states that each sensor could report:

- *No Proximity* (0% - 3% of maximum sensor reading): no body part near the sensor
- *Near* (3% - 20%): body part near sensor but not touching it (i.e. proximity).
- *On Hand* (20% - 50%): body part very close to sensor but not touching it
- *Gripped* (50% - 85%): body part partially covers sensor.
- *Held* (0.85% - 100%): body part completely covers sensor.

These five states allow us to theoretically discriminate $5^4 = 625$ different grasp states. However, only a subset of these correspond to actually occurring grasps. Our rule set for guessing the type of grasp is described in Table 5.1

When picking up the prototype or switching between grasps, sensor readings change drastically and quickly. To avoid false classifications in these cases, only static grasps are reported. A grasp is assumed to be static if the same grasp has been recognized five times in a row. As long as the classifier does not recognize a static grasp, it reports an *unsure* state. On average, the classifier took about 600-700 ms to settle on a grasp type.

| Grasp Type | Top Left | Bottom Left | Top Right | Bottom Right |
|---|---|---|---|---|
| On Table | No Prox. | No Prox. | No Prox. | No Prox. |
| In Pocket | Near | Near | Near | Near |
| On Hand | On Hand | On Hand | On Hand | On Hand |
| Grasp Top | Gripped | < Gripped | Gripped | < Gripped |
| Grasp Bottom | < Gripped | Gripped | < Gripped | Gripped |
| Grasp Left | Gripped | Gripped | Gripped | < Gripped |
| Hold Left | Held | Held | Held | Held |
| Grasp Right | Gripped | < Gripped | Gripped | Gripped |
| Hold Right | Held | Held | Held | Held |

Table 5.1: Heuristics for guessing the grasp type based on the states of the four sensors in the HandSense prototype. *Hold Left* and *Hold Right* are distinguished in a second step, described in Section 5.5.2

### 5.5.2 Handedness Recognition

One interesting advantage of using very sensitive capacitive sensors is their ability to measure the thickness of the body part touching them. For example, fingers touching the surface supply less capacitance than the ball of the hand.

As we found out during initial prototyping, this allows distinguishing between left-handed and right-handed grasps, even if all four sensors are completely covered by palm and fingers. Such a closed grasp would saturate sensors with low resolution. The highly sensitive sensors within HandSense return lower readings when fingers are covering them than when the ball of the hand is covering them. Thus, HandSense recognizes whether the left or the right side has higher sensor readings and thus classifies a grasp as either left-handed or right-handed (Figure 5.6).

### 5.5.3 Evaluation Setup

To evaluate how well our system performed for different users, we conducted a small user study.

The goal of the study was to determine the error rate and detection speed of our heuristic classifier, as well as finding out how we could improve the sensor layout.


**Figure 5.6:** HandSense is capable of distinguishing between left-handed and right-handed grasps, even if all sensors are completely covered. The high sensitivity of the capacitive sensors allows to infer the thickness of the tissue in contact with the sensor, thus distinguishing between fingers and palm.

Due to a lack of time, we recruited only four volunteers among colleagues and visitors to our lab. Due to a lack of additional volunteers, both authors took part in the study, too. This was not mentioned in the original paper but should have been avoided or at least disclosed. Effects of this decision on the study results are discussed in the following section. The one female and five male participants were between 26 and 28 years old (average: 27.2 years). Two participants were left-handed, the remaining four were right-handed. Except for the two authors, no participant had prior experience with grasp sensing or the HandSense prototype.

We assume grasp recognition to be used primarily within personal devices and tools, such as mobile phone. In these usage contexts, recognition performance during regular use is more informative than the performance on first-time use. Therefore, we opted to give our participants a short introduction to the prototype and let them try out the prototype in a training round before the trial started. We also assumed that this would limit learning effects in the trial. The training round consisted of each participant trying out each of the six grasp types we wanted to distinguish (grasp {top, bottom, left, right}, and hold {left, right}) five times.

The subsequent test round comprised 10 trials for each of the six grasps to be recognized, resulting in 60 randomized trials. At the beginning of the test round, the participant had to pick up the prototype from a table. During each trial, the participant was shown an icon depicting one of the six grasps (Figure 5.5). The participant had to hold the prototype in the designated grasp, until the classifier had recognized a grasp type. If the classifier could not settle on a grasp type within three seconds, the grasp was manually marked as not recognized. In case of a timeout, correct, or incorrect classification, the next grasp to be performed was displayed. In order to achieve realistic conditions for grasp sensing, participants were not allowed to put down the prototype between trials but had to migrate from one grasp to the next while holding it. This requirement probably reduced the observed recognition rate.

Due to occasional hardware and software errors, some participants conducted a few trials more than the others. Altogether, instead of 360 trials (6 tasks/user * 6 users * 10 trials per task), only 359 trials were conducted and analyzed.

### 5.5.4 Results and Discussion

For the original paper (Wimmer and Boring 2009), we calculated average error rates and standard deviations for each grasp, combining results of all participants. For this dissertation, I have re-analyzed the data. In the following, average recognition rates (i.e., 100% - *error rate*), individual recognition rates for each participant, and a confusion matrix are reported.

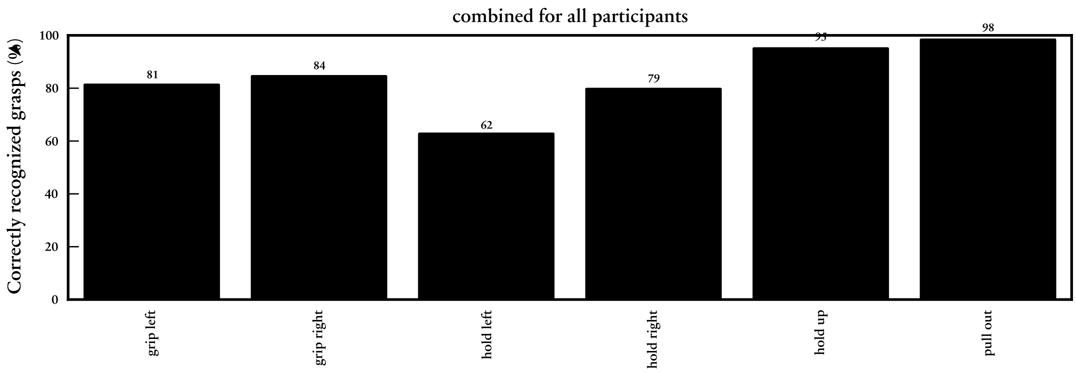


**Figure 5.7:** Recognition accuracy as reported in (Wimmer and Boring 2009) (instead of error rate, recognition rate is reported)

On average, HandSense correctly recognized the employed grasp in 81% of all trials. As Figure 5.7 shows, recognition rates for the different grasp types varied notably. Recognition rates for the two-finger grasps - *pull out* and *hold up* were fairly high (98% / 95%). *Grip left* and *grip right* were correctly recognized in 81% / 85% of all cases. The *hold left* and *hold right* grasps had the worst recognition rates of 62% / 79%.

Analyzing the log data for each participant separately offers deeper insights (Figure 5.8).

As mentioned above, both authors of the original paper - Sebastian Boring and I - participated in the study. This was originally done to collect more diverse data, not to 'improve' the study results. However, recognition rates for both of us were indeed better than those for any other participant - 93% (RW) resp. 96% (SB) on average. One might assume that the authors of a paper are generally more motivated than the other participants to correctly grasp the device. However, as the other participants had been recruited from our lab, I assume that they were very motivated, too.

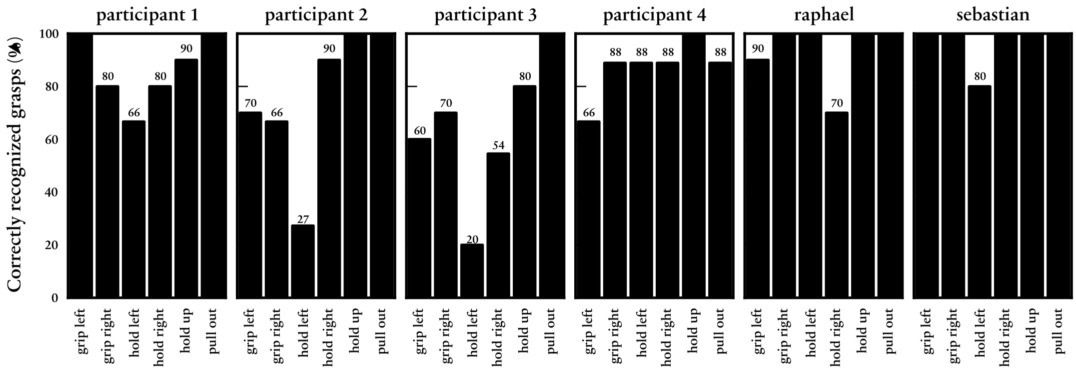


**Figure 5.8:** Recognition accuracy for each of the six participants in the study. Grasps by both authors, Raphael Wimmer and Sebastian Boring, were recognized better than those of other participants.

It is more likely that the system performed especially well in our trials because it was inadvertently optimized for recognizing our grasps. As I used my grasps as reference when developing the heuristics for grasp recognition, it is not surprising that these worked quite well in my case. In addition, I had worked on the prototype for some time, so that I might have subconsciously learned how to hold it in order to achieve good grasp recognition. My co-author Sebastian did not handle the prototype before the study. However, his hand size and shape are similar to mine - which may explain the good performance of the heuristics in his case. Looking at the individual recognition rates (Figure 5.8), it seems likely that simple heuristics which have been tuned for a particular person are able to correctly classify a small number of distinct grasps in more than 90% of all cases.

Summarizing the log data in a confusion matrix (Table 5.2) helps in better understanding the low recognition rate for *hold left* and *hold right* grasps.

For example, the confusion matrix shows that the *unsure* state makes up the majority of incorrect classifications.

| recognized as -> | grip left | grip right | hold left | hold right | hold up | pull out | no grip | un-sure |
|---|---|---|---|---|---|---|---|---|
| grip left | 52 | 1 | 0 | 3 | 1 | 1 | 1 | 5 |

| recognized as -> | grip left | grip right | hold left | hold right | hold up | pull out | no grip | un-sure |
|---|---|---|---|---|---|---|---|---|
| grip right | 1 | 49 | 1 | 1 | 0 | 0 | 1 | 5 |
| hold left | 2 | 0 | 37 | 8 | 1 | 0 | 1 | 10 |
| hold right | 1 | 0 | 1 | 47 | 2 | 0 | 0 | 8 |
| hold up | 0 | 0 | 1 | 0 | 57 | 1 | 0 | 1 |
| pull out | 0 | 0 | 0 | 0 | 0 | 58 | 1 | 0 |

Table 5.2: Confusion matrix for the HandSense evaluation. The first column describes the grasp employed by participants, the following columns show how often it was classified as a certain grasp.

When calculating the recognition rate, an *unsure* state was simply counted as an error. In the confusion matrix, those *unsure* states are reported as explicit recognition states. This allows distinguishing between cases of 'truly' incorrect recognition and cases where the algorithm determined that the data was not sufficiently clear to infer which grasp was being applied.

As can be also seen in the confusion matrix, *hold left* gets often incorrectly classified as *hold right*, but not the other way round. Rarely are both recognized as any of the other grasp types. This suggests that the thresholds used for distinguishing between *hold left* and *hold right* were slightly off, or that the sensors had different response curves that should have been corrected by calibration. Combining both grasps into a single *hold tightly* grasp would drastically reduce error rates.

A comparison to other approaches from related work can be found in Section 5.7.

## 5.6 Grasp Recognition Using Machine Learning

As heuristics proved to not work very well for arbitrary users, I switched to a hybrid recognizer for the HandSense demo that was presented at TEI 2009. This hybrid recognizer combines a machine-learning recognizer with a reduced set of heuristics. For the machine learning part, the libSVM module for Python (Chang and Lin 2011) is used, which implements a Support Vector Machine (SVM) classifier. The HandSense sensor data is ideal for machine learning approaches, as the sensor data needs only little preprocessing in order to be used as features in a pattern matching algorithm. The low number of sensors allowed to directly use each sensor reading as a feature for classification, avoiding the need for sophisticated feature reduction. As the SVM classifier proved to be not precise enough to distinguish between *hold left* and *hold right* grasps, those were combined into a class *hold*. The proven heuristics were used to distinguish between *hold left* and *hold right* once a *hold* grasp was recognized by the SVM classifier.

This refined prototype proved to work very well across many different users during the TEI 2009 demo session. However, due to time limitations no formal evaluation was conducted.

## 5.7 Comparison to Related Work

HandSense differs from previous and subsequent related work in three key aspects: a) higher sensor resolution, b) lower sensor count, and c) use of heuristics instead of machine learning.

| Prototype | # of sensors | resolution per sensor | update rate |
|---|---:|---:|---:|
| Smart Gun (Veldhuis et al. 2004) | 1936 | 8 bit | not reported |
| Tango (Kry and Pai 2006a) | 256 | 8 bit | 100 Hz |
| Samsung Prototype (Kim et al. 2006) | 64 | 8 bit | 30 Hz |
| HandSense (Wimmer and Boring 2009) | 4 | 16 bit | 25 Hz |
| Bar of Soap (Taylor and Bove 2009) | 72 | 1 bit | 100 Hz |

Table 5.3: Comparison of technical properties of different grasp sensing prototypes, sorted by date of publication

### 5.7.1 Sensor Resolution

As shown in Table 5.3, HandSense is unique in that it uses sensors with a resolution of 16 bit per sample. Most other sensors used in grasp-sensitive surfaces only offer a resolution of 8 bit or less. Some systems (not in the table) use the analog-digital converter (ADC) of a microcontroller which usually offers a resolution of 10 bit. As capacitance increases exponentially the closer an object gets to the sensor, a low resolution sensor only allows precise distance measurement either of very close objects or of distant objects. The higher resolution of the CapToolKit sensors used in HandSense allows for simultaneously measuring distances with micrometer resolution at close distances and with millimeter resolution at greater distances. This allowed us to reliably distinguish between left-handed and right-handed holding of the prototype, even though the hands covered the same surface area in both cases. No other prior or later grasp sensing system offers this capability.

### 5.7.2 Sensor Count

Compared to other grasp-sensitive prototypes, HandSense employs much fewer sensors (Table 5.3).

| Prototype | # of grasps | single-user accuracy | multi-user accuracy |
|---|---|---|---|
| Samsung Prototype (Kim et al. 2006) | 8 | 92% | 88% |
| HandSense (Wimmer and Boring 2009) | 6 | 93% | 83% |
| Bar of Soap (Taylor and Bove 2009) | 5 | 95% | 79% |

Table 5.4: Comparison of recognition rates for three grasp sensing prototypes, sorted by date of publication. For accuracy, the best recognition rate reported in the paper is given. As hardware, tasks, grasp types, data collection methods, and analysis methods wildly differ between all publications, direct comparisons of accuracies are not valid.

Most grasp-sensitive surfaces published so far cover a majority of the device. In contrast, sensor antennas only cover 8% of the surface of the HandSense prototype. As the results of the user study show, careful placement of the sensors at locations which offer much information about a grasp allows distinguishing about the same number of different grasps as systems with higher sensor count (Table 5.4).

### 5.7.3 Classification Approach

Overall, the recognition rates of HandSense heuristics compare favorably to those of more sophisticated machine-learning approaches (Kim et al. 2006; Taylor and Bove 2009); see Table 5.4. These achieve recognition rates of 90%-100% for grasps by the same user who trained the classifier and 80%-90% for grasps by different users.

However, the grasps to be employed in the HandSense study were much more restricted than those in the other two studies. For example, in the "Bar of Soap" study (Taylor and Bove 2009), participants were asked to hold the device as if they wanted to conduct a certain action. They were not given specific instructions on how to actually grasp the device. Additionally, different approaches to data collection and data analysis make it hard to compare results from different studies. For simple exploratory prototypes, heuristics may offer advantages over machine learning approaches. Simple sets of rules are straightforward to implement, whereas selecting a classifier, choosing optimal features, and training the classifier require both knowledge about machine learning and time for training. Heuristics also require researchers to first gain a conceptual understanding of the problem space instead of just feeding sensor data into a magical black box that somehow guesses the correct answer most of the time.

On the other hand, heuristics are probably not well suited for most commercial grasp sensing applications. They are hard to generalize for different users, requiring initial calibration for each user. In addition, extending the set of recognized grasps is harder with heuristics, as each extension requires the design of a custom heuristic, whereas machine learning algorithms are just retrained with the extended dataset.

However, heuristics may also be used in combination with machine learning approaches, as I explored with the HandSense hybrid recognizer. In this case, a machine learning classifier distinguishes a set of general grasp types, and heuristics provide further information about these - or vice versa.

## 5.8 Discussion

In this chapter, I presented HandSense, a system for grasp sensing which uses a small number of sensors and simple heuristics to achieve reliable classification of six different grasps.

HandSense encompasses several novel approaches to grasp sensing:

Commercial availability of CapToolKit (and soon OpenCapSense) allows designers without knowledge of sensor design to quickly and cheaply make tangible objects grasp-sensitive. Especially for prototyping grasp-sensitive objects, a small number of reconfigurable capacitive sensors allows iterating more often.

HandSense is capable of distinguishing between left-handed and right-handed grasps. This demonstrates that highly sensitive capacitive sensors may be used to infer grasp information that is not available to low-resolution (touch) sensors.

As verified in the user study, heuristics allow for reliably distinguishing between grasp types, with a recognition rate similar to comparable machine-learning approaches. However, heuristics that work well for a single user may not be suitable for different users. Unfortunately, as we pre-defined a set of grasps, our study can not be easily compared to other studies from the related work which let users choose a grasp for a certain action.

The three unique aspects of HandSense - low sensor count, high sensor resolution, and use of heuristics - demonstrate that there is not the *one right way* in grasp sensing. Rather, sensor hardware and classification algorithms are highly dependent on the grasp types to be distinguished.

Additionally, we were the first to discuss several novel and important issues for grasp sensing in the original paper, such as the concepts of *implicit grasp interaction* and *grasp affordances*. Those topics are covered in detail in Chapter 12.

As a generic technique for prototyping grasp-sensitive surfaces, HandSense still has four inherent limitations:

1) HandSense requires basic knowledge of soldering and attention to sensor placement.
2) CapToolKit sensors are no longer commercially available[2].
3) The size of the sensors makes it difficult to build extremely small objects.
4) As a single CapBoard only supports up to eight sensors, spatial resolution is limited.

Nevertheless, HandSense is still better suited for prototyping than other approaches using capacitive sensors.

Unfortunately, the small number of sensors/antennas severely limits the spatial resolution achievable with HandSense. As discussed in Section 5.3, a small number of sensors may be sufficient for recognizing certain grasp types if sensor locations are chosen carefully. However, in the early design phases, designers might not know how people will grasp an object. Therefore, optimal sensor placement requires extensive experience of several rounds of trial-and-error. Settling on a bad sensor layout early in the design process may unnecessarily constrain the recognition accuracy of interesting grasps and cause designers to compromise their designs. Ideally, a prototyping technique should have a very high spatial resolution and allow covering the whole surface. This allows designers to just capture high-resolution grasp signatures. Once it is known which grasp types need to be distinguished and where people grasp the object, sensor density and resolution can be decreased in certain areas. Those areas can be identified using *principal component analysis* (PCA) or other means, given a high-resolution corpus of grasp signatures.

In the following chapter I present a high-resolution prototyping technique for grasp-sensitive surfaces that only requires minimal soldering skills. As machine-learning algorithms have been explored in much detail by other researchers, my further research focused on sensor hardware and filtering approaches, and excluded classification algorithms.

[2] However, an early revision of its unofficial successor OpenCapSense is already available.

Chapter 6

# FlyEye:

## Non-Planar, Deformable Grasp Sensors Using Optical Fiber

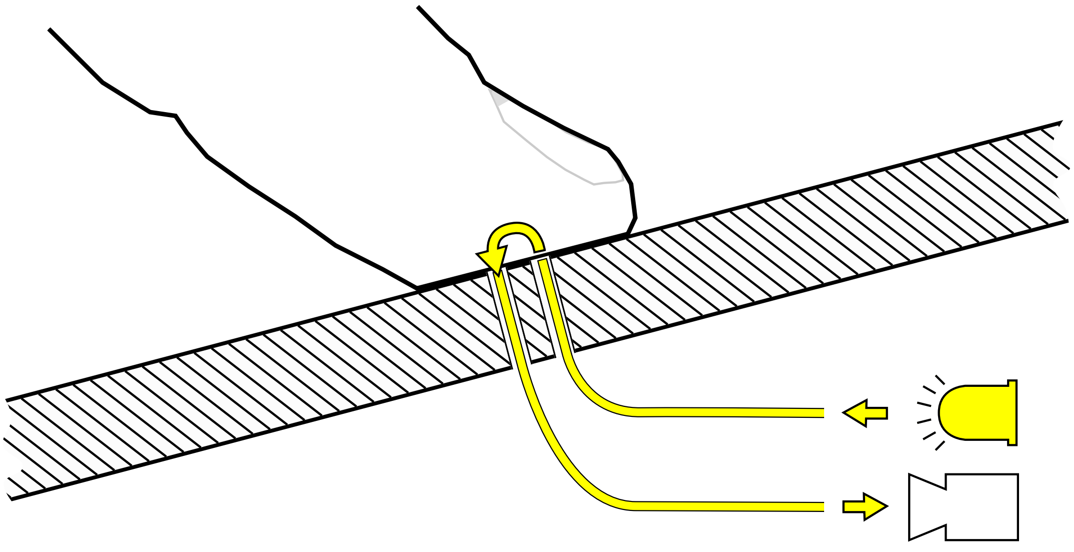

**Figure 6.1:** General principle of FlyEye: Through optical fiber, an infrared LED emits light from an object's surface. When a user touches the surface, the IR light gets partially reflected back into adjacent optical fibers which are connected to a camera. Computer vision algorithms translate the camera image into touch locations.

*FlyEye enables designers to make arbitrary surfaces grasp-sensitive without the need for custom sensors or soldering. To this end, FlyEye uses a camera sensor for capturing contact on the surface of an object. Many individual optical fibers are embedded into the surface of the object and connected to the camera. The camera may be either placed inside the object or outside. Custom differential illumination ensures robust recognition of touches and grasps under changing lighting conditions. A custom calibration algorithm associates pixels in the camera's image with touch locations on the surface. While FlyEye allows grasp-sensing on arbitrary shaped surfaces, without knowledge about electrical engineering, its usefulness is limited in practice. Augmenting large surfaces requires manual placement of hundreds or thousands of fibers. Additionally, fibers and camera occupy space inside the object, making it impossible to make solid objects grasp-sensitive. Nevertheless, FlyEye allows for building both tiny grasp-sensitive objects as well as large touch-sensitive installations. Its unique properties make it a complementary alternative to traditional capacitive sensors.*

**Attribution:** This chapter is based on my paper "FlyEye: Non-Planar, Deformable Grasp Sensors Using Optical Fiber" (Wimmer 2010a). Nobody else was involved in concept development, implementation, or writing the paper. Several figures from the paper have been reused in this chapter.

## 6.1 Motivation and General Concept

Building on the experience gathered with HandSense, the goal for FlyEye was to develop a sensing technique that does not require any knowledge about electronics, avoids custom, hard to obtain, electronic circuits, offers a high spatial resolution, and allows for rapid prototyping.

If one wants to avoid specialized hardware and custom electronics, this limits the number of interfaces for reading sensor data into a computer to common peripherals such as keyboard, mouse, microphone or webcam.

Utilizing computer keyboards or mice for digitizing arbitrary input is possible (Baudisch, Sinclair, and Wilson 2007; Baur, Hilliges, and Butz 2008) but requires soldering and significant modifications to the hardware. The microphone input of a computer may also be used for communication (Kuo et al. 2010). However, this requires custom electronics, too.

Webcams (and higher-end industrial cameras) are connected to a computer via USB or FireWire. They offer high bandwidth, low latency, and can be obtained easily and often cheaply. These properties have led to a widespread adoption of camera-based touch detection in interactive surfaces (Han 2005; Schöning et al. 2008; Benko, Wilson, and Balakrishnan 2008). As the raw sensor data provided by the camera is directly observable, debugging capturing setup and the processing toolchain is straightforward.

Thus, cameras were a promising option for prototyping grasp-sensitive surfaces. To visually capture the surface from the inside, objects need to be transparent or have holes drilled into them. However, even with fish-eye lenses, cameras embedded into the object can only capture a small portion of the object's surface. In order to capture the whole device surface, multiple cameras would have to be placed inside the object. This approach is not feasible for small objects. In addition, capturing the whole surface of a complexly shaped object would require careful placement of the cameras. This would hinder rapid, interactive prototyping.

FlyEye solves this issue by employing bundles of optical fiber to spread and guide the field of view of a camera. It therefore allows a single camera to capture the whole surface of the object. Using optical fibers also allows putting the camera outside the object, making very small grasp-sensitive objects possible.

To reliably sense grasps, FlyEye combines several approaches from optics, computer vision, and information visualization:

The ends of optical fibers are embedded flush into the object's surface (Figure 6.1). Some of these fibers are connected to an infrared (IR) light source at their other end. Once a finger or palm touches the surface, some of the IR light emitted from these fibers is reflected back into adjacent fibers whose other ends are connected to a camera. Thus, the camera sees all touched fibers as bright dots. In order to achieve robust recognition under variable lighting conditions, differential IR illumination is used.

Using a custom *relative mapping* calibration process, each bright dot in the camera image is mapped to a relative location on the surface. This allows FlyEye to recognize grasps without designers having to care about placement and order of fibers.

The remainder of this chapter is organized as follows:

- Section 6.2 relates FlyEye to other research.
- Section 6.3 describes the hardware design of FlyEye.
- Section 6.4 explains the *relative mapping* approach implemented for FlyEye.
- Section 6.5 presents prototypes that emphasize capabilities of the FlyEye concept
- Section 6.6 addresses general and specific limitations of concept and implementation.
- Section 6.7 concludes this chapter with a summary of FlyEye's properties and impact.

## 6.2 Related Work

Optical sensing methods are rarely used in grasp-sensitive surfaces (see Section 4.4). While FlyEye is presently the only grasp-sensing approach employing optical fibers, these have previously and subsequently been used in some *touch*-sensing approaches. Several research projects using optical fibers have been published in 2009/10. *FiberBoard* (Jackson, Bartindale, and Olivier 2009) and *FiberSense* (Bartindale, Jackson, and Olivier 2009) were published between submission and publication of my FlyEye paper. We did not know of each other's research before. With *FiberBoard*, Jackson et al. propose a system whose working principle is very similar to FlyEye. They employ optical fibers to reduce the thickness of FTIR-based or DI-based multi-touch tables.

Commonly, these tables use a camera to capture proximity and touch on a rear-projected screen or an LCD panel. In order to capture the whole surface from behind, the camera has to be equipped with a wide-angle lens and needs to be placed at a distance that is in the same order of magnitude as the diameter of the surface. To decrease the thickness of the table, mirrors can prolong the path of light. Some systems - such as the original Microsoft Surface - employ multiple cameras that each only cover a small part of the table. These can be placed closer to the surface than a single camera. With FiberBoard, individual optical fibers are embedded into a board with a grid of holes which is mounted behind the surface to be captured. The other ends of those fibers are bundled together and mounted in front of a camera. Thereby, the field of view is independent of the distance between camera and surface. The camera can thus be placed very closed to the surface, reducing the overall thickness of the interactive tabletop.

For FTIR-based surfaces, IR light is sent into the sides of a thick pane of acrylic glass. When a finger touches the glass pane, a part of the injected light is frustrated at the touch location and get reflect into the optical fibers mounted behind the glass pane. For DI, a second set of fibers is interspersed with the other fibers. This second set is connected to an IR light source. Through the fibers, the light source illuminates fingers in close proximity. A part of the reflected light is captured by the 'sensing' fibers and directed to the camera.

*FiberSense* extends the general principle of FiberBoard to non-planar surfaces. The proposed hardware setup is very similar to FlyEye. Bartindale et al. suggest that ambient light, FTIR, and DI may be used. In both systems, *FiberBoard* and *FiberSense*, the mapping of surface locations to pixels in the camera image is obtained by using a two-step calibration process. A projector projects a moving horizontal white line onto the surface. When the line passes a fiber on the surface, the corresponding fiber end at the camera lights up. This allows the system to determine order of fibers in the vertical direction. In a second calibration step, a moving vertical line provides the order of fibers in the horizontal direction. This allows for an absolute mapping of pixels in the camera image to locations on the surface.

While FlyEye and *FiberBoard/FiberSense* share the same general principle, there are several differences:

- FlyEye explicitly focuses on grasp-sensitive surfaces that cover the whole object. This requires a different spatial resolution than touch sensing.

- FlyEye uses differential illumination for increased robustness of sensor data.
- FlyEye employs a *relative mapping* calibration that is tailored to complex non-planar surfaces and rapid prototyping.

Rock-Paper-Fibers (Rudeck and Baudisch 2012) consists of a bundle of optical fibers connected to a camera on one end. The other ends of the fibers are left unbundled. By manually bundling these ends into one of several shapes with one's fingers, and swiping across this shape, one can invoke different actions. Inspired by FlyEye, Rock-Paper-Fibers uses Hough circles for identifying individual fibers in the camera's image. However, only ambient light is used for touch detection. Instead of creating a *relative mapping*, as presented with FlyEye, Rudeck and Baudisch count the number of fibers occluded by the swiping finger over time and match this gradient to a number of pre-recorded samples.

The *FuSA*$^2$ *Touch Display* (Nakajima et al. 2011) is a 'furry' surface made out of optical fiber. These fibers act as both input and output pixels. The authors mention that they plan to employ an algorithm that "may resemble a previous calibration technique [29]", referencing FlyEye.

# 6.3 Hardware Design

FlyEye uses optical fibers embedded into the surface of an object to detect touches and recognize grasps. To this end, optical fibers are into inserted many small holes drilled into the object's surface. The inside ends of those fibers are bundled into two fiber bundles whose ends are cut flush. One fiber bundle is connected to a modulated IR light source. Those fibers emit IR light from the surface. The other, larger fiber bundle is mounted in front of a camera which can thereby capture the level of light falling into each individual fiber.

## 6.3.1 Fiber Choice and Placement

Optical fibers may be made out of glass or plastics (Wikipedia 2013). Glass fibers offer a lower signal loss, and are therefore preferred for data transmission over long distances. However, glass fibers are very brittle and require special equipment for handling and customization. Optical fibers made from Poly(methyl methacrylate) (PMMA, acrylic glass) are cheap (approx. 0.30 EUR / m), very flexible, and can be cut easily.

For FlyEye, I used PMMA fibers with a width of 0.5 mm to 1.0 mm. In general, a width of 1.0 mm is preferable for most applications, as it allows for easy manual customization while still being thin and flexible enough to fit several hundred fibers into a small object.

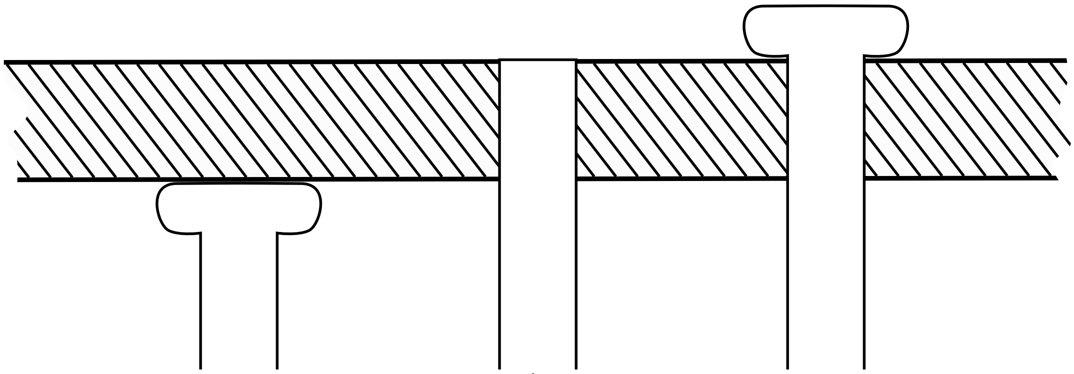


**Figure 6.2:** Optical fibers can be embedded in a surface in three ways: a) b) c)

One end of each fiber needs to be attached to the object's surface. This may be facilitated by broadening the end of the fiber. Broadening can be easily done by pushing the end onto a hot surface, allowing it to partially melt and form into the shape of a nail's head[1]. The so formed nail-head ends allow attachment to the surface in three different ways (Figure 6.2a-c):

a) Fibers are glued to a transparent surface from behind
b) Fibers are glued into holes drilled into the surface and get beheaded.
c) Fibers are inserted into holes drilled into the surface and held in place by the nail-head end

Which of these is the best approach depends on application, surface properties, and fiber diameter. Gluing fibers onto the inside of a surface (Figure 6.2a) is very tedious, works only for transparent surfaces, and results in very fragile attachments. However, the surface won't get damaged. Gluing fibers flush into holes in the surface (Figure 6.2b) retains a smooth surface. This mode of attachment is reasonably robust if the surface is a few millimeters in thickness. The fastest and most robust attachment is to just insert the fiber until the nail-head end touches the surface (Figure 6.2c). The fiber is usually held in place by friction but may also be covered with transparent glue or silicone. This also fills gaps between fiber ends, resulting in a reasonably smooth surface.

Not cutting off the broadened end also increases the angle of incidence and thus the sensitivity of the fiber. For most FlyEye prototypes, attachment method 'c' was employed as it is by far the fastest and most robust.

The general process for building a grasp-sensitive object using the FlyEye method is as follows:

[1] For the FlyEye prototypes, I pressed the side of a hot soldering iron onto each fiber's end for a short time.

1) Pick an object that should be made grasp-sensitive. The object needs to be hollow to accommodate the optical fibers.
2) Drill a lot of small holes into the surface at regular intervals (see below). Hole diameter should equal fiber diameter.
3) Cut fiber strands of appropriate length, so that the fibers can be easily routed to the camera.
4) Broaden one end of each fiber by pushing it onto a hot surface.
5) Insert sensing fibers into most of the holes. Pay no attention to order. Leave some regularly spaced holes empty for the illuminating fibers.
6) Bundle the other ends of all sensing fibers tightly together with hot-shrink tube.
7) Cut the bundled end flush with a sharp knife.
8) Polish the bundled end using fine-grained sanding paper; finish polishing with regular printer paper.
9) Repeat steps 5-8 with the illuminating fibers.
10) Optionally cover the surface with transparent silicone, hot glue, or transparent varnish.
11) Attach the illuminating fiber bundle to an IR light source, and connect the sensing fiber bundle to a camera.

The spacing of fibers depends on the required resolution. Bartindale et al. (2009) suggest a distance of 10 mm between fibers. This ensures that a finger of average width always covers at least one fiber end. However, FlyEye relies on a pair of illuminating and sensing fibers to increase robustness. It requires a finger to always touch at least two adjacent fibers. Therefore, spacing should be lower than 10 mm; ideally it is 5 mm or lower. Spatial resolution may be varied depending on grasp affordances. For example, it might not be necessary to place any fibers at locations which are not touched during grasps. At locations where high resolution is necessary to reliably differentiate grasps, fiber density may be increased.

As sensing and processing does not need to happen directly at the touch location, camera and IR light source may be placed outside the grasp-sensitive object, being connected to the surface via a long optical fiber tether. This allows making very small objects grasp-sensitive.

Fibers are usually inserted one by one and bundled together once all fibers are in place. Therefore, it is impractical to maintain the order of fibers on the surface also within the fiber bundle. Two fibers that are adjacent on the surface are usually not adjacent within the bundle. FlyEye achieves the correct mapping between pixels in the camera image and touch locations on the object's surface by using a custom calibration step (see Section 6.4).

### 6.3.2 Active, Differential IR Illumination

FlyEye employs modulated IR light to achieve robust recognition of touches. However, a simplified, passive, version of FlyEye could work with just ambient light (Figure 6.3a).

With **passive illumination**, ambient light from the sun or artificial light sources falls onto the object's surface, travels through the fibers, and illuminates the fibers' ends. The illuminated fiber ends show up in the camera image as circles with uniform brightness.

When the user touches the surface, they occlude some fiber ends, hindering light to enter those fibers. These occluded fibers show up darker in the camera image. However, such a setup is prone to recognition errors under non-uniform lighting conditions or whenever ambient light levels change. In order to work at all, such a system would have to be calibrated for the current lighting setup. As the amount of light falling onto the sides of the object changes depending on its orientation, a graspable object would need to always have a fixed orientation in order for FlyEye to work. In addition, the user's body or hand might shield parts of the surface from light without actually touching them, causing erroneous detection of touches.

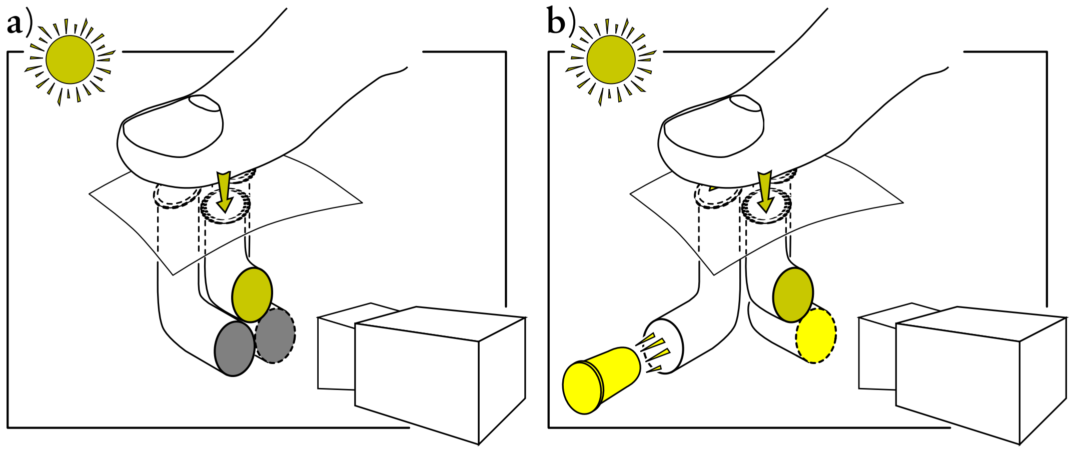


**Figure 6.3:** Passive illumination (a) uses ambient light to distinguish between touched and untouched fibers. Active illumination (b) employs a light source and re-appropriates some fibers as light emitters. This makes touch recognition more robust under low-light conditions. However, with very bright ambient light it becomes difficult to distinguish between touched and untouched fibers, as both appear bright.

**Active illumination** (Figure 6.3b) mitigates these issues. Similar to *FiberSense*, some of the fibers are not connected to the camera but to a light source. These fibers need to be interspersed with sensing fibers so that a touch always covers both a lit fiber and a sensing fiber. The touching finger reflects some of the light emitted from a lit fiber back into adjacent sensing fibers. Those fibers show up brighter in the camera image. It is not necessary to add a transparent spacing layer on top of the surface. Even if the finger completely covers the fibers, subsurface scattering within the finger tissue allows light to pass from lit fibers to adjacent sensing fibers.

Fibers that are not covered receive ambient light. As long as the fibers lit by ambient light are not brighter than the ones lit by the employed light source, touches can be detected. However, as only a small fraction of emitted light is reflected back into the sensing fibers, the light source has to be very powerful.

While visible light may be used, FlyEye employs IR light that is invisible to the human eye. This avoids blinding the user. In addition to IR light, visible light may be emitted through the fibers for displaying information or giving visual feedback.

As most artificial light sources emit only little IR light, using IR light and adding an IR-pass filter in front of the camera increases contrast between touched and non-touched fibers. For most FlyEye prototypes a single IR LED (Osram SFH485) was used to illuminate all fibers.

In practice, some of the fibers allow a lot less light to pass through them than others because cutting and melting fibers significantly alters their transmittance. Therefore, an untouched fiber might appear brighter in the camera image than a fiber that is being touched.

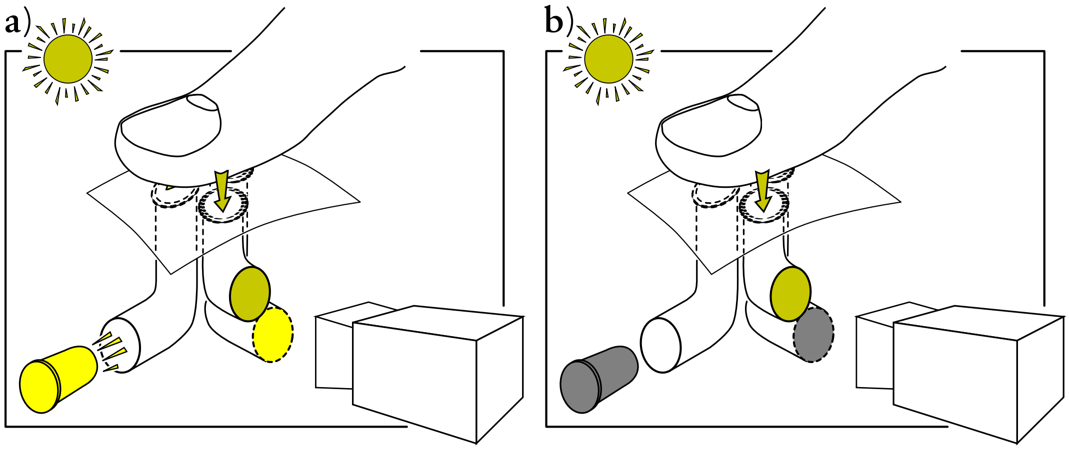


**Figure 6.4:** Differential illumination increases robustness against changes in ambient light. For this technique, even frames are captured with active illumination (a), whereas odd frames are captured without active illumination (b). Touched fibers therefore appear to blink with a frequency that is half of the frame rate. Fibers that are not being touched remain at the same brightness level. Calculating a difference image of subsequent frames shows touched fibers with high contrast.

To counter this effect, FlyEye employs **differential illumination**. Thereby, active illumination is toggled on and off for subsequent captured frames (Figure 6.4). Thus, touched fibers appear alternately bright (Figure 6.4a) and dark (Figure 6.4b) while untouched fibers always remain at the ambient light level. Calculating the difference image between subsequent frames results in a high-contrast image where touched fibers appear very bright and untouched fibers appear very dark, independent of ambient light levels and variations in transmittance between fibers.

Switching the IR light source on and off every other frame can be done in a few ways. The most flexible approach is to have the capturing software control the LED via the parallel printer port or via a microcontroller (e.g., Arduino) connected to the host PC via USB.

Most industrial cameras also offer a trigger output that is activated during image capture. Connecting the IR LED to a camera's trigger output ensures synchronous illumination and requires less additional hardware than software-controlled LEDs. High-end cameras can be configured to trigger the output only for every second captured frame. Industrial cameras at the low end of the price range - such as the Point Grey Firefly MV used in the FlyEye prototypes - only have a non-configurable output that is triggered at every frame. A simple circuit enables differential illumination with these cameras, too: In order to only activate light every second frame, a binary counter IC (CMOS IC 4518) is used. Its counter input is connected to the camera's trigger output. The output pin for the lowest bit - which changes on every counter increment - then toggles the LED on every trigger pulse. As the counter IC only provides limited power, a transistor (BC107B) is placed between counter IC and LED.

Modulated, very bright IR light could be used to further increase contrast (Echtler et al. 2009). However, for FlyEye this was not necessary, and therefore not implemented.

## 6.4 Relative Mapping

When inserting dozens or hundreds of fibers, it is not feasible to keep them ordered until bundling them. Therefore, the arrangement of fibers on the surface is not the same as within the bundled end. However, for touch sensing and grasp recognition, it is necessary to know where the surface is being touched. Therefore, a mapping between lit pixels in the camera image and touched fibers on the surface needs to be established. For touch sensing, the absolute position of each touch on the surface needs to be known. For grasp sensing, it is not always necessary to know exact touch locations. If no rotational invariance is required, machine-learning classifiers can be trained using the sensor data from a few exemplary grasps per grasp type to be recognized. However, low-dimensional feature vectors (i.e., consisting of fewer sensor values) are better suited for machine learning classifiers than high-dimensional ones (Taylor and Bove 2009) as the resultant model has less computational complexity and memory requirements. In addition, low-dimensional feature vectors are generally more robust against small differences between grasps than high-dimensional feature vectors.

Without preprocessing, the feature vector has a dimension equal to the number of pixels in the camera image. For robust grasp classification, it is necessary to significantly reduce the number of features. For FlyEye, the goal is to transform the raw camera image (“which pixels are bright?”) to a feature vector containing a low number of ‘surface areas’ that are being touched (“which areas on the surface are bright?”). To do so, adjacent fibers need to be combined into larger groups. Therefore it is necessary to reconstruct adjacency between fibers. To this end, FlyEye employs a custom calibration approach called “relative mapping”. In this chapter I present this calibration method and the complete FlyEye filter chain that turns camera images into touch locations.

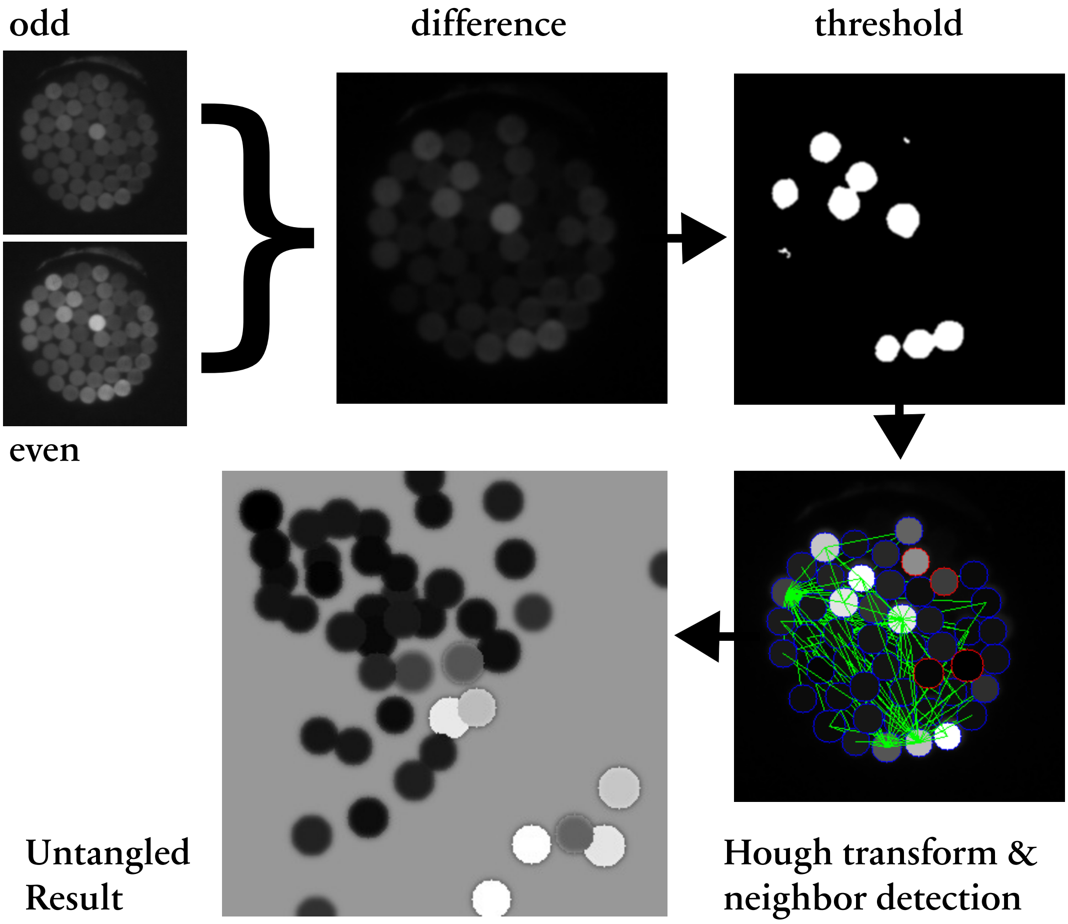


**Figure 6.5:** For detecting touch and grasping, FlyEye employs an approach that includes background subtraction (a), locating individual fibers using Hough transforms (b), and a relative mapping algorithm (c) that re-orders the detected fibers so that fibers that are adjacent on the surface are also adjacent in the output image.

### 6.4.1 Mapping of Fibers: Absolute vs. Relative

In order to identify which fiber end in the camera image belongs to which other fiber end emerging from the surface, calibration is necessary. A naive approach would be to touch every single fiber on the surface once and record which fiber end in the camera image lights up. This becomes very tedious with an increasing number of fibers.

A more efficient approach is to project changing light patterns onto the surface and correlate these light pattern with the lit fibers in the camera image (Bartindale, Jackson, and Olivier 2009; Wimmer 2010a). Bartindale, Jackson, and Olivier (2009) suggest a two-step calibration process using an off-the-shelf video projector that was originally developed for flat surfaces (Jackson, Bartindale, and Olivier 2009). The projector is placed at close distance to the surface to be calibrated, so that the projected image covers the whole surface. In the first step, a bright horizontal line moving from top to bottom is projected onto the surface. The calibration algorithm records all lit fibers in the camera image for each position of the moving line of light. This is repeated in the second step with a vertical line moving from left to right. Thus, for each fiber a horizontal and vertical location can be determined.

Bartindale, Jackson, and Olivier (2009) do not further discuss how well their (planar projection) mapping is actually suited for non-planar, complex forms with omnidirectional touch sensors. While it is a quite elegant solution for planar surfaces, it has some limitations for non-planar surfaces: Fibers emerging in a right angle to the projector or facing away from it do not receive any light. Therefore, the object needs to be rotated between calibration passes to cover the whole surface. To correctly merge sets of calibration data from different angles, projector and object positions need to be controlled exactly. This impedes rapid prototyping. In addition, projected light may not be able to completely cover objects with cavities or overlapping surfaces . Due to these limitations, FlyEye does not employ such an *absolute mapping* approach.

Instead, FlyEye utilizes the fact that it is not necessary to know the absolute position of each fiber in order to reduce the number of features. Simply knowing which fibers are adjacent to a given fiber on the surface allows for grouping fibers into larger areas. To determine adjacency, I developed a fast, simple, and iterative calibration method called *relative mapping*.

In this process, a finger is dragged across the surface to be calibrated one or more times. At any given time, the finger touches a small number of adjacent fibers. As long as only a single finger touches the surface, all fibers that appear bright in the camera image at the same time have to be adjacent on the surface. From this local adjacency data, global adjacency of all fibers can be calculated. The basic *relative mapping* allows for generating low-dimensional feature vectors for grasp recognition and can also be used to recognize simple touch gestures on the surface. If the absolute positions of a few fibers are known, *relative mapping* can also estimate the absolute locations of all other fibers.

The *relative mapping* algorithm implemented for FlyEye comprises three steps:

- identifying individual fibers
- finding neighbors
- untangling the neighbors

These steps may be executed in parallel to incrementally increase the quality of the mapping. For the FlyEye prototypes, this algorithm was implemented in Python using OpenCV[2] for image acquisition and processing and NetworkX[3] for the Fruchterman-Reingold graph optimization algorithm. The source code is available online[4]

### 6.4.2 Identifying Individual Fibers

In order to determine the brightness of the fiber ends, these have to be located within the camera image. This is straightforward, as fiber ends show up as circles of consistent brightness (Figure 6.5a). For finding all fibers, the user drags their finger across the whole surface, causing all fibers to light up subsequently. As described in Section 6.3, first, the difference between two subsequent frames is calculated in order to increase robustness against changes in ambient light. To this end, the frame with lower average brightness is subtracted from the other one. The resulting image is converted to a grayscale color space. A Gaussian blur filter is applied to remove noise. A threshold filter splits the image into a black background and white circles for touched fibers (Figure 6.5b). This image is again blurred slightly to reduce thresholding artifacts. A circular Hough transform locates all circular areas of consistent brightness with a diameter equivalent to the fibers' diameter. In the calibration GUI, located fibers are highlighted by red circles, indicating progress and erroneous identifications. The locations of all fibers identified this way are stored in a calibration file.

### 6.4.3 Finding Neighbors

Once all fibers within the camera image are located, adjacent fibers can be identified. To this end, the user drags a finger or other small object across the surface a second time. All fibers that appear bright in the camera image at the same time are assumed to be adjacent on the surface. This adjacency information is stored in an undirected graph. Every fiber is represented as a node. For every captured frame, edges are inserted between all adjacent fibers. Dragging the finger across the whole surface thus generates a graph connecting all fiber nodes. All edges have the same weight. The quality of the adjacency graph depends on the ratio of finger size to fiber spacing on the surface. If

[2] `http://opencv.org/`

[3] `http://networkx.lanl.gov/`

[4] `http://www.medien.ifi.lmu.de/team/raphael.wimmer/projects/FlyEye/`

the finger has a too small diameter, it might only cover one fiber at a time. In this case, no adjacency can be determined. If the finger is too wide, it always covers many fibers of which most are not actually close to each other. In this case, information about 'real' adjacency is lost. Ideally, the finger's diameter is between once and twice the distance between adjacent sensing fibers. As it may prove difficult to grow fingers of appropriate diameter for each specific fiber spacing, a reflective plastic rod may be used instead.

This adjacency information might also be captured together with the individual fiber positions during the first step. In my reference implementation I opted to keep these two steps separate in order to retain flexibility.

### 6.4.4 Untangling the Neighbors

In the final step, the adjacency graph is used to virtually re-arrange the fibers in the camera image so that their arrangement reflects the arrangement of fibers on the surface. To this end, all nodes within the graph are laid out in a two dimensional space so that adjacent nodes are kept close together, while non-adjacent nodes are kept far apart. In other words, the graph's nodes need to rearranged so that as few edges as possible overlap. The Fruchterman-Reingold force-directed "spring layout" algorithm (Fruchterman and Reingold 1991) does exactly this, iteratively generating an optimized graph in which nodes which share edges are placed closely together. This layout therefore mirrors the arrangement of fibers on the surface. Coloring each node of this untangled graph with the brightness of the associated fiber within the camera image creates a map of touched areas on the surface. Due to the limited information gained in the calibration process, this map is distorted, however. It is arbitrarily rotated, scaled, and skewed compared to the actual fiber arrangement. As the Fruchterman-Reingold-algorithm is non-deterministic, the final node positions vary between different runs of the algorithm. Nevertheless, the map allows aggregating fibers with the same brightness into 'touched areas', interpolating values between neighboring fibers, and recognizing touch gestures.

### 6.4.5 Potential Optimizations

While the approach described above works reliably, some improvements that were not incorporated into the prototypes are possible.

For example, the *relative mapping* algorithm might also be applied simultaneously while the user interacts with the device, gradually increasing the quality of the interaction the more fiber positions are known.

For touch sensing, generating absolute mappings would be helpful. A relative mapping can be easily transformed into an absolute mapping by determining the absolute locations of a few fibers and using these as anchors. The nodes of these anchor fibers are placed at fixed positions within the graph and are not subject to the Fruchterman-Reingold re-arrangements. Therefore, the other nodes would be automatically arranged between those anchor points.

## 6.5 FlyEye Prototypes

I have built three prototypes with different shapes over the course of this research project.

The first prototype (Figure 6.6c) comprises 75 strands of optical fiber (diameter: 1 mm) laid out in a planar hexagonal grid with a size of 16 x 12 mm. It was used for developing the general concept and the relative mapping algorithm. Using a hexagonal grid ensures that all 51 sensing fibers are directly adjacent to one of the 24 illuminating fibers. Fiber ends on the surface were covered with a thin layer of hot glue to achieve a smooth surface. The fiber grid has a resolution of 3.76 $mm^2$ per fiber or approx. 13 dpi. If necessary, effective sensor resolution can be increased further by interpolating between adjacent fibers.

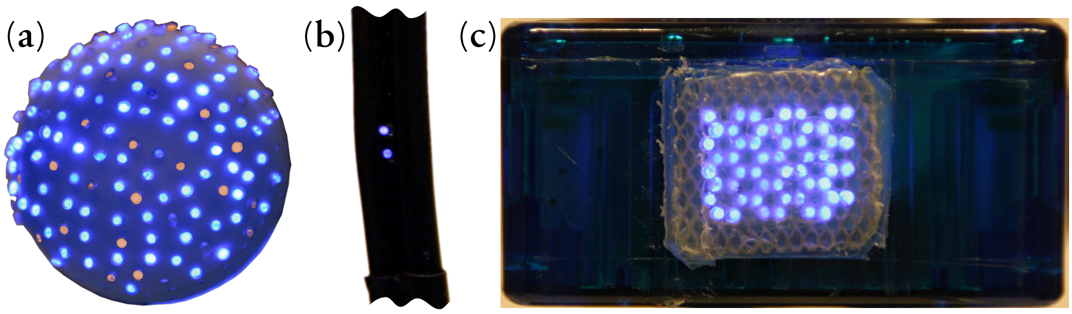


**Figure 6.6:** Three exemplary FlyEye prototypes have been built. FlyEye allows touch-enabling non-planar surfaces like a ping-pong ball (a). It also makes it possible to embed tiny touch sensors into flexible surfaces such as cables (b). This allows e.g. for controlling a portable audio player. For preliminary tests, a hexagonal grid of fibers was built, too (c). While FlyEye employs invisible IR light, the *sensing fibers* were connected to a visible light source for this illustration.

As an example of a mid-resolution, non-planar touch-sensitive surface, one half of a ping-pong ball (Figure 6.6a) was equipped with 250 optical fibers (1 mm), divided into approx. 50 illuminating fibers and 200 sensing fibers. As the ball has a diameter of only 36 mm, the camera is placed outside. The fibers cover a surface area of 2000 $mm^2$ which results in a sensor resolution of 8 dpi. After inserting all fibers, the ball's surface was smoothed by applying a thin layer of transparent silicone. All demonstrations have been done with this ball.

FlyEye also allows for building tiny touch sensors using only two fibers. These can be embedded into very thin objects in order to make them touch-sensitive. In this case, only a single photoresistor is needed instead of a camera. For example, tiny media control buttons may be embedded into a headphones cable without requiring bulky electronics within the cable. As an example of this approach, headphone cables have been equipped with two 0.5 mm fibers which are routed within the cable alongside the copper threads. (Figure 6.6b).

## 6.6 Limitations

FlyEye is a rapid-prototyping technique for touch-sensitive surfaces that does not require knowledge of electronics or soldering. However, it has a number of limitations that limit its usefulness for some practical applications:

*Ease of use*. Cutting, flattening, inserting, and bundling several hundred fibers is tedious, even for small surfaces.

*Reusability*. Fibers can not be easily reused for other prototypes, as they need to be clipped to a different length for each prototype.

*Modifying the object*. Many holes need to be drilled into the surface, and fibers take up most of the space within the object. Therefore, it is often not feasible to instrument existing objects. Instead, 3D-printed replicas or empty casings need to be obtained.

*Expressiveness*. Unlike capacitive sensors, optical sensors are not able to distinguish between fingers and other, non-conductive objects. Therefore, FlyEye is more prone to erroneously detect touches. Unlike HandSense, FlyEye does not allow measuring further properties of the touching hand, such as the thickness of the tissue.

How these limitations affect the utility of FlyEye is discussed in the following and final section.

## 6.7 Discussion and Impact

In this chapter, I have presented FlyEye, a technique for prototyping grasp-sensitive surfaces.

Key contributions of my research are:

a) a novel prototyping technique that can be used for simple and complex prototypes
b) a novel, fast approach to fiber mapping that does not require an external infrastructure for calibration.

c) three prototypes that demonstrate the feasibility of the concept

Except for the differential illumination, FlyEye does not require knowledge of electronics or soldering skills. Instead, designers cut strands of optical fiber and embed these into the surface of an object and connect them to a camera. Calibration is a simple manual process. The whole calibration and tracking software pipeline operates on images. This makes it easy to visually identify optimal calibration parameters and erroneous fiber mapping. By requiring little previous knowledge, FlyEye enables anyone to build grasp-sensitive devices. FlyEye can be used to augment fully flexible objects and does not interfere with other sensing techniques. FlyEye does not replace existing sensing techniques but is another option in the designer's toolbox for building grasp-sensitive and touch-sensitive objects.

However, making large surfaces grasp-sensitive requires cutting and embedding hundreds or thousands of fibers. Therefore, the process is not ideal for prototyping large grasp-sensitive objects. The sensing technique may also be used in large, one-off installations, however.

In addition, the surface of the object needs to be perforated. Optical fibers take up much of the space on the inside. Therefore, it is often not possible to augment existing objects. Instead, an empty hull needs to be acquired or fabricated. Therefore, FlyEye is not well suited for augmenting arbitrary existing objects with touch-sensing capabilities but requires careful planning.

As mentioned in Section 6.2, ideas from FlyEye have partially inspired at least two other research projects. The FlyEye paper has been cited a few times so far. Most importantly, my experience developing and evaluating FlyEye made it clear to me that a good prototyping tool needs to be both accessible - which FlyEye is - and require little effort - which FlyEye does not. Therefore, FlyEye is not only a sensing technique in its own merit but was also a stepping stone towards the sensing technique presented in the following chapter: TDRtouch.

# Chapter 7

# TDRtouch:

## Versatile, Flexible Grasp Sensing Using a Single Cable

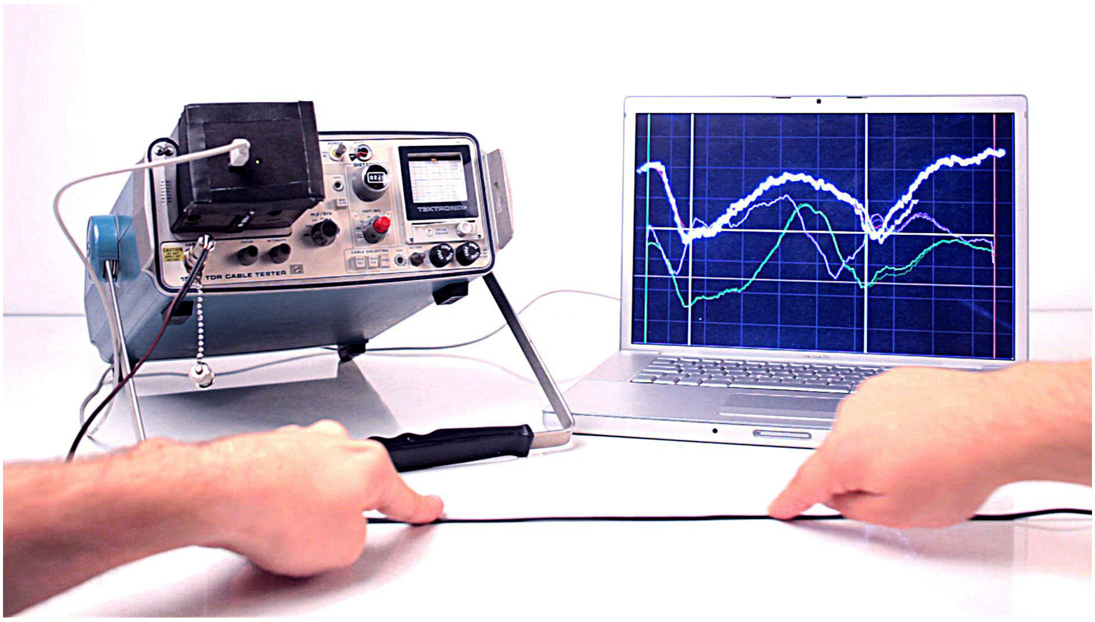

**Figure 7.1:** Time Domain Reflectometry (TDR) is a sensing technique for locating faults along a cable. We utilized the underlying principle for sensing multiple simultaneous touches along a cable. Laying out the cable as a space-filling curve or wrapping it around an object allows for rapidly prototyping touch- and grasp-sensitive artifacts. For our investigations we used a Tektronix 1502 time domain reflectometer for scanning the cable. Its analog screen is digitized using a camera and analyzed live on a laptop.

*In this chapter I present TDRtouch, a grasp sensing technique built on Time Domain Reflectometry (TDR). TDR is a measuring technique originally invented for precisely locating cable faults over long ranges of cable. TDRtouch utilizes TDR for locating multiple simultaneous touches and proximity along a cable. With TDRtouch, arbitrary objects can be made grasp-sensitive by wrapping a cable around the object, or by laying out conductive traces in a two-dimensional pattern. TDRtouch offers several unique properties, e.g., chaining multiple sensing cables together, identifying cables attached to the sensor via embedded markers, and building deformable and stretchable sensor surfaces. Several application scenarios and a quantitative evaluation of the performance offered by our prototypical implementation give an overview of TDRtouch's capabilities.*

**Attribution:** This chapter is based on the UIST 2011 paper "Modular and Deformable Touch-Sensitive Surfaces Based on Time Domain Reflectometry" written by me and Patrick Baudisch (Wimmer and Baudisch 2011). It has been expanded with further details, examples, and analyses. I am the sole author of the technical explanations and historical research. I also conducted and analyzed all performance measurements. However, Patrick Baudisch helped me tremendously in analyzing and refining concept and applications. I am indebted to the following students who helped with the research: Markus Zimmermann wrote the first implementation of the analyzer software, Robin Palleis worked on a *relative calibration* algorithm for TDR (not discussed in this chapter), Corinna Ragutt investigated touch-sensitive papercraft using TDR (not discussed in this chapter), and Christoph Viegener implemented modular touch-sensing tiles. I would also like to thank our anonymous reviewers at UIST 2011, especially the submission coordinator, who gave very helpful feedback. Several pictures from the UIST paper have been reused in this chapter. Photographers are attributed in the figures' captions.

## 7.1 Motivation

While FlyEye offers a high spatial resolution and does not require knowledge about electronics, building large prototypes with it can become very tedious. Therefore, FlyEye inspired the next research goal - finding a prototyping approach for grasp-sensitive surfaces that is both extremely simple and very fast.

Strings and cables have four general properties that make them ideal for prototyping grasp-sensitive surfaces:

- they can be wrapped around almost arbitrarily shaped objects, covering the whole surface area
- they are very flexible, allowing soft, deformable objects to be made touch-sensitive
- they can be easily cut to length, chained together, or stored on a spool
- only a single connection between sensor and cable (complexity = O(1)) is needed

for arbitrarily large surfaces, whereas other approaches require O(2n) or O($n^2$) connections.

With *TDRtouch* (Figure 7.1) we implemented a touch-sensitive *piece of string* by re-appropriating an existing sensing technology - *Time Domain Reflectometry* (TDR).

The remainder of this chapter is organized as follows:

- Section 7.2 gives an introduction into TDR.
- In Section 7.3 I describe how TDR can be used to sense touches on a cable and give an overview of related work.
- Section 7.4 explains why and how TDRtouch is a versatile and powerful tool for prototyping grasp-sensitive surfaces; it presents several examples highlighting the unique properties.
- Section 7.5 contains a description of our implementation, TDRtouch.
- Section 7.6 comprises extensive measurements that give insight into general and implementation-specific properties of TDRtouch.
- Section 7.7 contains a discussion of inherent and implementation-specific strengths and weaknesses of TDRtouch.
- Finally, Section 7.8 concludes this chapter with a discussion of future work.

## 7.2 Time Domain Reflectometry (TDR)

*Time Domain Reflectometry (TDR)* is a sensing technique which was invented in the early 1960s (Oliver 1964). It is used for locating *discontinuities* in electrical conductors, such as cables, PCB traces, or soil. With TDR, such discontinuities are changes in the *characteristic impedance* of the conductor, i.e., changes in resistance or inductance of the wire or changes in conductance or capacitance between a wire pair (Figure 7.2). These discontinuities are often indicators of existing or imminent faults in a cable. Locating them allows for repairing such faults on-site instead of pulling out the whole cable. Therefore, TDR is of essential importance for investigating cables that can not be easily removed for testing, such as wiring in walls or submarine cables.

### 7.2.1 General Principle

A time domain reflectometer[1] consists of two components: a pulse generator and a voltage sampler (oscilloscope). The pulse generator sends a very short electric pulse into a cable (more exactly: a wire pair). This pulse travels along the cable at a certain speed,

[1] in the following referred to as *reflectometer*.

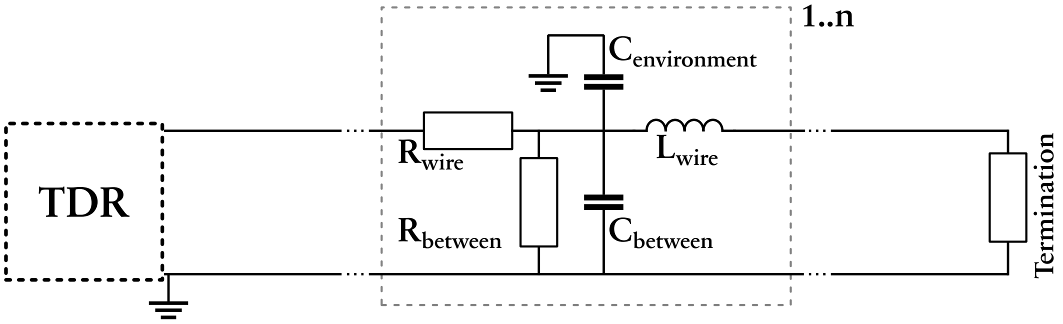


**Figure 7.2:** Lumped circuit model of a terminated transmission line connected to a time domain reflectometer. Its characteristic impedance depends on the resistance and inductance of the signal-bearing wire as well as on the conductance and capacitance between the wire pair. The latter is also influenced by the conductance and capacitance between signal-bearing wire and the environment/earth. Conductance between signal-bearing wire and the environment has been omitted in this figure, as it plays only a minor role for TDR.

the *velocity of propagation* (VoP). For common cables, VoP is about 50% - 70% of the speed of light. At every discontinuity, a small part of the pulse is reflected back towards the reflectometer, travelling at the same absolute speed as the primary pulse. The primary pulse travels further along the cable with slightly reduced amplitude and may cause additional echoes at subsequent discontinuities. All echoes within a time window are captured by the reflectometer's sampling unit. Measuring the time between pulse emission and return of each echo allows determining the distance to each discontinuity. The shape of a returning echo, i.e., the amplitude and width of the pulse, indicates the type of discontinuity.

*Increases* in characteristic impedance result in echoes with a positive amplitude, while *decreases* in impedance result in echoes with a negative amplitude. Typical TDR traces for a cable are shown in Figure 7.3. A break in the cable or an open end results in a complete reflection of the pulse (Figure 7.3b). A short in the cable results in an inverted reflection of the pulse. As the whole pulse gets reflected back in these cases, no further discontinuities along the line can be determined (Figure 7.3c).

TDR is also often called *cable radar* because its principle of operation is very similar to radar. This basic principle has been improved significantly since. For example, spread-spectrum time domain reflectometry (SSTDR) and Frequency-Domain Reflectometry (FDR) increase robustness and expressiveness of TDR.

SSTDR (Smith, Furse, and Gunther 2005) sends spread-spectrum signals into the cable. The reflections are then correlated with the original signal. SSTDR is more robust against noise and can be used for finding faults in *live* wires, i.e. wires simultaneously carrying electrical signals.

Frequency-Domain Reflectometry (FDR) injects signals with different frequencies into

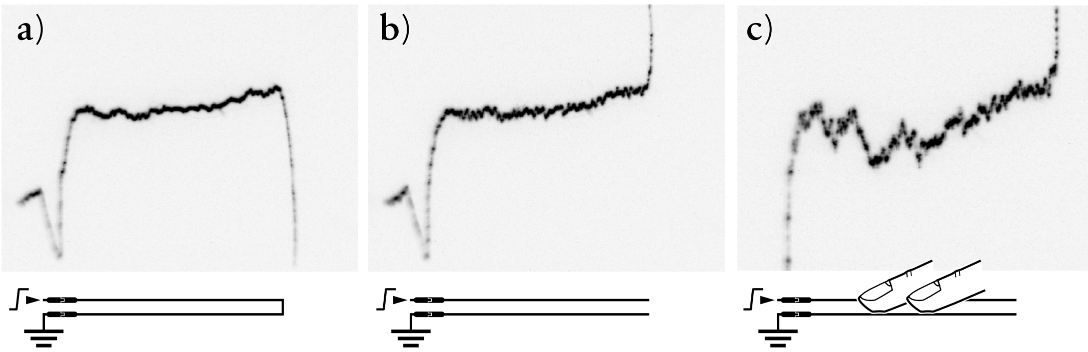


**Figure 7.3:** Typical features in TDR traces: a) wire pair with a shorted end, b) wire pair with a an open end, c) two fingers touching the cable cause two dents in the signal trace. For many cable types, the change effected by touches is not much larger than the noise floor.

the cable and measures the resulting echoes. FDR may be used to gather more detailed information about faults than TDR. However, FDR conducts several hundred or thousand measurements with different frequencies during one scan. Therefore, a single FDR scan takes several seconds, making real-time monitoring of fast changes impossible (Mohr LLC 2012).

### 7.2.2 Range and Resolution

Time domain reflectometry in general allows for locating discontinuities with an accuracy of less than a millimeter (Mohr LLC 2012), and over the range of several kilometers (Howarth, Coates, and Renforth 2006).

Effectively achievable range and resolution of a reflectometer are primarily determined by the capabilities of pulse generator and sampling unit. However, there is a device-independent general trade-off between range and resolution.

While travelling along the cable, the pulse loses power due to the cable's resistance. The amount of loss depends on the cable type. Once the amplitude of the pulse echoes is in the same range as baseline noise, discontinuities can no longer be detected. Therefore, the range of a reflectometer is determined by the electric power of the pulse, which in turn depends on its amplitude and width. The higher the pulse's voltage, and the longer the pulse, the further along the cable a reflectometer can detect discontinuities.

However, in order to detect discontinuities which are located closely together, short pulses are better. If the pulse width is in the same range as the distance between discontinuities, the reflections of multiple discontinuities blur into a single broader reflection.

The steeper the edges of the pulse, the more precisely the exact location of discontinuities can be determined, as it is easier to make out the beginning of a pulse echo.

Therefore, resolution also inherently decreases with distance because the injected pulse gets flatter, i.e., wider and less tall, with distance.

Due to these constraints, pulse amplitude and width need to be optimized either for range *or* resolution, depending on the specific application.

While effective resolution depends on the oscilloscope's sampling rate and depth, the length of the injected pulse (more exactly: its rise time) needs to be in the same order of magnitude as the positional resolution to be achieved. A pulse with a rise time of one nanosecond would have a slope of approximately 0.2 meters. To allow distinguishing discontinuities which are closer together, reflectometers need to generate pulses with rise times in the picosecond range.

Directly sampling pulse echoes in the picosecond range requires an oscilloscope with a gigahertz sampling rate. Such oscilloscopes were not available in the 1960s and are still extremely expensive today. Therefore, early and current time domain reflectometers implement *random equivalent sampling (RES)* (Lee, Sung, and Park 2003). RES exploits the fact that a cable's properties stay more or less constant over subsequent measurements. Therefore, multiple measurements with a low sampling rate can be combined to a high-resolution scan of the cable. To this end, each measuring pass is slightly shifted in time with regard to the previous pass. The achievable effective sampling rate therefore depends on the length of the delay steps.

Using RES, current high-resolution reflectometers such as the Mohr CT100HF achieve an effective sampling rate of 1.3 THz (Mohr LLC 2012). On the other end of the product spectrum, long-range TDR employs pulses with voltages of up to 70 V, achieving ranges of 100 km with an accuracy of 150 m (Howarth, Coates, and Renforth 2006).

### 7.2.3 Pulse vs. Step Signals

Instead of a short pulse, many reflectometers inject a step signal, i.e., the 'pulse' only has a rising edge. The voltage level is kept constant during each measurement pass and only returns to 0 V after the measurement. Therefore, not complete pulses but voltage steps are reflected back at discontinuities.

While step-based TDR is slightly less intuitive than pulse-based TDR, it offers several advantages (AEA Technology Inc. 2013; Mohr LLC 2012):

- for the resolution of such measurements only the rise time is important, not the pulse length.
- step-based TDR suffers significantly less from multiple reflections at cable ends.
- step-based TDR achieves a better signal-to-noise ratio than pulse-based TDR because more power can be sent into a wire.

- step-based TDR offers a shorter *dead zone*, the length at the beginning of the cable where no discontinuities can be detected.
- the most interesting advantage of step-based TDR is that the amplitude of the 'returning' voltage level is proportional to the impedance along the cable. This makes it easier to distinguish different types of discontinuities and allows for directly determining the characteristic impedance of any cable segment.

For these reasons, step-signal reflectometers are prevalent now. All figures in this chapter have been captured from a step-signal reflectometer.

### 7.2.4 Applications of Time Domain Reflectometry

Since its inception, TDR is primarily used for locating wire faults - most importantly shorts and breaks - in cables which are buried or otherwise inaccessible, such as submarine cables (Worzyk 2009, 146) or cabling in ships (Mohr LLC 2009) and aircraft (Smith, Furse, and Gunther 2005).

As conductance and capacitance between the wire pair depend on the properties of the surrounding environment, changes in characteristic impedance also provide information about changes in the cable's environment.

This opens up a wide range of sensing applications. For example, changes in characteristic impedance may hint at water intrusion into the cable. TDR is also used to measure soil moisture (Malicki et al. 1992). Helically wound cables embedded into concrete also allow for measuring strain and movement in bridges and buildings (Sun et al. 2009). High-resolution TDR setups are also capable of detecting faults in microchips (Han et al. 2005).

Most interestingly for HCI researchers and practitioners, TDR also allows for building versatile touch sensors.

## 7.3 Using TDR for Touch Sensing

### 7.3.1 History and Related Work

In one of the very first publications on time domain reflectometry, Oliver (1964) already mentions that TDR is also sensitive to touches:

> *"In exposed circuits one can touch the line to produce an added echo. Then, by running the point of contact along the line till this added echo coincides with the*

> *system echoes, one can literally put his fingers on the troubles. In a coaxial cable, one can produce a reflection by squeezing the cable."* (Oliver 1964)

While Oliver omits a detailed explanation of this effect, the principle behind it is straightforward: Touching both exposed wires of a transmission line increases both conductance and capacitance between them, thereby decreasing the characteristic impedance of the touched cable segment. Therefore, a touch shows up as a dent in the signal trace. In effect, TDR-based touch sensing can be seen as a sophisticated *transmit mode* capacitive sensing technique. While this effect has apparently been used early on to quickly and interactively locate discontinuities, it took several decades until researchers recognized its potential for touch sensing.

In 1992, Kozik and Taylor (1992) were granted a patent on an "Apparatus to determine coordinates"[2]. The patent - which expired in 2004 - succinctly describes but never explicitly mentions TDR:

> *"In a preferred embodiment, the signal conducting line is a strip of copper etched in a serpentine pattern on a plastic material. A ground line is etched adjacent to and parallel with the signal conducting line so that a discontinuity can be formed by touching the two lines with a finger or a conductive member. Additionally, the signal conducting line has a characteristic impedance and a load that matches the characteristic impedance is connected to the second end of the signal conducting line. As a result of impedance matching, unwanted reflection in the signal conducting line is minimized. In one preferred form of operation, a pulse is generated at the first end of the line and a first portion of the pulse energy is propagated down the line and a second portion of the pulse energy passes through the counter to start a count. When an impedance discontinuity is developed by a finger touching the apparatus surface, a pulse is reflected back down the line to the counter. The reflected pulse stops the counter at the program identifiable count so that a specific action can be taken by a computer system in response to the program identifiable count."* (Kozik and Taylor 1992)

While the patent text correctly describes the general principle and properties of TDR, it does not offer any information about actual feasibility or required sensing resolution. The approach presented in the patent also only supports single-touch operation. Later patent applications for TDR-based touchscreens also do not discuss the feasibility of the approach[3].

[2] This patent predates Collins' patent (Collins 2003) which we presented as the first patent on TDR-based touch sensing in our UIST paper.

[3] e.g., *Touch sense determined by characterizing impedance changes in a transmission line*, US 20120271580 A1

Huang, Hung, and Liu (2009) were the first to demonstrate that TDR is indeed a suitable technique for touch sensing. They etched a co-planar waveguide[4] of about 300 cm length laid out in serpentines into an A4-sized copper-plated circuit board. Using a *Tektronix TDS 8000* oscilloscope, Huang et al. were able to detect the location of a single touch along the sensing line at a distance of up to 1680 mm. They also note in passing that locations of multiple simultaneous touches can be detected. No formal measurements have been conducted, however. In personal communication[5], Huang added that he did not intend to pursue this topic further. He also pointed out that first experiments with ITO[6] had been unsuccessful because its high resistivity quickly weakens the signal.

Both Kozik et al. and Huang et al. propose TDR as a novel sensing technique for touch screens. However, it is unlikely that TDR with its quite complex sensing hardware will be able to replace existing capacitive sensing techniques. For example, touchpad manufacturer Synaptics had been exploring the use of TDR for touchpads but deemed it too expensive to be used commercially[7].

Nevertheless, TDR's unique properties actually allow for a much wider application spectrum. While current touch screens require multiple electrical connections on at least two of their four edges, TDR only requires two wires attached to only one side of the substrate. This allows for building smaller, modular touch-sensitive surfaces with very little effort. Furthermore, the cable or coplanar waveguide may have an arbitrary shape and curvature. This allows for building non-planar, deformable grasp-sensitive surfaces. Du to its great flexibility and physical malleability, TDR-based grasp sensing offers itself as a versatile rapid prototyping technique.

In our research, Patrick Baudisch and I first systematically investigated general properties, limitations, and interesting applications of TDR for touch and grasp sensing. In addition, I conducted measurements to determine relevant quantitative properties and limitations of TDR. Our approach - described in the remainder of this chapter - is called *TDRtouch*. It comprises a reference implementation and a set of design guidelines.

### 7.3.2 Cable Choice for TDR-based Touch Sensing

The choice of cable strongly determines the achievable resolution and robustness for touch sensing and grasp sensing. However, none of the previous publications has discussed properties and limitations of different cable types.

---

[4] A *coplanar waveguide* is a flat transmission line consisting of a central, signal-bearing line and two grounded lines - one on each side. Such transmission lines are often etched into circuit boards. The grounded lines shield the signal line from crosstalk by adjacent signal lines.

[5] Chi-Fang Huang, personal communication, 22.02.2010 and 28.08.2010

[6] Indium-tin oxide (ITO) is a transparent, conductive coating used primarily for circuit traces and electrodes in touch screens.

[7] Patrick Worfolk, Director of Research for Synaptics, personal communication at UIST 2011 (October 2011).

Touching an insulated wire pair introduces a discontinuity at the touch location as the touching finger capacitively couples to both wires and therefore increases the capacitance between them (Figure 7.4a). When touching an uninsulated wire pair - as described by Oliver (1964) - the touch not only increases capacitance between wires but also the conductance between them. How much the conductance increases depends on the skin conductivity of the touching finger. However, skin conductivity varies greatly between persons and across different environments. In addition, increased conductance also increases power loss along the cable, reducing achievable sensing range. Therefore, touch sensing works best without direct electrical contact between finger and wire pair. The following explanations focus on capacitance changes effected by touching an *insulated* wire pair.

To reliably distinguish between touches and ambient noise, it is necessary to maximize the *relative* increase in capacitance caused by a touch. To this end, the base capacitance between wires should be minimized, whereas the capacitance added by the touching finger should be maximized.

The basic electrical rules for a capacitor show that three cable properties can be optimized[8]:

- the larger the *distance* between two wires, the lower the capacitance between them,
- the smaller the *area of the wires facing each other*, the lower the capacitance between the wires,
- the larger the wire surface covered by a touching finger, the greater the capacitance change caused by it.

Therefore, a flat ribbon cable, conductive copper traces on a circuit board, or parallel strips of copper foil are most suitable for TDR-based touch sensing (Figure 7.4b). Circuit traces should have mitered corners to avoid unwanted reflections.

Ideally, the wires/traces are spaced about 10-20 mm apart so that they can be simultaneously touched by a single finger. However, wide traces increase capacitive coupling to the environment (stray capacitance), causing the pulse to lose power more quickly. This effect substantially reduces the achievable sensing range along the cable. Therefore, cable choice requires a trade-off between range and resolution.

In addition, conductive objects very close to the cable can increase capacitance between the wires. While this effect is mostly neglectable in practice, affixing the cable to a metal surface or wrapping it around a metal object causes its characteristic impedance to vary wildly depending on how close to the surface each part of the cable is. Therefore, conductive objects are less suitable for being augmented with TDR-based touch/grasp sensing.

[8] The dielectric properties of the material between wires also affect the base capacitance. As it cannot be easily determined or changed by the user, this effect is ignored in the following.

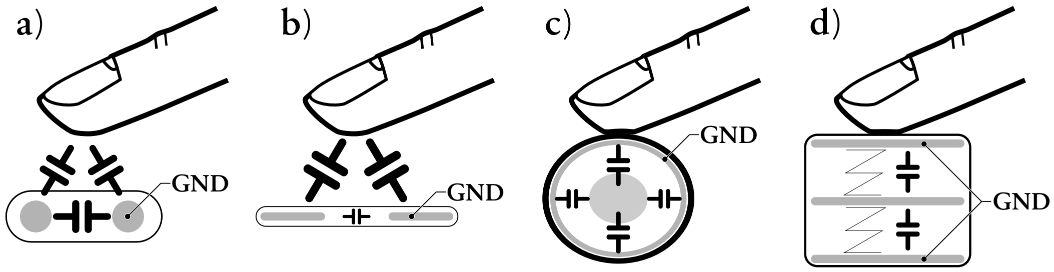


**Figure 7.4:** Cable shape determines how strongly a touch increases capacitance between wires of a transmission line. The lower the intrinsic capacitance between both conductors, the higher the capacitance change effected by a touch. Therefore, wide, flat cables (b) are better suited for touch sensing than standard cables (a). Coaxial cable (c) effectively shields the inner conductor from external capacitive coupling, making it unsuitable for touch sensing. However, pressing with sufficient force deforms the cable, changing the capacitance between central wire and shielding, allowing force sensing. This effect is even stronger in a multi-layer substrate comprised of copper foil conductors separated by a compressible material (d). In order to reduce stray capacitance, the signal-bearing layer may be sandwiched between to grounded layers, similar to a coplanar waveguide.

While capacitive coupling via body and environment also increases the capacitance between signal-bearing wire and grounded wire, the effect is very small and mostly neglectable due to high resistance of the body. However, this effect can be exploited to enable a less robust single-touch mode on a single wire (Wimmer and Baudisch 2011). As this effect has not been described previously, it is explained in the following.

Modeling electrical properties of TDR is outside the scope of this thesis. The following abstract explanation gives an - hopefully correct - overview of the principle of operation of single-wire TDR touch sensing. In this special case, the standard transmission line model still applies. However, the electric field surrounding the signal-bearing wire is no longer *captured* (or *focused*) by a second wire or shielding. Therefore, signal amplitude decreases rapidly with distance, severely limiting sensing range in this mode of operation. Also, the electric field is much more diffuse, no longer allowing to distinguish changes in capacitance caused by single touches. However, a finger touching the bare wire grounds it at the touch location, causing a change of resistance at the touch location. The smaller the distance between reflectometer and touch, the faster the amplitude of the TDR trace decreases, allowing for tracking a single touch location along the wire.

Various informal experiments with different cables and setups have shown that one-wire TDR requires a bare wire to be touched, can only sense a single touch at a time, and has a limited range of only a few meters of wire. It is not possible to detect mere proximity to the wire. Environmental properties can have a significant influence on sensing performance. For example, a big grounded object near the wire may make it impossible to reliably track a single touch[9].

While one-wire TDR offers only very limited range, resolution, and reliability, it enables applications where a single touch needs to be tracked on a single wire, such as for a touch-enabled guitar string.

Coaxial cable (Figure 7.4c), where the inner conductor carries the signal while the surrounding conductor is connected to ground, significantly reduces such signal loss. In this arrangement, the outer layer shields the inner conductor against external electric fields. Therefore, the inner conductor is also shielded from capacitive coupling to the environment, preventing power loss. This setup does not allow for detecting touches because a touching finger cannot increase capacitance between inner and outer conductor. As the pulse is usually inserted in the inner conductor, *shunt mode* capacitive sensing is not applicable here. Nevertheless, tightly squeezing the cable reduces the distance between inner and outer conductor, creating a small discontinuity. As substantial force needs to be applied for deformation, normal coaxial cable is not suitable for touch sensing. However, the underlying effect, already mentioned in passing by Oliver (1964), can be magnified by dual-layer or triple-layer substrates (Figure 7.4d). Such cables consist of two conductive layers, such as copper foil, separated by a deformable spacing layer, such as silicone or rubber foam. Pressing downwards on such a cable brings the conductive layers closer together, increasing their capacitive coupling. Such cables are not touch-sensitive but highly force-sensitive. Sandwiching the signal-bearing layer between two grounded layers reduces stray capacitance compared to a dual-layer substrate.

In summary, cable choice strongly depends on the intended application. Large coplanar waveguides, such as parallel strips of copper foil, offer the highest sensitivity but only limited range. The smaller the distance between parallel wires, the better the resolution of adjacent touches. Coaxial cable and dual-/triple-layer substrates offer long sensing ranges but are only sensitive to force, not to touch.

Detailed measurements for several cables are reported in Section 7.6.

### 7.3.3 Covering Surfaces

TDR traces capture impedance changes along the cable, i.e. only in one spatial dimension. For grasp-sensitive surfaces, two-dimensional sensor arrays are necessary.

Traditionally, touch-sensitive surfaces are implemented either as an array of individual sensors or as matrix of sensing lines which form virtual sensors at each crossing.

[9] In one experiment, I was able to track a finger sliding along the temple of eyeglasses using one-wire TDR. However, this worked only with glasses lying on a table. When the glasses were worn, the large capacitive coupling between head and eyeglasses made it impossible to sense touches anymore.

TDRtouch offers and requires a third sensor layout. By laying out the cable in serpentines or other *space-filling curves*, such as Hilbert Curves (Hilbert 1891), a single cable can cover a two-dimensional area (Figure 7.5). This allows for making arbitrarily shaped surfaces touch- and grasp-sensitive. Cylindrical objects may also be covered seamlessly (Figure 7.6) - which is not possible with traditional sensor designs (see e.g., Song et al. 2011).

a) b)

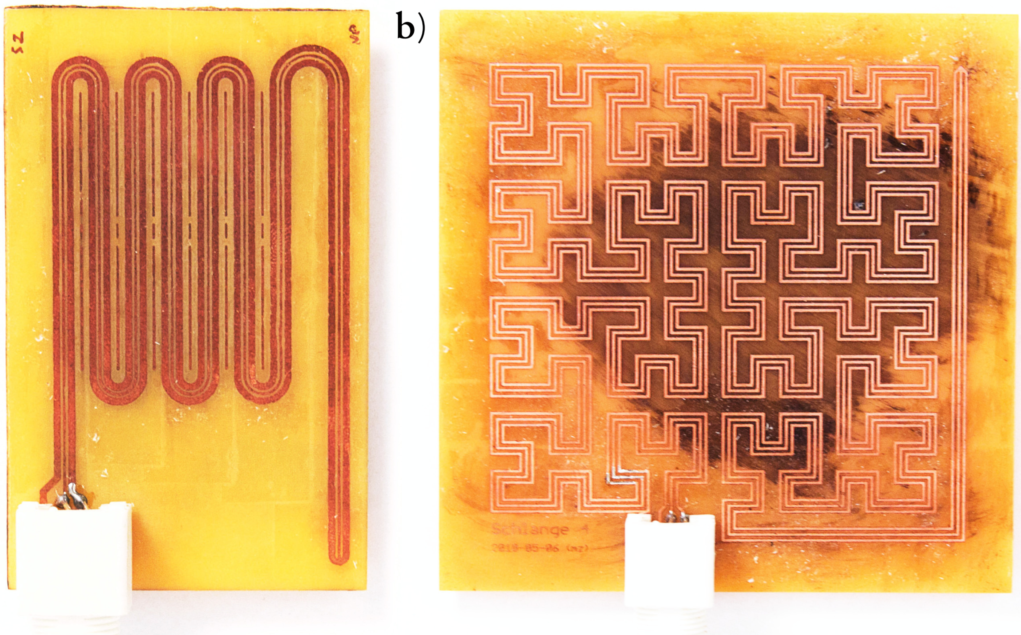

**Figure 7.5:** Two prototypes showing how surfaces may be covered by laying out coplanar waveguides in serpentines (a) or a Hilbert curve (b).

Unfortunately, TDRtouch does not offer a uniform resolution in both dimensions. While sensing along the cable allows for a resolution in the millimeter range, resolution in the orthogonal direction depends on the spacing of the cable segments that make up the surface. The closer these are to each other, the higher the resolution. However, when cable segments are placed too close to each other, capacitive coupling induces crosstalk in adjacent wires, causing multiple time-shifted pulses to spread along the cable. As a rule of thumb, parallel cable segments should be spaced at least one cable width apart. This results in a practical resolution of two times the cable width in the direction orthogonal to the cable.

Therefore, cables must be spaced far enough apart from each other to reliably avoid such crosstalk. In our experience, this spacing should be at least two times the cable's width. Coplanar waveguides surround the signal wire with ground wires on both sides. While this reduces crosstalk, it also increases the width of the cable. Overall, we did not see significant advantages using coplanar waveguides instead of appropriately spaced two-wire cables. Crossing cables also introduces crosstalk and erroneous touch detection and

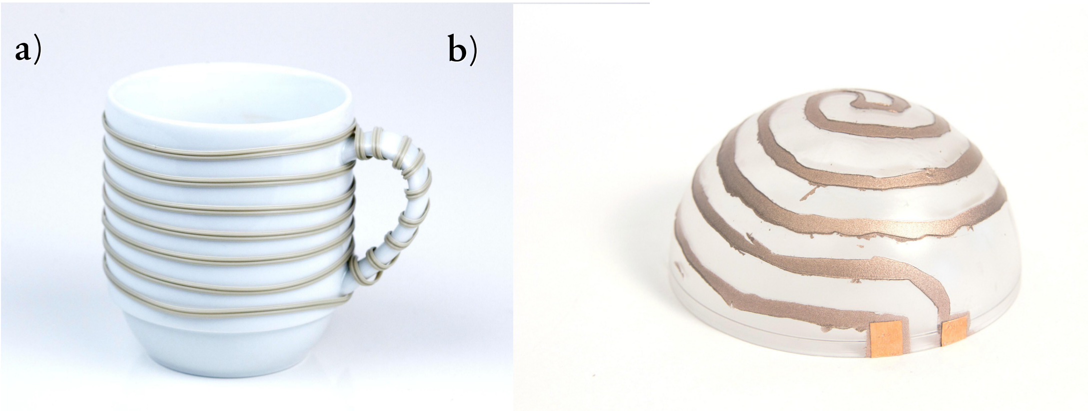


**Figure 7.6:** Flat or nonplanar surfaces may also be covered in spirals: a) a cup made grasp-sensitive by wrapping a cable around it; b) non-planar dome made touch-sensitive with a spiral of conductive paint

should be avoided.

Of course, requiring ample space between parallel cables reduces resolution orthogonal to the cable directions. This is a major inherent limitation of TDRtouch that can not easily be resolved. However, interpolation between touches sensed on adjacent cable segments can mitigate this problem. Using Hilbert curves instead of serpentines allows for a more uniform resolution in all directions.

## 7.4 TDR as a Prototyping Technique for Grasp-Sensitive Objects

TDRtouch is designed to fulfil all three requirements for a grasp-sensitive surface, as outlined in Section 4.4.2:

**Deformable Sensors**: TDRtouch requires only a single two-wire cable or parallel conductive traces. This allows for creating nearly arbitrarily shaped sensor areas and for instrumenting non-planar and deformable surfaces.

**Sensing Shapes**: TDRtouch is able to resolve fingers in close proximity to each other. Furthermore, contiguous touch areas show up as long dents in the TDR trace. While TDRtouch only has limited resolution in the direction orthogonal to the cable direction, this can be compensated by appropriate cable layouts, such as Hilbert curves.

**Appropriate Representation**: TDRtouch reports touch/proximity along a cable. By transforming and correlating this raw data, different representations of the applied grasps may be extracted. The spatial layout of the cable may be determined manually or using an automatic calibration approach as described for FlyEye.

In addition, TDRtouch offers a number of unique features that make it very suitable for prototyping:

TDRtouch allows for building extremely thin and small sensors, can make large areas touch-sensitive using a single cable, facilitates deformable sensors, allows for daisy-chaining multiple sensor elements, and offers a way to distinguish between different attached cables. These features are described in more detail in the following sections.

## 7.4.1 Building Extremely Thin Sensors

With traditional capacitive sensors, such as CapToolKit, the sensor IC needs to be close to the sensor electrode or have a shielded connection to it. For each sensor electrode a separate connection to the controller is necessary.

TDR requires only a single two-wire cable that does not need to be shielded. This allows, e.g., for turning arbitrary cables into touch sensors. To demonstrate this, I made an unmodified headphones cable touch-sensitive, allowing control of music playback by touching certain positions on the cable. To avoid damaging our sensitive reflectometer, no audio signal was simultaneously sent to the headphones, however.

Thin flexible flat cables (FFC) may be wrapped around or glued onto curved surfaces without much effect on the object's shape. This allows for augmenting tangible artifacts with grasp-sensitive areas without affecting their shape (Figure 7.7).

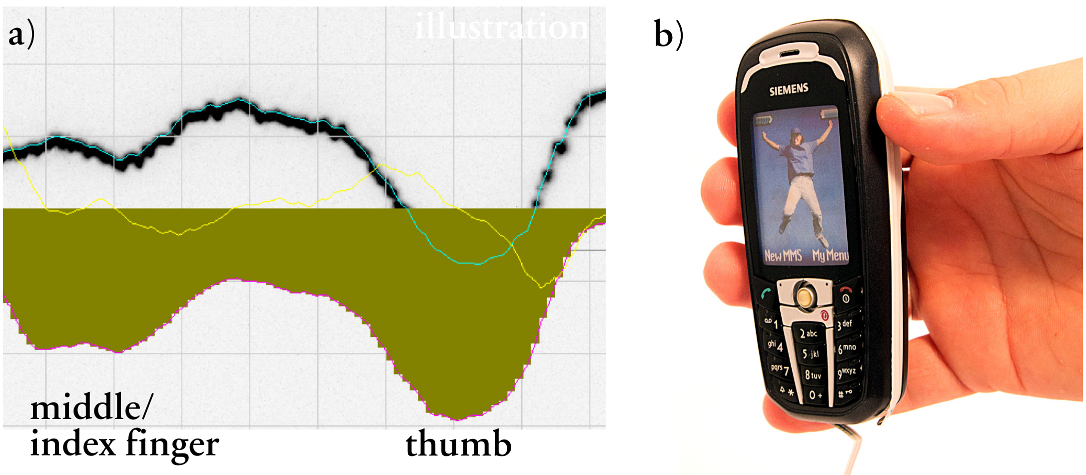


**Figure 7.7:** a) Exemplary TDR trace for b) a mobile phone dummy that has been made grasp-sensitive using a strip of flexprint cable. (The TDR trace has been recreated using the same prototype at a later date than the photo. It has been inverted, flipped, and cropped. Ring finger and small finger are not touching the phone in the TDR trace.) (Photo: Doris Hausen)

With TDRtouch, only two wires are necessary for forming a touch-sensitive surface of arbitrary size and connecting it to the reflectometer (Figure 7.8c). In *Big-O notation*, generally used to denote the complexity of algorithms, TDRtouch could be characterized as having a complexity of $O(1)$, i.e. the number of connections between controller and sensing area is independent of the size or resolution of the sensing area. Common two-dimensional capacitive touch sensor designs require $O(n^2)$ or $O(n)$ connections for capturing touches at $n \times n$ locations (Figure 7.8a, b). Routing those connections without affecting signal quality requires much experience and effort[10].

Thus, TDRtouch allows for building smaller sensors and reduces the time needed for routing connections. The overall length of electrodes is similar for all three approaches. However, as TDRtouch only employs a single measuring circuit for the whole surface, the average distance between measuring circuit and touch location (along the electrode) is much larger for TDR than for the other two approaches. As resolution of capacitive sensing approaches decreases with distance along the electrode, TDRtouch inherently offers a lower physical resolution than a capacitive touch sensor using a block or matrix layout.

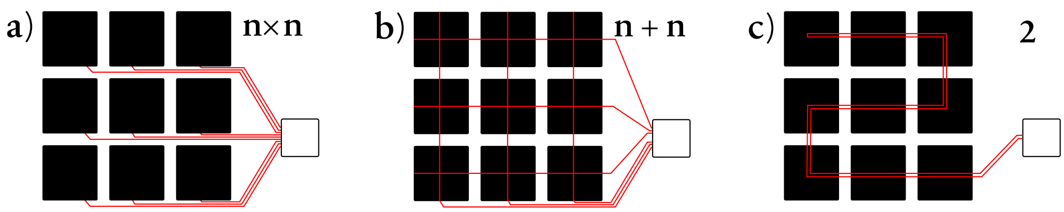


**Figure 7.8:** a) Connecting an $n \times n$ array of individual capacitive sensors to a controller requires $O(n^2)$ connections; b) to reduce the number of connections, common capacitive touchscreens and touchpads use a matrix of horizontal and vertical electrodes, resulting in $O(n)$ connections between sensor area and controller; c) in contrast, TDRtouch requires only two, i.e. $O(1)$, connections.

Furthermore, while TDR usually requires a cable with two parallel wires - or a central wire surrounded by a shielding, we found out experimentally that TDR also allows locating a single touch on a single, *bare* wire, as described in the previous section.

## 7.4.2 Sketching Sensors

Prototyping interactive artifacts is both iterative and highly manual. Sketching user interfaces is an established practice in an interaction designer's toolkit. As TDRtouch only requires two parallel conductive wires or metal traces, it allows for sketching not only

[10] See, e.g., the progress reports for an open-source capacitive touchscreen controller at `http://www.wiretouch.net/`.

graphical but also touch-sensitive elements of a physical user interface (Figure 7.9). Designers can draw traces with conductive ink, use strips of copper foil, or print complex traces using conductive ink[11]. This speeds up prototyping compared to soldering or attaching individual sensors to a surface.

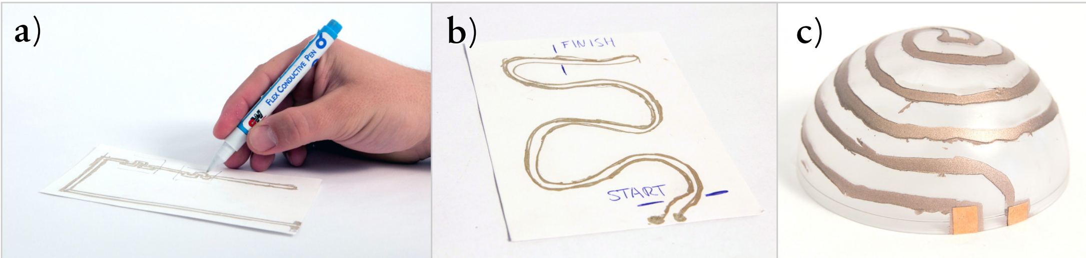


**Figure 7.9:** With TDRtouch, designers can easily sketch arbitrarily shaped touch sensors: a/b) drawing a simple user interface using a conductive ink pen; c) a spiral-shaped touch sensor created by applying conductive paint over a stencil.

## 7.4.3 Deformability

A deformable touch sensor is of twofold advantage when prototyping grasp-sensitive objects. First, it allows for augmenting deformable, soft objects with touch-sensitivity. Second, it eases prototyping, as the touch-sensitive surface does not have to be custom-fitted to the object.

Building stretchable electronic circuits is not trivial, however. One option is to use stretchable materials such as conductive yarn or carbon nanotubes that may be stretched to about twice their length (Rogers, Someya, and Huang 2010). However, connecting multiple sensors with stretchable, conductive yarn is tedious and requires custom mechanical connectors because conductive yarn cannot be soldered to other components. Stretchable nanomaterials are just starting to become commercially available. As an alternative to stretchable materials, one may also achieve stretchability at a larger scale by laying out circuit traces in serpentines. This allows stretching in the direction perpendicular to the serpentines. Regardless of the scale in which stretchability is implemented, there is no flexible replacement for most electrical components, such as sensing ICs.

TDRtouch works with both intrinsically stretchable and stretchably arranged conductive materials. As no further components need to be attached to the cable, TDRtouch allows for building completely deformable grasp-sensitive surfaces. To demonstrate this, we implemented several prototypes of stretchable sensors made of various materials (Figures 7.10, 7.11, 7.12). While a serpentine pattern allows stretching only in one direction, two-dimensional deformability may be achieved by laying out the cable in a Hilbert Curve pattern (Figure 7.12).

[11] e.g. via `http://www.inkjetflex.com/`

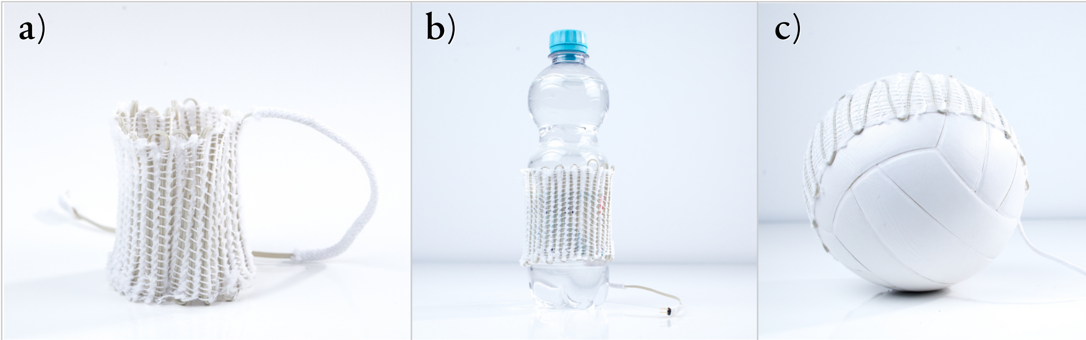


**Figure 7.10:** A stretchable wristband with embedded wire serpentines (a) can be pulled over objects of different shapes and sizes (b/c) in order to make them touch-sensitive. However, resolution orthogonal to the cable direction shrinks with increasing diameter.

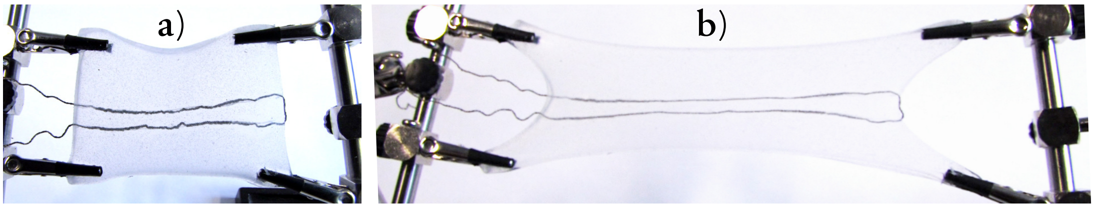


**Figure 7.11:** A touch sensor made of silicone and conductive yarn can be stretched to twice its length. Strands of conductive yarn were inserted into the stretched silicone sheet with a needle. Relaxing the sheet compresses the yarn, allowing it to be stretched again.

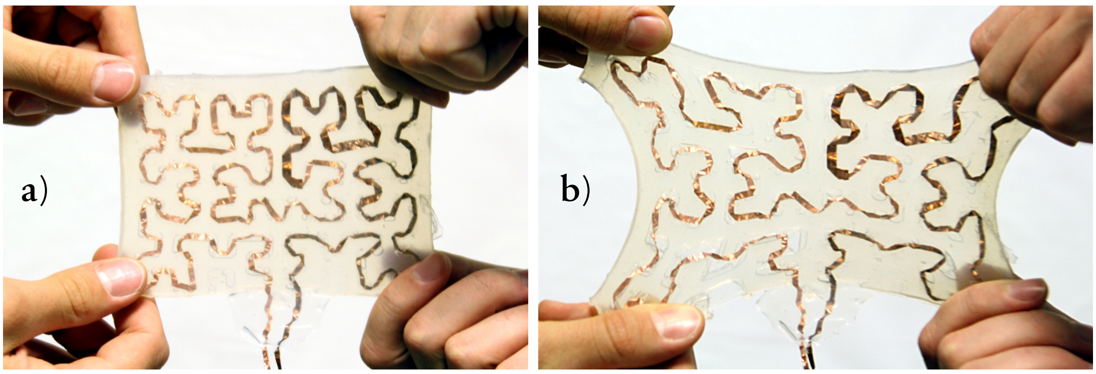


**Figure 7.12:** Space-filling curves - such as Hilbert Curves - allow for creating deformable, stretchable touch-sensitive surfaces. Here, two layers of copper foil - divided by a thin silicone layer - are enclosed in a silicone sheet.

### 7.4.4 Modularity

With TDR, multiple lengths of cable can also be daisy-chained to form a larger touch sensor. This allows for quickly extending the touch-sensitive area, and makes it easy to connect multiple sensor areas to one reflectometer. To demonstrate this property, we have built several prototypes (Figures 7.13, 7.14). Embedding copper foil in masking tape (Figure 7.13a) results in a flexible, tearable touch sensor that may be easily attached to arbitrary surfaces. By connecting the copper foil traces of multiple strips of masking tape, complexly shaped touch-sensitive surfaces can be achieved.

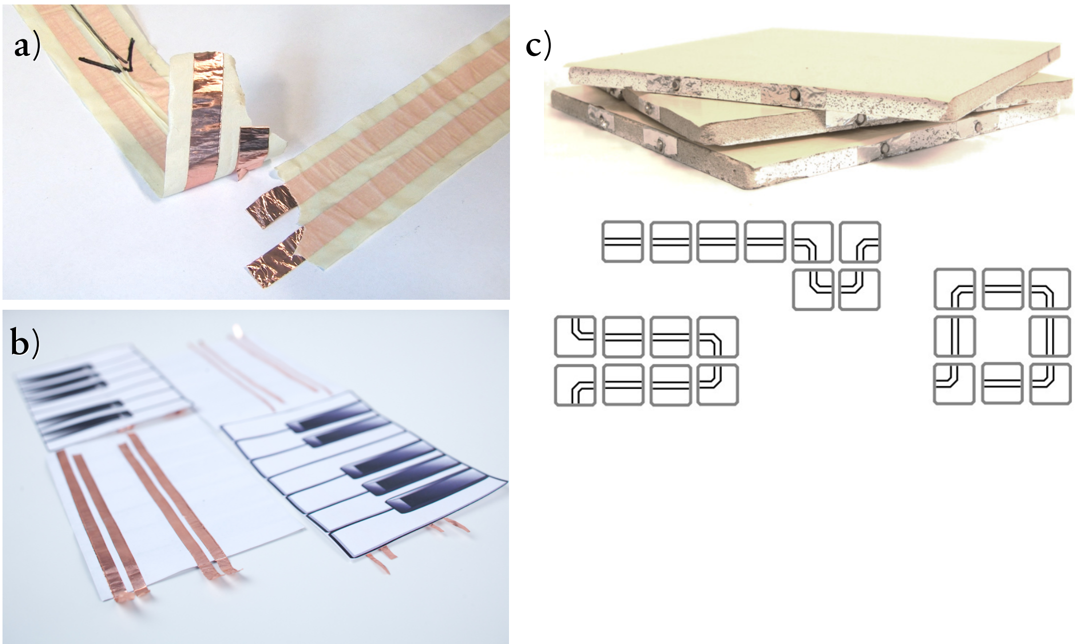


**Figure 7.13:** TDR allows daisy-chaining multiple sensing cables. To evaluate and demonstrate this concept, we built a) masking tape with embedded copper foil traces, b) a paper piano that can be extended with additional octaves, c) floor tiles for locating people's position in a room.

Modules can be connected using magnets, conductive ink, conductive glue, or mechanical connectors. While theses connectors show up as dents in the signal traces, they are static and are removed by the background subtraction pass of our software. As the parallel traces within such tiles often have different lengths, the distance between the pulses in signal line and ground line increases with length, weakening the pulse. This effect can be mitigated by choosing tile layouts where both parallel traces have the same overall length.

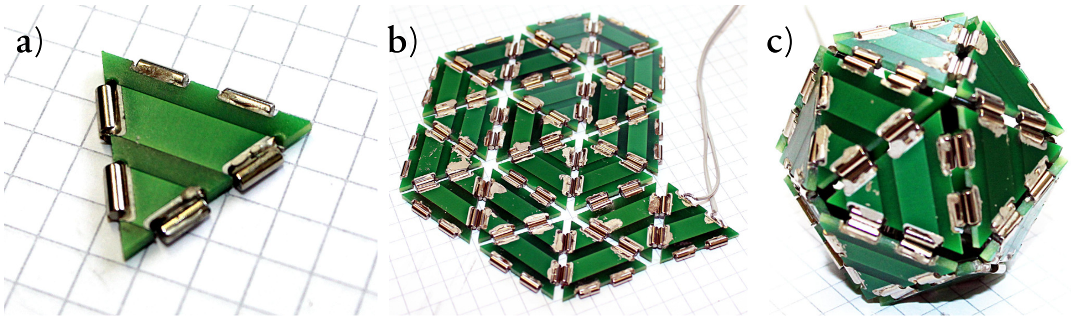


**Figure 7.14:** Tiny triangular tiles (a, side length: 20 mm) with parallel conductive traces and magnetic connectors form flexible, flat (b) or rigid, three-dimensional (c) touch-sensitive structures. In the pictured prototype, multiple magnets were slightly misaligned, resulting in bad electrical contact between tiles. Therefore, it is not clear how well these tiles will work.

### 7.4.5 Identification

Modularity allows for quickly exchanging and combining sensing cables. However, different sensing cables have different layouts, characteristic impedances, VoPs, or are designed for different applications. Therefore, a cable needs to be calibrated on first use in order to determine the relationships between touch locations on the cable and minima in the signal trace. When prototyping touch-sensitive surfaces, re-calibration may become tedious. We solved this problem by attaching unique identification (ID) markers to sensing cables.

These markers are simply thin strips of copper foil which are wrapped around the cable (Figure 7.15). As they increase capacitive coupling between the wire pair, they show up as dents in the TDR trace. Assigning each sensing cable a unique dent pattern allows for automatically loading calibration data and custom applications on connecting that cable to the reflectometer.

### 7.4.6 Emulating Lower Resolutions

The use cases presented in this chapter demonstrate that TDRtouch can be versatile prototyping tool. However, some of its current limitations - mainly size and cost - require designers to consider smaller and cheaper commercial touch sensors for mass production. TDRtouch may also be of use in this regard, assisting in selecting sensors with appropriate resolution. Its high spatial resolution allows on-the-fly downsampling to a (virtual) lower spatial resolution by integrating over subsegments of the TDR trace (Figure 7.16). Thus, designers can interactively determine the required resolution for

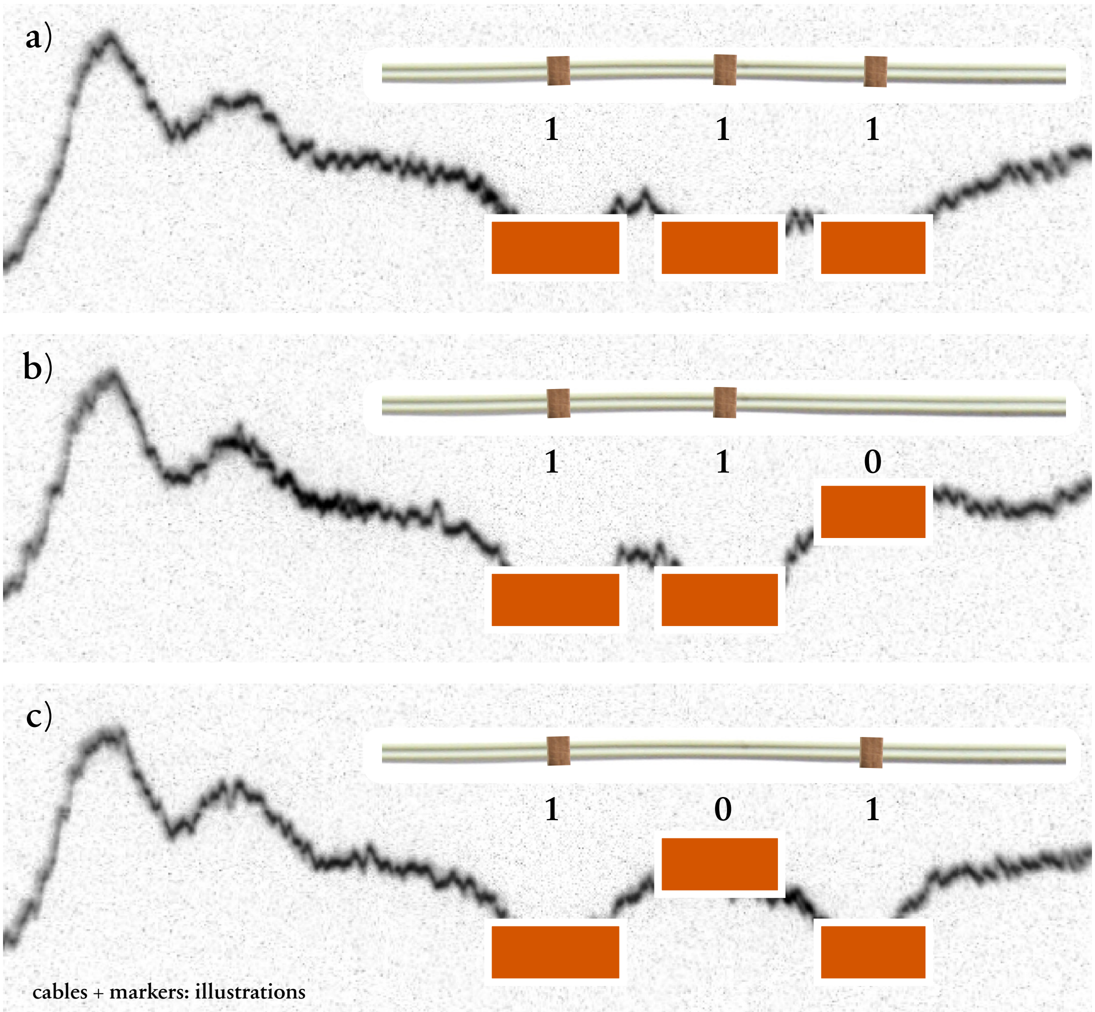


**Figure 7.15:** Different cables can be automatically identified by ID markers attached to the start of the cable. These markers are made of copper foil and show up as unique dent patterns in the signal trace.

reliably detecting certain grasps.[12]

## 7.5 Our Low-Cost Implementation

We used an analog Tektronix 1502 time domain reflectometer for our research (Tektronix 1986). Its cathode ray tube (CRT) output is digitized using a Point Grey FlyEye camera. The captured TDR traces are then filtered and analyzed using a custom Python frame-

[12] In practice, TDRtouch may not accurately simulate touch sensors if their spatial sensitivity falloff is steeper than the reflectometers rise time.

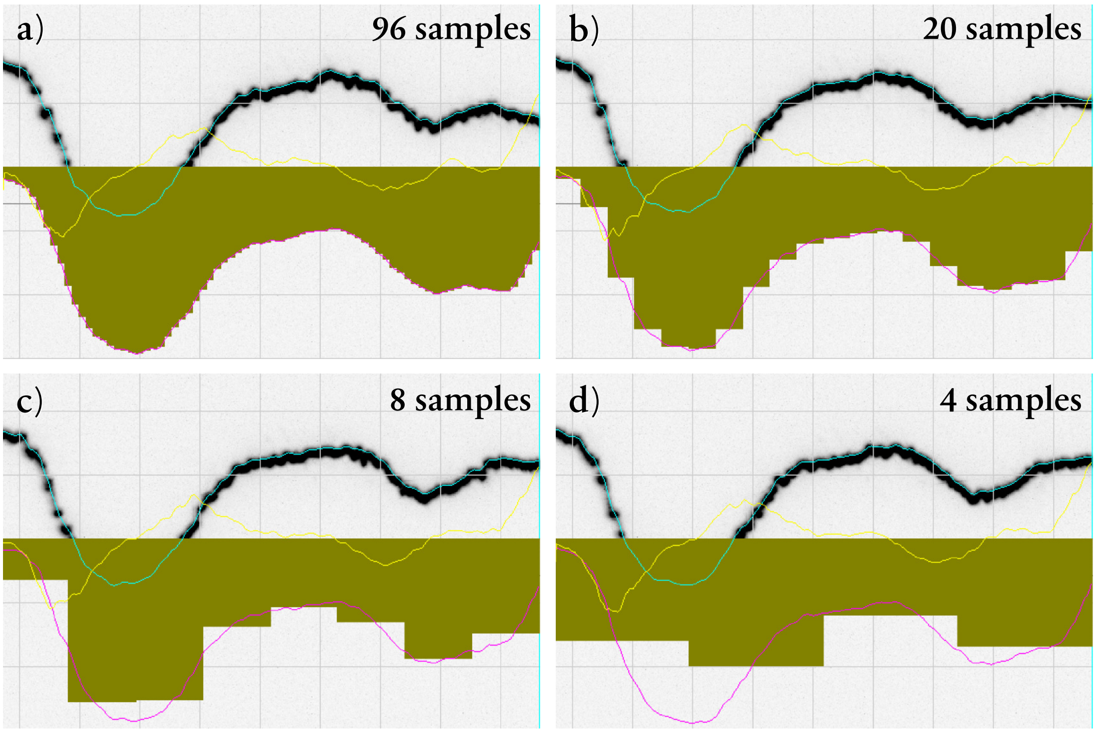


**Figure 7.16:** With TDRtouch, designers can interactively determine the minimum spatial resolution required for reliable grasp recognition. Arbitrary lower spatial resolutions may be simulated by integrating over subsegments of the TDR trace. Figures a-d show various virtual lower resolution calculated from the same TDR signal. While the virtual sensor resolutions in a-c) are sufficient for identifying the specific grasp used for holding a mobile phone (see Figure 7.7), the resolution in d) is obviously too low to discern individual minima. (All screenshots have been inverted and cropped.)

work (Figure 7.17).

### 7.5.1 Hardware

The Tektronix 1502 is one of the few[13] time domain reflectometers with real-time operation and high spatial resolution (Clarke 2003). It had been originally developed for the US Navy in the early 1970s (Clarke 2003; Mohr LLC 2009).

The Tektronix 1502 is a step-signal time domain reflectometer with a maximum range of 2000 ft (approx. 600 m) (Tektronix 1986, 1–3). It achieves a relatively high spatial resolution by employing *random equivalent sampling* as described above. Unlike its partially

[13] While a few newer devices with similar resolution exist, the cheapest one, MOHR CF100HF (Mohr LLC 2009), started at about EUR 18,000 (MOHR offer from 2011). As we did not know beforehand whether TDR was indeed usable for reliable touch sensing, we opted for buying a refurbished Tektronix 1502 at eBay for EUR 290.00.

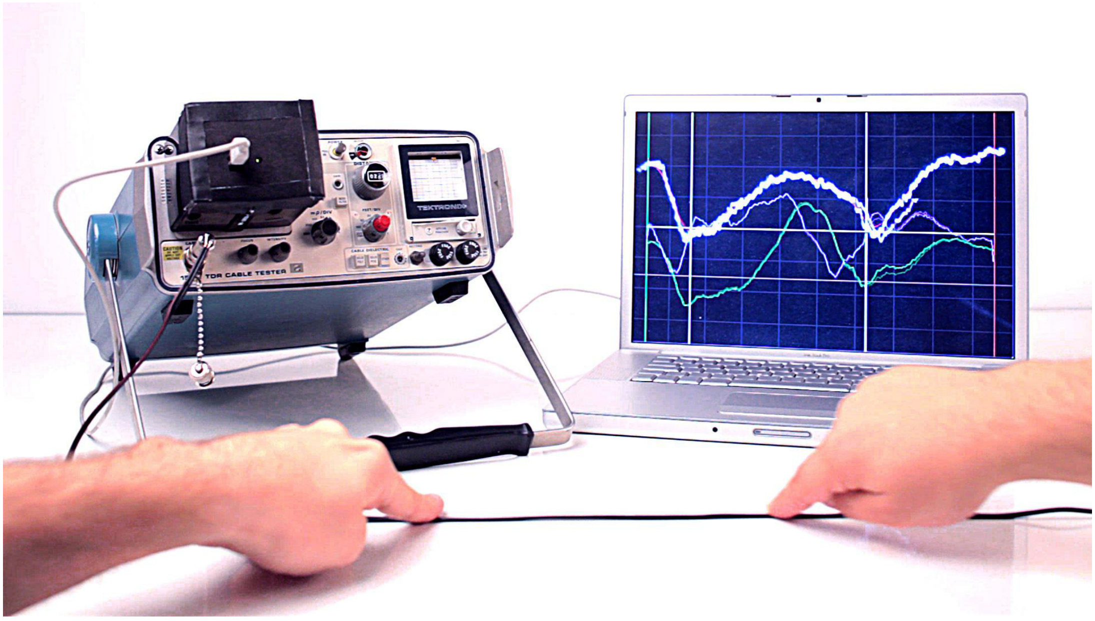

**Figure 7.17:** The setup used for our investigations: A Tektronix 1502 time domain reflectometer is used for scanning the cable. Its analog screen is digitized using a camera and analyzed live on a laptop.

digital siblings 1502B and 1502C, the 1502 uses a tunnel diode near its breakthrough voltage to produce a step signal with a rise time of 140 ps (Tektronix 1986, 1–1) and a voltage of 200 mV (Tektronix 1986, 3–4). As the diodes within the device are very sensitive to high voltages, the Tektronix 1502 can not be used for testing live wires, i.e. wires which concurrently carry signals or power[14]. The cable to be tested is attached to the device via a common BNC connector. The reflected signal is shown on a small analog CRT with an update rate of approx. 20 Hz. Dedicated knobs allow zooming into the trace and scrolling within it both in the horizontal and the vertical axis.

As a fully analog device, the Tektronix 1502 does not offer a dedicated PC interface. While it possible to retrofit an RS-232 interface to the device (Evett, Steven R. 2000), output speed is severely limited to one measurement every 20 seconds. To achieve real-time capture of TDR measurements, we mounted a Point Grey FireFly camera with a resolution of 640 $\times$ 480 px in front of the CRT so that the camera could capture the whole screen. To prevent reflections on the screen, a black, light-tight enclosure shielded camera and screen from ambient light. The camera's capturing rate was set to 20 fps to match the update rate of the CRT. At the 1502's maximum zoom level (21 cm of cable per screen), the camera's horizontal resolution of 640 px allows for a sensing resolution of 0.3 mm/px. In theory, this allows locating the peak of a discontinuity with an accuracy of 0.3 mm. Zooming all the way out shows a cable length of 400 m on the screen.

[14] There seems to exist a rare adapter for coupling the Tektronix 1502 to live wires, however.

While our setup might seem overly complicated, it offers significant benefits over using a semi-digital Tektronix 1502B or 1502C. First, the tunnel diode used in the 1502 achieves a shorter rise time than the digital equivalents used in 1502B and 1502C (140 ps vs. 200 ps), resulting in higher resolution. The 1502's analog CRT offers a higher update rate than the LCD used in the other models (20 Hz vs. 10 Hz). Finally, the LCD used in the 1502B/C models only displays 251 samples at a time. Therefore, capturing 640 samples per measurement from the 1502's CRT offers at least the same spatial resolution.

### 7.5.2 Software

The camera image is captured and processed using a Python script. This script uses *OpenCV* for capturing and image analysis, *numpy* for signal analysis, and *PyGame*, an *SDL* wrapper for various demo applications. The initial version was written by Markus Zimmerman during his project thesis. I further refactored and extended the script and implemented a plugin architecture for demos and measurement tools. The script is available on GitHub as open source under a BSD license[15].

The processing pipeline is implemented as follows (see Figure 7.18):

1) A frame is captured from the camera or loaded from an image sequence specified on the command line.
2) Optionally, slight lens distortion can be eliminated by a warp filter. As this filter is computationally complex, it introduces a latency of several frames. Therefore, this filter is usually disabled.
3) The *signal trace* (red) is extracted from the camera image by locating the brightest pixel in each pixel column. It is stored as a 640-sample numpy array.
4) Bends in the cable and varying distance between wires lead to static changes in the characteristic impedance along the cable. To extract only dynamic discontinuities caused by touches, a reference "untouched" trace is subtracted from the signal trace, resulting in a *calibrated trace* (green).
5) An adjustable low-pass filter is applied on each calibrated trace to remove noise in the spatial domain. Applying a low-pass filter across all samples of a single trace offers two benefits over purely temporal noise filtering. First, it effectively removes all small local minima, leaving only the wider negative peaks caused by touches. Second, spatial noise filtering allows for significant noise reduction without introducing additional latency.
6) The preprocessed trace is stored in a buffer together with the previous *n* traces.
7) An adaptive moving average filter is applied on each corresponding sample of all traces. It reduces noise in the time domain and stabilizes the location of negative peaks, eliminating jitter. The *adaptive* filter smoothes slow changes in the traces

[15] `https://github.com/raphaelwimmer/tdran`

more aggressively than fast changes. This allows the filter to react instantly to new touches. At the same time, the precision of the measurement quickly increases over time (Wimmer 2011c).

8) Touches show up in the filtered trace as negative peaks. The horizontal position of the peak indicates the touch location whereas the peak's amplitude indicates the force or proximity of the touch. Peaks are detected by calculating the first derivative of the graph (blue) and finding all zero-crossings. To avoid erroneous classification of noise as touches, only peaks which exceed a specific threshold are accepted as touch points.

All filter parameters need to be individually adjusted depending on cable choice and application.

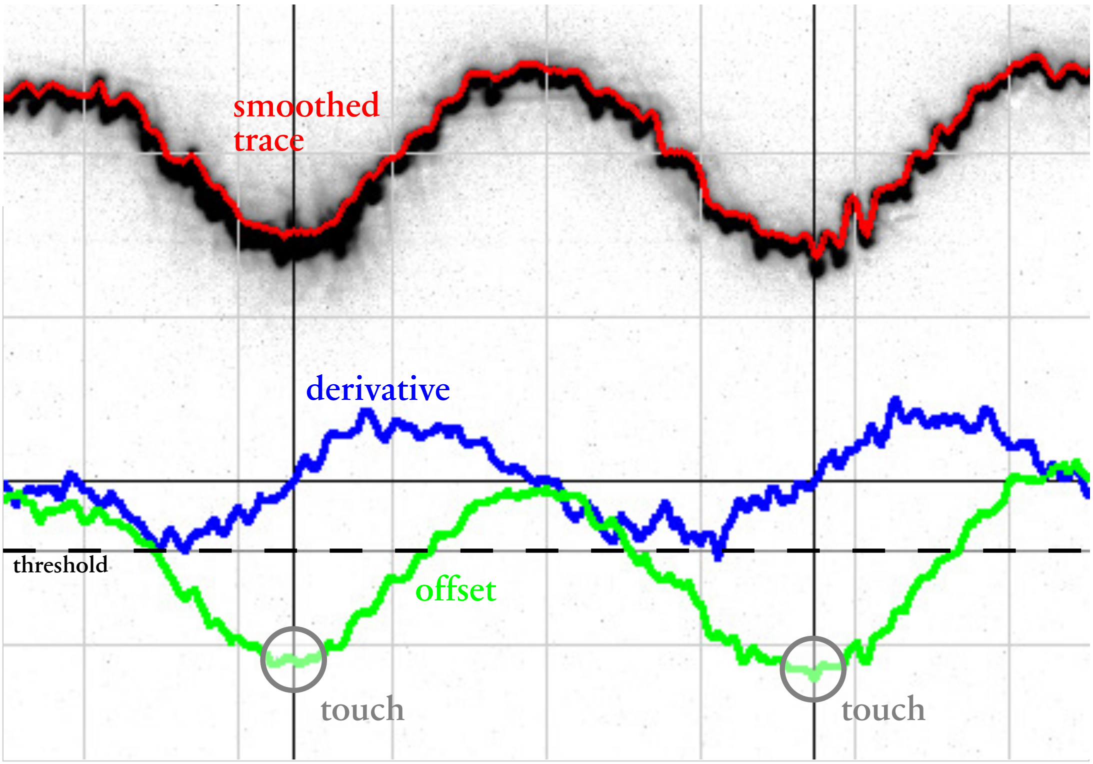


**Figure 7.18:** Algorithmic detection of two simultaneous touches: From the raw TDR trace (black line), a smoothed polyline (red) is extraced. From this smoothed trace, a reference trace is subtracted (not shown) which had been captured while the cable was not being touched. Touches are assumed to be at the local minima of the resulting offset trace (green). These are all locations in the offset trace where the dents go below a certain threshold (dashed line) and the first derivative of the offset trace (blue) is zero. Black and white have been inverted in this screenshot.

## 7.6 Performance Measurements

In order to determine capabilities and limitations of TDR-based touch sensing, I planned and conducted several measurements. As documented in Chapter 4, quantitative performance metrics for novel touch sensors are rarely reported. Therefore, it is not straightforward to directly compare TDR to other sensing techniques. In datasheets, usually a sensor's signal-to-noise ratio (SNR), positional accuracy/precision, and update rate are reported. However, these properties often depend on specific electrode choice and layout, environmental conditions, and other parameters.

The following performance metrics were selected in order to give an honest and useful overview of theoretical and practical capabilities: signal-to-noise ratio, precision of detected touch position, resolution of adjacent touches, range, and latency

For most of these, both raw properties and performance of our signal processing chain were determined. This allows for comparing TDR-based touch sensing to other sensing techniques while also giving an indication of the performance to be expected when using our setup in practice.

General effects of cable layout are described in Chapter 7.3.2. To inform the choice of cable for different applications, most measurements were conducted with five different cables with coplanar wires[16] (Figure 7.19):

a) flexible flat cable (FFC), 0.8 mm wide, 0.5 mm gap, 0.5 m long
b) ribbon cable, ∅0.4 mm, 0.9 mm gap, 2.5 m long
c) loudspeaker cable, ∅1.5 mm, 1.2 mm gap, 11 m long
d) copper foil traces, 8 mm wide, 10 mm gap, 0.5 m long
e) copper foil traces, 20 mm wide, 20 mm gap, 0.5 m long

### 7.6.1 Measuring Setup

Except for loudspeaker cable *"c"* - which already had a BNC connector soldered on - all cables were connected to the reflectometer via a 95 cm long cable with alligator clips. Cables under test were laid out in a straight line of approximately 50 cm and taped to a wooden board.

A *standard touch* was defined as a light touch with the pad of an index finger, exerting a force of 1 Newton. To ensure consistent force across trials, standard touches were simulated by placing small metal weights on the cables. These were chosen so that their effect on the TDR trace was equivalent to the change effected by a standard touch. For

[16] Note: the order of this list is slightly different from the one presented in our UIST paper

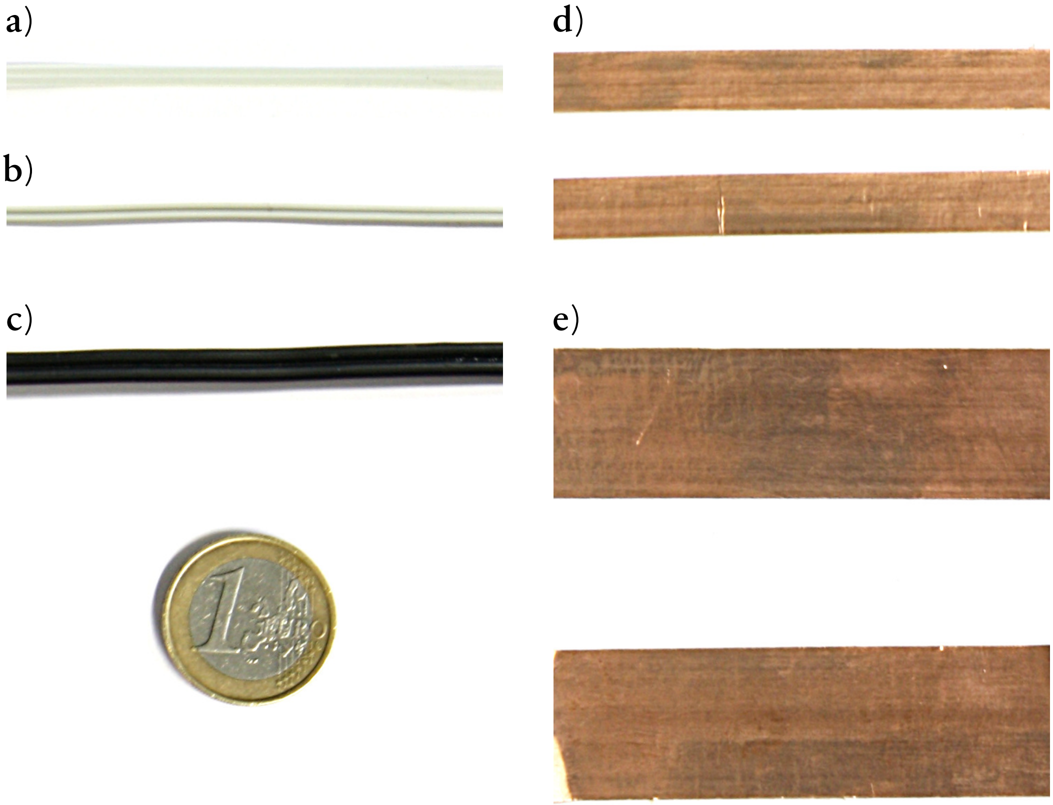


**Figure 7.19:** Overview of different cables evaluated for TDR touch sensing: a) Flexprint ribbon cable, b) ribbon cable, c) loudspeaker cable, d/e) copper foil traces.

the flat cables (a,d,e) a small ferrite block (28 mm x 14 mm x 8 mm, weight 10 g) was found to be ideal. For measuring the bare copper foil traces, a sheet of paper was put between weight and copper foil to avoid creating a short between the traces. When touching thicker cables, the finger's tissue partially encloses them, increasing capacitive coupling. As rigid weights do not wrap around the cable, a different weight - a small steel plate (22mm x 26mm x 3mm, weight 10g) - had to be chosen for for cables *b* and *c* in order to simulate the same standard touch. Unless otherwise noted, the weight was placed on the cable to be tested at a distance of 250 mm from its start, respectively 120 cm from the reflectometer's output.

The Tektronix 1502 displays absolute changes in characteristic impedance in units of $m\rho$ (millirho). The symbol $\rho$ denotes the percentage of reflected signal, $U_{reflected}/U_{pulse}$. Therefore, 1 $m\rho$ equals a reflection of 1/1000 of the pulse's amplitude. As the Tektronix 1502 emits 200 mV pulses, 1 $m\rho$ equals 0.2 mV.

For all measurements, our *tdran* software was used together with a custom Python plugin that calculated absolute changes in amplitude ($m\rho$) respectively location (mm) from changes in the captured TDR trace. Appropriate conversion factors were determined experimentally. The software also automatically logged and calculated all statistics presented in this chapter.

Obviously, this measurement setup is limited by the capabilities of the Tektronix 1502. However, informal experiments with a Mohr CT100HF indicate little difference in practical performance between these devices. Therefore, our measurement results should indicate the typical performance to be expected with current hardware in the price range of up to EUR 10,000.

### 7.6.2 Signal to Noise Ratio

Background noise shows up in the TDR signal traces in two ways:

- *spatial noise*, i.e. aberrations from the 'real' signal within a single signal trace.
- *temporal noise*, i.e. aberrations across subsequent signal traces

Mostly, this is probabilistic noise caused by stray capacitance and interfering electric fields. However, spatial and temporal noise are not necessarily correlated all the time. For example, radio interference might introduce significant noise in the time domain. However, it affects the whole cable at the same time, therefore introducing little spatial noise. On the other hand, slightly uneven cable properties cause differences in velocity of propagation and characteristic impedance along the cable. These changes effect a temporally consistent distortion of the trace. Depending on the electrical characteristics of different cable segments, these also might show different amounts of temporal noise. Finally, as the Tektronix 1502 employs random equivalent sampling, temporal noise also shows up as spatial noise.

As it is not feasible to accurately model the noise properties of all different cable setups within this thesis, a simpler approach is used: In order to quantify noise levels and signal-to-noise ratio (SNR)[17], I calculate the standard deviation of the signal in a no-touch state from the baseline[18]. To capture both spatial and temporal noise characteristics, first the standard deviation (from the baseline) of each single trace (400 samples) is calculated. Then, the maximum of the standard deviations of 200 traces (~10 s) is taken. This approach has certain weaknesses. For example, the standard deviation of the noise floor was only determined for the no-touch state. In theory, noise might become much stronger when the cable is touched. However, this was never observed in practice.

[17] For an in-depth discussion of signal-to-noise ratio and other performance metrics, see Atmel's Touch Sensor Design Guide (Atmel Corporation 2000).

[18] In the UIST paper, the term *RMS* (root mean square) was incorrectly used instead of *SD*. RMS and SD are effectively the same, therefore the values reported in the paper are still correct and indicative of the performance. However, RMS describes deviation from the *actual* position, whereas SD describes the deviation from the *mean* of all samples (Deakin and Kildea 1999). As only the latter could be determined, SD is the correct term to use.

However, I believe that this approach gives a reasonable upper limit for the amplitude of the noise floor. While the approach might over- or underestimate actual noise levels, I do not expect such errors to have a significant effect on the overall characteristics of this sensing technique. More importantly, these errors would be independent of cable choice. Therefore, the measured values should be suitable for comparing different cable types.

To reflect both actual usage and "plain" performance, all measurements were conducted two times - once with a 32-sample AMA filter, and once without. The results, documented in Table 7.1 (columns 1-3), show that the signal-to-noise ratio for a single touch is always over 20 dB (12 dB without filtering), which is easily sufficient for detecting touches on the cable, even under difficult circumstances. Raw SNR (i.e., without filtering enabled) for cables d and e is comparable to that of standard capacitive touchscreen controllers (25:1 - 80:1, (Barrett and Omote 2010)). Due to their plastics coating, cables a-c are significantly less sensitive.

| | Amplitude ($m\rho$) | Noise SD ($m\rho$) | SNR (dB) | Pos. Err. SD (mm) | Min. Touch Dist. (mm) |
|---|---|---|---|---|---|
| a | 30 | 2 (5) | 23 (15) | 5 | 150 |
| b | 40 | 3 (7) | 22 (15) | 1 | 40 |
| c | 30 | 3 (7) | 20 (12) | 4 | 20 |
| d | 220 | 3 (9) | 37 (27) | 1 | 25 |
| e | 150 | 3 (7) | 34 (26) | 1 | 25 |

Table 7.1: Sensing resolution for different cable types when using an adaptive moving average filter (32 samples). Raw values (without filtering applied) in parentheses. From left to right: Amplitude of change caused by a touch, standard deviation of noise floor, signal-to-noise ratio for a touch, standard deviation of calculated touch position for a static touch, minimum distance at which two adjacent touches can be distinguished. Electrical properties of cables a – e are described above. Source: Wimmer and Baudisch (2011).

### 7.6.3 Precision and Positional Jitter

The amount of background noise not only determines the signal-to-noise ratio but also introduces jitter to the positions of local minima caused by touches on the cable. Smoothing each TDR trace together with the adaptive moving average filter described in Section

7.5.2 strongly reduces such jitter. Typical jitter was measured by placing the standard metal weight on each cable and recording 800 frames (40 seconds) with full filtering enabled. For each cable, the standard deviation of the recorded touch position was calculated. As shown in Table 7.1 (column 4), the ribbon cable and both copper foil traces allow for a precision of 1 mm (standard deviation). Therefore, our TDR implementation - including temporal and spatial filtering - seems to offer sufficient precision for most touch-sensing applications. In comparison, precision was rather low for FFC (a, SD: 4 mm) and loudspeaker cable (c, SD: 5 mm). Nevertheless, this precision is probably still acceptable for many applications.

Absolute accuracy of the measured position was not evaluated due to time constraints.

### 7.6.4 Minimum Distance Between Individual Touches

For multi-touch interaction and grasp sensing, it is necessary to reliably distinguish two or more touches that are close together. Therefore, for each cable I determined the smallest distance between two concurrent *standard touches* at which our implementation is still able to clearly distinguish them. This was done by placing two standard weights on the cable and decreasing their distance (measured center to center) until our algorithm started reporting a single touch instead of two. This happens when the two peaks get so close to each other that the dent between them gets erased by background noise. The results shown in Table 7.1 (column 5) indicate that our setup offers an acceptable minimum distance between touches for most cable types. While the FFC (a) requires a surprisingly high minimum distance of 150 mm, the other cables that were tested allow for minimum distances of 20-40 mm. As a human finger is 10-20 mm wide, cables b-e allow distinguishing two fingers that are placed right next to each other - which is the absolute minimum distance that two touches can be apart. Therefore, these cable types are suitable for grasp sensing.

In addition to the cable type, the rise time of the pulse has a major influence on achievable minimum distance. The 1502's rise time of 140 ps results in a rising edge of about 20 mm length. This means that the peaks of touches start to merge once their distance get lower than 40 mm. At a distance of 20 mm, SNR is effectively halved. A shorter rise time, i.e. steeper edges of the peaks, would allow for even smaller distances between to touches. Touching the cable with a greater force than the 1 N we used for testing, or holding the cable between two fingers, also generates steeper edges, resulting in an even lower minimum distance between touches. Therefore, the reported metrics should be seen as lower bounds that can certainly be improved.

### 7.6.5 Sensing Range Along the Cable

Sensing range of TDR is determined by cable choice and power of the pulse. Due to the cable's impedance, any signal injected into the cable gets weaker while it travels along the cable. As only a tiny fraction of the original pulse gets reflected back at discontinuities, these echoes start to blend with background noise with increasing distance. The better insulated the electric field between both wires, the less power loss occurs along the line. On the other hand, the more sensitive to touches a cable is, the worse is the achievable sensing range. Therefore, coaxial cable offers a larger sensing range (but less sensitivity to near and far touches) than coplanar wave guides or twisted-pair cables. This fundamental relationship unfortunately severely limits the size of touch-sensitive surfaces built with TDRtouch.

Practical sensing range was measured exemplarily for the loudspeaker cable (cable c) by placing the standard weight on the cable at multiples of 50 cm. The amplitude of the local minimum decreased in a linear fashion with distance. At a distance of about 6 m, the peak blended with the ambient background noise, making touch recognition impossible. By firmly grasping the cable in a fist, a much larger peak can be caused. This peak could be reliably detected at a distance of up to 20 m.

In general, sensing range can be improved by increasing the voltage of the pulse. However, an increase in voltage also increases the rise time of the pulse, reducing spatial resolution. As it is not possible to change the output voltage of the Tektronix 1502 (200 mV), the relationship between pulse voltage and sensing range could not be evaluated.

### 7.6.6 Latency

For interactive applications, low input latency is important. Capturing a single TDR trace is fast - a length of 50 m can be scanned in less than 1 $\mu$s.

However, sampling and post-processing introduce additional delay. The more filtering gets applied to the signal, the better the SNR, but also the higher the latency. I measured the latency of our implementation using the same filter settings that were used in determining SNR and precision. Cable d was used for all latency measurements.

We experimentally determined latency for tapping and dragging. Tapping latency is the amount of time between a finger starting to touch the cable and a our software detecting a touch. Dragging latency is the amount of time between the finger stopping at a defined point after a dragging motion and our software reporting a touch at exactly that point. We further distinguished between slow (1 Hz) and fast tapping (2 Hz), respectively slow (125 mm/s) and fast (500 mm/s) dragging in order to gain a broader picture of the latency. These values were chosen arbitrarily. Each of the four conditions was conducted manually 20 times in a row, and the average for each condition was calculated. In four cases, our implementation did not register a touch at all. These cases were not included in the calculations. In order to achieve constant manual tapping and dragging speeds, a click track was played back and the experimenter (me) synchronized his actions to the audible clicks spaced at 0.5 and 1 second intervals.

I measured the latency of our implementation by simultaneously capturing both cable and visualization with a video camera at 25 fps. Afterwards, the video was analyzed frame by frame to determine how long it took from the physical action (tapping/dragging) to the output on the computer screen. This low-cost approach offers a temporal resolution of 40 ms per frame. In the worst case, the length of the measured time span is off by 80 ms. In the average case, the measured length is off by 40 ms. By averaging the results of all 20 runs per condition we achieved an effective time resolution of less than 10 ms.

Overall latency consists of four parts:

- constant delay between a touch and it showing on the CRT of the Tektronix 1502: 25 ms on average at a 20 Hz update rate
- constant delay of our capturing and filtering pipeline: (about 90 ms, determined experimentally as described above)
- variable delay of the AMA filter
- screen lag: 8 ms on average at a 60 Hz update rate (estimated)

For slow and fast tapping, the AMA filter does not add any delay. Therefore, overall latency is 120 ms which is comparable to standard touchscreens (80 ms) and seems to be acceptable for many applications (Anderson, Doherty, and Ganapathy 2011). However, Jota et al. (2013) show that an input latency of more than 20 ms starts affecting users - and an even lower latency of 2 ms should be targeted for dragging interaction.

As the AMA filter was optimized for tapping, i.e. rapid changes in amplitude, it performs poorly in the slow and fast dragging tasks For slow dragging, overall latency is about 800 ms; for fast dragging, latency is about 400 ms. This is not acceptable for most applications. Therefore, only light filtering, no filtering, or another low-latency filter should be used for dragging.

As most of the latency is introduced by our prototypical software implementation, it should be possible to achieve a latency of better than 20 ms by increasing the update rate and implementing peak detection and filtering in hardware.

### 7.6.7 Discussion

While TDR is an established sensing technique for fault detection in cables, its utility for touch sensing had not been explored before. The properties documented in this section allow researchers and practitioners to evaluate whether TDRtouch is the right tool for their touch-sensing needs. TDRtouch allows for reliably detecting the touch locations of a large number of simultaneous touches with high spatial resolution and a high update rate. Our specific setup offers a signal-to-noise ratio comparable to standard touchscreens, spatial resolution in the millimeter range, a sensing range up to 20 m, an update rate of 20 Hz and latency of 120 ms. All of these properties can still be significantly improved by using state-of-the-art hardware and optimizing filtering algorithms.

Major weaknesses are the inherent range/sensitivity and range/resolution trade-offs which make it hard to implement very large touch-sensitive surfaces.

We discovered a further limitation that has not been described in related work but is very relevant for our use case: TDR is susceptible to radio interference from nearby mobile phones. Three factors contribute to this susceptibility:

- The long cable of a TDR setup acts as an antenna, picking up radio signals,
- TDR employs pulses in the GHz range - which is also the frequency range used by GSM and 3G mobile networks.
- TDR devices measure voltages in the mV range and therefore easily pick up the high-power transmissions of mobile phones.

In practice, this means that the signal trace gets completely garbled each time a mobile phone within a range of less than one meter transmits a packet. For mobile applications, radio interference would need to be mitigated by appropriate digital filtering or by synchronizing touch sensing with pauses in radio communication.

## 7.7 Discussion and Impact

Overall, TDRtouch is an extremely versatile prototyping technique for grasp-sensitive surfaces and may also be used for interactive installations. Using only a flexible two-wire cable allows for building small and large, non-planar and deformable grasp-sensitive surfaces.

TDRtouch fulfills all three requirements for a grasp-sensitive surface, as outlined in Section 4.4.2.

Additional, unique features - such as modularity and automatic identification of attached sensors - allow for rapid prototyping.

However, a number of limitations have already been mentioned:

- TDR in general is susceptible to radio interference.
- Similar to other capacitive sensing techniques, TDRtouch does not work well near conductive or metallic objects; this limits the technique to augmenting only non-conductive surfaces.

- Resolution orthogonal to the cable direction is significantly lower than along the cable.
- For completely covering two-dimensional surfaces, great lengths of cable are required; the power loss along the cable results in decreased sensitivity at large distances
- Our current implementation is still quite bulky and expensive

While the first two limitations can not easily be mitigated, except possibly through elaborate active shielding, application-specific trace layouts can counter the third limitation. Approaches for solving the final two limitation are discussed in the next section.

Like FlyEye, TDRtouch has not been evaluated in actual grasp recognition tasks beside simple informal tests. However, the measurements described in this chapter, as well as experiments that resulted in the design guidelines I presented, indicate that TDRtouch offers a resolution comparable to other grasp-sensitive surfaces while allowing for extremely easy and quick prototyping.

Our research on TDRtouch started in 2010 and was published first in 2011. Since then our paper has been cited about a dozen times - generally as "somehow-related" work in papers proposing novel sensing techniques. Olberding et al. (2013) explicitly mention that TDRtouch inspired their choice of wire layout[19]. To my knowledge, no scientific research or commercial products have been published so far that supersede, expand or contradict our work. While external impact so far has been low, we have continued our research on this topic. In the following section, I suggest areas for future work.

## 7.8 Future Work

Our low-cost TDRtouch setup works surprisingly well and allowed for evaluating key properties of TDR-based touch sensing. However, it is not yet suitable for non-technical users and commercial applications. The obvious next step is to create a commercial product or open-source design that addresses the current limitations, mentioned in the previous section.

I see the following tasks:

### 7.8.1 Reducing Price

The most important limitation of TDRtouch is its current price point. Old Tektronix 1502 reflectometers are rare and usually still cost more than 1,000 EUR. New time domain

[19] However, Olberding et al. incorrectly describe TDRtouch as only being able to sense a single touch.

reflectometers - such as the MOHR CF100 series - cost upwards of 10,000 EUR. For TDR to become accepted as a prototyping tool, a reflectometer should probably cost not more than 200 EUR. Once the price falls below 50 EUR, TDR might also get used in low-quantity commercial products and one-off installations. If we manage to get the price of a reflectometer circuit below 10 EUR, TDR-based touch sensing might be regularly used in commercial products.

The first price point - 200 EUR - can probably be achieved by implementing pulse generator and sampling unit using off-the-shelf parts and custom-programmed microcontrollers or FPGAs.

For reaching the lower price points, specialized ICs would need to be manufactured which only seems feasible in the long run.

To this end, we have investigated a few existing TDR designs and potential components (Bartling 2009; Skierucha et al. 2012; Starecki and Misiaszek 2006; acam messelectronic GmbH 2000) and considered other approaches, such as using time-of-flight cameras or laser distance meters for high-resolution time measurements.

### 7.8.2 Reducing Size

By employing ICs and moving the user interface into the host computer, it should be easily possible to shrink the reflectometer to about the size of a pack of cards. Integrating all necessary electronic components into a specialized IC - a long-term goal - would allow for embedding the reflectometer directly into the grasp-sensitive artifact.

### 7.8.3 Increasing Range, Resolution, and Sensitivity

While TDRtouch offers acceptable range, resolution, and sensitivity for many prototyping applications, extending the sensing range to several kilometers and increasing resolution to less than a millimeter would open up many new applications. Furthermore, increasing sensitivity would allow for using smaller cables, which in turn would allow for a higher sensor resolution orthogonal to the cable direction.

To do this, pulse voltage could be raised, rise time reduced, and sampling depth increased. Improvements in digital filtering may improve resolution and sensitivity further. However, it seems plausible that initially spatial resolution will be significantly lower than that of our TDRtouch setup.

### 7.8.4 Expanding Applications

Besides improving performance and availability of the reflectometer, there are several other research directions that warrant further investigation.

For example, it would be worthwhile to systematically investigate the properties of coaxial cable with regard to TDR-based touch sensing. While coaxial cable does not allow for sensing light touches or proximity, locations of punctual pressure on the cable can be determined via TDR. Coaxial cable offers significantly lower power loss along the cable than coplanar waveguides or parallel wires. Therefore, suitably deformable coaxial cable might allow for building very long touch sensors using TDR.

In our experiments, we only connected a single, continuous cable to the reflectometer. However, TDR can also detect discontinuities in cables with multiple branches. While the measured distance to the discontinuity does not give an indication on which branch the discontinuity is located, multiple sensing passes from different ends of the cable can be used to exactly pinpoint discontinuities. Multi-branch TDR would allow for much more complex cable layouts, potentially enabling interesting new applications.

Optical Time Domain Reflectometry (OTDR) is very similar to standard (metallic) TDR but uses optical fiber as the medium. Getting OTDR-based touch sensing to work would allow for true single-fiber operation and make sensing immune to nearby conductive objects and radio interference. While OTDR has been mentioned in one touch screen patent (Kim 2011), the feasibility of this approach has not been demonstrated. Therefore, it is not yet clear how well OTDR-based touch sensing works in practice.

### 7.8.5 Easing Prototyping

Finally, there is still a lot of room for improvement of the prototyping tools and workflows. For example, mapping a touch location in the signal trace to a point on the grasp-sensitive surface requires tedious manual calibration. Relative mapping, as described in the FlyEye chapter might significantly reduce calibration effort. Besides better software support, a set of flexible, self-adhesive cables optimized for touch sensing might accelerate prototyping further. To this end, proper trade-offs between sensing resolution and flexibility need to be made.

At the time of publication of this dissertation, we are still looking into all of these topics but have not yet published further results.

Chapter 8

# HandSense, FlyEye, TDRtouch - Comparison and Discussion

*In this chapter I summarize the properties of the three prototyping techniques proposed in the previous chapters - HandSense, FlyEye, and TDRtouch - and compare them to each other and to approaches from related work. Criteria for comparison are quantitative and qualitative properties with regard to grasp sensing as well as suitability for rapid prototyping. HandSense is inexpensive and versatile but offers only low spatial resolution. FlyEye is easy to use and offers high spatial resolution but is less suited for iterative refinement of prototypes. TDRtouch offers a high resolution and is very versatile. However, it is currently significantly more expensive than the other two techniques. If all properties are given equal weight, TDRtouch is ranked as preferable to HandSense and FlyEye. In practice, application-specific constraints will result in different weightings. The detailed comparison in this chapter allows for choosing the most suitable technique for a given application. Comparing the three proposed techniques to approaches from related work - prototyping techniques and prototypes - indicates that in many cases all three approaches offer sufficient performance and suggests areas for improvement.*

**Attribution:** This is an original chapter containing only new material written for this thesis.

## 8.1 Introduction

In the previous three chapters I described three techniques for prototyping grasp-sensitive surfaces.

HandSense offers a small number of highly sensitive capacitive sensors connected to a controller board. Due to their size, their position needs to be carefully chosen in order to reliably distinguish between different grasps.

FlyEye employs dozens or hundreds of optical fibers which are connected to a camera on one end and embedded into the surface of an object on the other end. This allows for high-resolution proximity sensing and touch sensing. However, routing the optical fiber bundles within the graspable object requires lots of space.

TDRtouch requires only a single cable that can be wrapped about arbitrarily shaped objects or embedded into their surfaces. Due to its size, the reflectometer used in our reference implementation needs to be placed outside the grasp-sensitive object. At the time of writing, no cheap and/or small reflectometer is commercially available.

Each associated research project focused on a different aspect:

With HandSense, the main goal was to make capacitive sensing usable for prototyping, and to explore how suitable a small number of high-sensitivity sensors are for grasp sensing. The focus was primarily on grasp classification, whereas capturing and preprocessing steps were quite straightforward.

FlyEye showed further ways to reduce barriers to prototyping. It allows designers without soldering experience to build grasp-sensitive prototypes and installations. The relative mapping algorithm developed for FlyEye allows for semi-automatically ordering unordered spatial sensor data. For FlyEye, the focus was on sensing hardware and preprocessing of the captured raw data.

TDRtouch simplified prototyping grasp-sensitive surfaces even further. Arbitrary objects may be made grasp-sensitive in a non-destructive way, just by wrapping a cable around them or covering them with a mesh. TDRtouch focused even more on the sensing hardware, and to a smaller part on preprocessing of sensor data. Grasp classification is only mentioned in passing.

Over the course of these projects, the focus moved from classification and preprocessing towards the design of the sensor hardware. The primary reason for this is that grasp classification has received a lot of attention already, as described in Chapter 4, whereas no other suitable prototyping techniques for grasp-sensitive surfaces exist.

This chapter focuses on a comparative evaluation of these three prototyping techniques.

- In Section 8.2, the three techniques are compared with regard to important characteristic properties.
- Section 8.3 extends the comparison to other existing grasp-sensitive surfaces and prototyping tools for touch sensing.

# 8.2 Comparison: HandSense, FlyEye, TDRtouch

In the following, HandSense, FlyEye, and TDRtouch are compared to each other with regard to the following criteria. These have been chosen to give a realistic and useful overview of the differences between the techniques:

**Quantitative sensor properties:**

- spatial resolution
- sensor bit depth
- latency

**Qualitative sensor properties:**

- support for non-planar surfaces
- deformability
- quality of sensor data

**Suitability for prototyping:**

- ease of use
- cost
- incremental refinement

## 8.2.1 Spatial Resolution

As discussed throughout the previous chapters, spatial resolution is a very important property of a grasp-sensitive surface. Spatial resolution is comprised of two factors - sample spacing and sampling aperture (Smith and others 1997, chap. 25). A common measure for the sampling aperture is the width of the 10%-90% transition of the edge response.

As the research presented in this dissertation was conducted over the course of several years and accompanied by a learning process, no formal measurements have been conducted with each technique that would allow for directly comparing them. Furthermore, HandSense prototype and FlyEye prototypes were salvaged for components for other projects. Therefore, it is not easily possible to conduct new measurements under controlled conditions.

In addition, environmental conditions and the intrinsic properties of the object that is to be made grasp-sensitive have a major effect on practically achievable resolution.

Nevertheless, practical resolution of HandSense, FlyEye, and TDRtouch can be roughly estimated for 'standard' conditions, i.e. without major influence of nearby conductive objects (HandSense, TDRtouch) or extreme lighting conditions (FlyEye):

**HandSense** offers a sample spacing of 20 mm - the diameter of a single CapSensor. Sampling aperture depends on several factors but is approximately 50 mm.

**FlyEye** offers a sample spacing of 4 mm - the spacing between sensing fibers on the ping-pong ball prototype. Sampling aperture is estimated at 1 mm, i.e. only objects more or less touching the sensing fiber cause light to enter it.

**TDRtouch** offers a sample spacing of less than 1 mm along the wire. Sample spacing perpendicular to the wire is equivalent to the distance between serpentines or windings of the cable. This is about 5 mm - 20 mm, depending on cable type. Sampling aperture is about 20 mm for a typical cable.

Overall, resolution of FlyEye is significantly higher than that of TDRtouch, which in turn has a significantly higher resolution than HandSense. FlyEye is capable of resolving two directly adjacent fingers. TDRtouch can only resolve adjacent fingers if they are next to each other on the cable, not if they are on different serpentine segments. HandSense is not able to resolve adjacent fingers, except if their placement is restricted by grasp affordances (see Chapter 11.)

### 8.2.2 Sensor Bit Depth

Bit depth describes the available resolution to describe the value of a single sample. It is related to SNR and other measures of signal quality. A sensor with which is only able to distinguish between two states (e.g., touched or not) has a bit depth of 1 bit. In practice, higher bit depth allows for detecting proximity, measuring touch force, or also distinguishing between different parts of the body touching the sensor.

**HandSense** uses a CapBoard with a raw resolution of 16 bit. In practice, a value range of about 80% (15 bit) was used for classifying grasps.

**FlyEye** uses a camera with a bit depth of 8 bit per pixel. As the camera was calibrated to cover the complete brightness range between non-touched and touched state, the effective bit depth is at least 7 bit.

**TDRtouch** has a theoretical maximum bit depth of around 8 bit ($log_2(480 px vertical image size)$). In practice, the effective SNR of 12-37 dB (see Section 7.6) means a bit depth of 2-6 bit.

Overall, these values and practical experience suggest that all three techniques offer sufficient bit depth to distinguish between proximity, light touch, and strong touch.

### 8.2.3 Latency

Faithfully capturing static grasps does not require a high sensor update rate. However, the system's latency - the time between a user grasping the object and it reacting to the grasp - should be low. While the maximum acceptable system response time depends on the application (Miller 1968), a system response time of less than 100 ms is seen as not noticeable (Anderson, Doherty, and Ganapathy 2011), However, for some tasks overall latency needs to be lower than 10 ms (Ng et al. 2012; Jota et al. 2013) in order to not be noticeable at all. System reaction time is comprised of times for capturing, preprocessing and classifying grasp information. Usually, preprocessing and classification take more time than data acquisition.

**HandSense** offers a raw update rate of 50 Hz for each channel. As only two of the eight channels are active at a time, actual update rate for a complete sensor scan is 12.5 Hz, resulting in a latency of 80 ms. In practice, waiting for three consecutive identical SVM classifications resulted in a system response time of about 600-700 ms.

**FlyEye** has an update rate of 30 Hz. No further latency measurements have been conducted. As the time-consuming calibration step had to be conducted only once per setup, actual system response times should be similar or shorter than those measured for HandSense.

**TDRtouch** has a maximum update rate of 25 Hz, limited by the update rate of the cathode ray tube. In practice, the latency for data acquisition and preprocessing - but without classification - is about 120 ms.

Overall, all three techniques offer latencies of 40 ms or lower for data acquisition, which describes the lower bound for overall system response time. In practice, system response time is limited by the speed of preprocessing and classification, which vary with application and processing power. The results for HandSense suggest a system response time of less than one second for all three techniques. This seems acceptable for basic grasp sensing tasks. In general, a latency of 100 ms would be preferable. Latencies of about 10 ms - which are desirable for interactive manipulation - are not currently possible with any of the presented techniques.

### 8.2.4 Support for Arbitrarily Shaped Surfaces

Rarely does a graspable object have a box shape. For ergonomic and aesthetic reasons, many graspable objects - especially tool handles - have curved surfaces. Therefore, a grasp sensing technique needs to support non-planar surfaces.

**HandSense** allows for attaching non-planar sensor electrodes that mimic the curvature of the object. However, due to the sensors' size, the minimal diameter of any part of the object is about two centimeters.

**FlyEye** allows for covering arbitrarily shaped surfaces. However, the space requirements for routing the optical fibers within the object makes it less suitable for very small objects.

**TDRtouch** also allows for covering any shape, even very small ones. However, some shapes may require careful routing of the cable in order to cover the complete surface without overlapping cable segments.

Overall, all three techniques allow for augmenting non-planar surfaces with grasp sensing. FlyEye is the most versatile approach. TDRtouch is especially suited for very small objects which do not allow for electronics or fiber bundles inside.

### 8.2.5 Deformability

None of the grasp-sensitive artifacts presented in the related work section are deformable in a major way. Therefore, deformability might be of minor importance for grasp sensors. However, deformable sensors are helpful when prototyping, allowing the designer to easily fit sensors to arbitrarily shaped surfaces. Furthermore, objects with deformable surfaces allow for giving tactile and haptic feedback, conform better to the user's hand, and may reduce strain on joints and muscles during interaction. Finally, many graspable objects - such as bags, paper sheets, clothing, or cables - *are* inherently deformable. Therefore, deformable sensors may facilitate novel user interfaces for existing and novel graspable objects.

**HandSense** employs hard electrodes with diameters of at least 10 mm. However, these only cover parts of the surface and need not be rigidly connected to each other.

**FlyEye** employs flexible optical fibers with a small diameter that are not connected to each other. Embedding these in a deformable material - such as silicone - makes a arbitrarily deformable surface. However, routing of the fibers imposes limits to deformability for small devices.

**TDRtouch** allows for using a variety of stretchable and deformable conductors. Thereby it enables large and small, arbitrarily deformable surfaces.

Overall, all three techniques support deformable surfaces. TDRtouch and FlyEye are preferable over HandSense as the latter requires rather large non-deformable electrodes.

### 8.2.6 Proximity Sensing

While grasping requires the user to actually touch the object, sensing proximity may enhance grasp interaction in three ways: First, capturing not only contact areas but also proximity of body parts provides additional data for reconstructing hand postures and classifying grasps. Second, proximity sensing allows the grasp-sensitive object to anticipate grasps, possibly reducing latency and power consumption in a non-grasped state. Third, proximity sensing may support mid-air gestural interaction near the object, augmenting its capabilities.

Both for capacitive sensors and sensors based on optical reflection, the power supplied to the sensor determines the achievable range. Therefore, only the actual capabilities of the implemented prototypes are described in the following.

**HandSense** allows for sensing proximity of a finger or hand several centimeters from the surface.

**FlyEye** offers only a sensing range of a few millimeters.

**TDRtouch** offers a sensing range of a few millimeters up to several centimeters, depending on the width of the cable. Typically, grasp-sensitive surfaces based on TDRtouch only support millimeter ranges.

Overall, all three techniques offer minimal proximity sensing. HandSense is also able to sense movement several centimeters away.

### 8.2.7 Ease of Use

The primary goal for HandSense, FlyEye, and TDRtouch was to make prototyping grasp-sensitive surfaces accessible to interaction designers with limited technical background. Furthermore, as a prototyping technique, each of three approaches should allow for quickly building simple and complex systems.

**HandSense** requires a limited amount of one-time soldering for assembling sensors and controller if not bought as a kit. Sensors can be easily added or removed during runtime. If custom electrode shapes are required, these need to be soldered to a pin header which acts as a detachable connector between sensor and electrode. Sensor data is straightforward to interpret and transform depending on the application.

**FlyEye** requires a minimal amount of soldering for the camera-triggered IR light source. Cutting, deforming, and bundling optical fibers requires little skill. However, this may become tedious for large surfaces. The relative calibration required for correctly mapping fiber ends may not be as easy to understand and debug as HandSense's simple sensor values.

**TDRtouch** may require a minimal amount of soldering for attaching cables to connectors. However, in many cases different cables can be attached to the reflectometer via alligator clips. Wrapping a cable around an object requires little planning and time. Sensor data can be visually inspected and debugged.

Overall, TDRtouch is the most intuitive and convenient solution for prototyping touch-sensitive surfaces. HandSense requires some soldering but is easy to set up and debug.

FlyEye requires only a minimal amount of soldering but may be tedious for larger surfaces. Its representation of sensor data is the least intuitive of the three techniques.

### 8.2.8 Cost

While low cost may not be the most important property of a prototyping tool, it certainly contributes to its adoption. Overall cost consists of one-time expenses and recurring expenses for building new prototypes.

**HandSense** requires a one-time investment in a controller board (ca. 50 EUR) that can be reused for different prototypes. While sensors (ca. 5 EUR each) may be detached from their electrodes in theory, in practice this may not be feasible due to space constraints in the prototype. Electrodes usually consist of low-cost sheet metal and do not significantly contribute to overall cost.

**FlyEye** requires a one-time investment of about 100 EUR for camera and IR source. One meter of optical fiber (1 mm) costs about 0.50 EUR[1]. A prototype with 100 fibers which are 20 cm long would cost about 10 EUR. Optical fibers may not easily be reused, especially if the surface was coated with glue or silicone.

**TDRtouch** requires a relatively high one-time investment for the reflectometer. A new reflectometer with sufficient resolution costs about 10,000 EUR. Recurring costs for cable, copper foil or conductive paint are minimal.

Overall, HandSense is the least expensive technique, closely followed by FlyEye. TDRtouch requires a very high initial investment. As described in Section 7.8, a custom TDR circuit design might drive down costs significantly in the future.

### 8.2.9 Incremental Refinement

Prototyping interactive systems is usually an incremental, iterative process of design, implementation, and evaluation. Therefore, a good prototyping tool should allow for incremental changes to be made. For grasp-sensitive surfaces, the designer might want to change the shape of the surface as well as layout and density of sensors.

**HandSense** supports incremental refinement well. Individual electrodes and sensors can be swapped easily. Electrode may be removably attached below the surface using glue or sticky tape. Cable length can easily be changed.

**FlyEye** requires drilling holes into the surface to be augmented with sensors. This severely limits incremental refinement of sensor layout. Sensor density may be increased by drilling additional holes.

[1] at conrad.de

**TDRtouch** allows for easy refinement of sensor layout and density. HandSense and FlyEye require a hollow object for installing sensors respectively fibers. With TDRtouch, cables can be attached to the inside of thin cases or on the surface of solid objects. However, usually the whole cable needs to be removed and re-attached for most of these changes. Depending on how the cable is affixed to the surface, this can be done very quickly (masking tape) or may require a lot of time (conductive paint). Cables can be reused multiple times, whereas copper foil may rip or break after a few adjustments. Lines of conductive paint can not easily be reused but may be modified by painting over them or scratching away parts of them.

Overall, FlyEye is least suited for making incremental changes to sensor layout or density. HandSense or TDRtouch both support incremental design processes, albeit in different ways. As TDRtouch usually employs cables/traces on the surface of the object, changing the sensor layout does not require opening the prototype.

### 8.2.10 Summary: Good, Fast, Cheap, Pick Any Two

*"Everything is best for something and worst for something else."* (Buxton 2007)

As described in the previous section, all three proposed techniques for prototyping grasp-sensitive surfaces Table 8.1 lists subjective rankings of the three techniques for each of the previously described requirements.

| | HandSense | FlyEye | TDRtouch |
|---|---|---|---|
| spatial resolution | + | +++ | ++ |
| sensor bit depth | +++ | + | ++ |
| latency | +++ | ++ | ++ |
| support for non-planar surfaces | + | ++ | +++ |
| deformability | + | ++ | +++ |
| proximity sensing | +++ | + | ++ |
| ease of use | + | ++ | +++ |
| cost | +++ | ++ | + |
| incremental refinement | ++ | + | +++ |

Table 8.1: Comparison of quantitative, qualitative, and prototyping-related properties of the proposed prototyping techniques for grasp-sensitive surfaces.

If each of these requirements were assigned the same importance, the overall ranking based on absolute number of "+" symbols or determined by Condorcet voting[2] would be: $TDRtouch > HandSense > FlyEye$. In practice, application-specific requirements determine which weight needs to be given to individual requirements.

Engineering and design often require trade-off decisions to be made between different qualities. In project management, this requirement for trade-offs is exemplified by the saying *"Good, fast, cheap - pick any two"*[3].

While this principle for projects can not be generally applied to tools, it holds true for the three prototyping techniques presented in this dissertation. HandSense, FlyEye, and TDRtouch occupy different sides of the so-called *project triangle* (Figure 8.1).

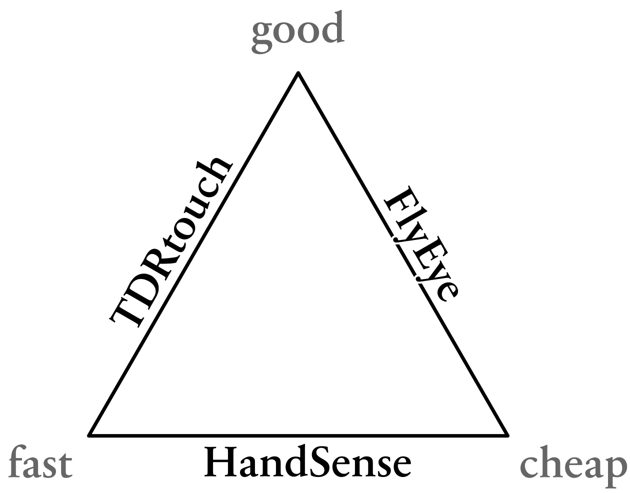


**Figure 8.1:** The three presented approaches with regard to three basic requirements of a prototyping technique: speed, cost, and quality

**HandSense** is a fast and cheap prototyping technique with limited resolution. **FlyEye** offers a high resolution and is cheap but requires significantly more setup time. **TDRtouch** allows for rapid, incremental prototyping and offers a high resolution. However, at the moment adequate reflectometers are rare and expensive.

In summary, TDRtouch would be the preferable prototyping technique, were it not for its high initial cost and its non-uniform coverage of surfaces. Therefore, reducing the cost of high-resolution TDR seems like a worthwhile endeavour. HandSense is the right approach for quick prototypes that do not require high resolution. FlyEye offers

[2] As confirmed using the online tool at `http://www.cs.wustl.edu/~legrand/rbvote/calc.html`, the ranking is the same for all commonly employed Condorcet voting methods.

[3] Although often quoted, the origins of this saying seem unclear. A quick Google Scholar search showed the first mention of this phrase by Nogami (1982). It probably dates back even further, as documented in `http://www.barrypopik.com/index.php/new_york_city/entry/fast_cheap_good_pick_any_two`

a high resolution but is less suited for iterative rapid prototyping than the other two techniques.

## 8.3 Comparison to Related Work

Being prototyping tools for grasp-sensitive surfaces, HandSense, FlyEye, and TDRtouch need to be compared to both prototyping tools and grasp-sensitive surfaces.

### 8.3.1 Grasp-Sensitive Surfaces

Section 4.4 presented a number of prototypical implementations of grasp-sensitive objects. Only one of these has employed commercially available grasp-sensitive surfaces (Kauffman et al. 2003). Instead, most researchers have built their own sensors for various reasons.

Looking at which of these surfaces could have also been implemented with HandSense, FlyEye, or TDRtouch gives an indication of the practical utility of these tools, and shows which requirements are not yet covered by them. As most of the publications from related work do not provide enough information to replicate sensing hardware and post-processing algorithms it is not possible to reliably compare their actual performance requirements. Therefore, only a subjective comparison of required spatial resolution and achievable resolution using HandSense, FlyEye, and TDRtouch is made.

Several early prototypes use only a small number of touch-sensitive electrodes (Harrison et al. 1998; Hinckley and Sinclair 1999; Hinckley et al. 2000; Mäntyjärvi et al. 2004). They could easily be implemented using HandSense.

Several other prototypes employ high-resolution capacitive or resistive sensor matrices (Kauffman et al. 2003; Pai et al. 2005; Kim et al. 2006; Taylor and Bove 2009; Song et al. 2011; Liu and Guimbretière 2012; L.-P. Cheng et al. 2012a). While many of these may be implemented using FlyEye or TDRtouch, some require a very high spatial resolution that is not yet achievable with these techniques (Song et al. 2011; Liu and Guimbretière 2012). As discussed in Section 5.7, in some cases a low-resolution technique such as HandSense may successfully used instead of high-resolution sensors.

Sato, Poupyrev, and Harrison (2012) use only a single electrode with spread-spectrum capacitive sensing. The presented applications could also be realized using HandSense.

Overall, the three techniques presented should be sufficient for building most of the presented grasp-sensitive artifacts. However, both prototypes which sense how a user grasps a pen use sensor grids with a very high resolution. It is not clear, whether such high-resolution sensors indeed offer superior classification performance than a lower-resolution sensor. Investigating required minimum resolutions for reliable grasp recognition would be worthwhile topic for future research.

### 8.3.2 Prototyping Techniques

While prototyping tools for physical computing are in widespread use (e.g. the Arduino board and ecosystem), tools for prototyping touch- or grasp-sensitive artifacts have been rare. Hudson and Mankoff (2006) presented a sensor board, *BOXES*, which supports up to eight binary touch sensors. BOXES is very similar to HandSense but offers only binary resolution and no proximity sensing.

Since HandSense, FlyEye, and TDRtouch were published, a number of new prototyping tools for touch- and grasp-sensitive surfaces have been presented. While only none of them are specifically intended for grasp sensing, most may also be used for this purpose. In addition, these tools show directions for further improving the techniques underlying HandSense, FlyEye, and TDRtouch.

**TactileTape** (Holman and Vertegaal 2011) is essentially a linear potentiometer on a flexible substrate. It offers cheap, flexible, single-touch sensing. However, it does not support multiple touches or touch-enabling surfaces.

**Midas** (Savage, Zhang, and Hartmann 2012) is a sophisticated prototyping tool for capacitive sensors. A GUI editor allows for defining shape and layout of the sensor electrodes which can then be printed, etched, or engraved. The system automatically takes care of routing the PCB traces. The controller board used with Midas supports connecting up to 20 binary touch sensors or a smaller number of touch sliders consisting of multiple individual electrodes. While not explicitly intended for prototyping grasp-sensitive surfaces, Midas may also be used for building simple ones. Its GUI significantly eases sensor layout. However, similar to HandSense, users have to assemble a controller board.

The **cuttable multi-touch sensor** presented by Olberding et al. (2013) is a flexible substrate with a grid of individual capacitive sensor electrodes connected to a central controller via traces laid out in a tree or star topology. This allows for cutting the substrate to nearly arbitrary shapes while still preserving connection of most sensing electrodes. The implementation presented in the paper supports up to 48 binary sensors. Using conductive inkjet printing, anyone can easily print such a sensor substrate. The substrate itself is flexible but not stretchable. However, by cutting out slices, the substrate may be wrapped around differently shaped objects without the need for stretching parts. Therefore, the cuttable multi-touch sensor offers a versatile option for prototyping grasp-sensitive surfaces. This use case is not discussed in the paper, however. Due to the sensor layout, spatial resolution is rather limited, similar to HandSense.

**Printed optics** (Willis et al. 2012) allow for embedding arbitrarily shaped optical waveguides within 3D-printed models. This approach allows for offloading the task of embedding optical fibers in an object to a machine. However, as suitable 3D printers are slow, prototyping speed may not be improved significantly.

**PrintSense** (Gong et al. 2014) is a flexible substrate with printed electrodes. A versatile controller supports capacitive touch sensing and proximity sensing, as well as resistive force sensing and detection of folding. One of the use cases demonstrated in the paper is distinguishing four different grasps on a cylindrical object. As PrintSense is a successor of the cuttable multi-touch sensor by Olberding et al. (2013), its uses are similar. Resolution is also similar to that of HandSense. The ability to also sense touch force may help with grasp classification.

**Resigraphs** (Holman, Fellion, and Vertegaal 2014) are essentially resistor arrays with force-activated shortcuts. While the authors suggest that up to 30 touch locations can be distinguished, only a single touch at a time can be detected. Unfortunately, the system is not described in sufficient detail to determine the actual usefulness.

In summary, while no more sophisticated prototyping tools for grasp-sensitive surfaces have been published yet, several prototyping tools have been published which may also be used for grasp sensing, or which show ways for improving the three techniques presented in this dissertation. In particular, recent developments in printing of flexible PCBs and three-dimensional structures allow for quickly and more or less cheaply printing arbitrary sensor layouts.

While prototyping sensors has been made cheaper and more accessible, all of the presented prototyping techniques for flexible substrates offer only a very limited spatial resolution of less than one centimeter. High-resolution touch- or grasp-sensitive substrates seem like a worthwhile topic for future research.

## 8.4 Summary

HandSense, FlyEye, and TDRtouch were the first prototyping techniques for grasp-sensitive surfaces. Their capabilities complement each other, allowing designers to choose the most suitable one.

HandSense is a versatile tool built on CapToolKit. It was the first commercially available prototyping tool for capacitive touch sensing and proximity sensing. Using HandSense, I demonstrated that a small number of high-resolution sensors may offer similar performance as more sophisticated sensor designs. HandSense is the only technique that allows for sensing not only touch but also the thickness of the touching tissue, which allows for distinguishing different parts of the hand.

FlyEye allows for prototyping touch- and grasp-sensitive surfaces almost without custom electronics. The relative mapping algorithm developed for FlyEye allows for easy calibration of sensors with unknown mapping of inputs.

TDRtouch allows for sensing touches and grasps using a single cable. We were the first researchers to determine the performance of TDR for touch sensing and to explore the design space. With TDRtouch, we also presented for the first time modular and deformable touch-sensitive surfaces.

All three techniques had a significant impact on other researchers which not only shows in several citations. In multiple cases, the publications prompted other researchers and practitioners to ask for my advice, which resulted in several cooperations.

Shortcomings of the presented research include: a lack of qualitative hands-on evaluations with designers and a lack of measurements quantitatively comparing all three techniques to each other.

Future work furthermore includes the design of a versatile high-resolution grasp-sensitive surface and development of a predictive model of spatial sensor resolution incorporating different sensing techniques (Wimmer 2010b).

# III

# Towards Grasp Interaction

# Chapter 9

# Challenges for Grasp Interaction

*In essence, grasp interaction is concerned with the question '*Why *did the user grasp the object in exactly this way?' Despite over 100 years of research on human grasping - and nearly 20 years on grasp sensing - we still know little about how and why people grasp in everyday interactions. If we implement grasp-sensitive artifacts without understanding grasping, bad things can happen. In addition to correctly interpreting grasps, a major challenge is making grasp interaction usable.*

**Attribution**: This chapter only contains original content written for this dissertation.

## 9.1 Introduction

As the previous chapters have shown, grasp sensing may facilitate and enhance rich interaction with graspable objects. However, I argue that we do not really understand how grasp interaction can be implemented safely and user-friendly at the moment. In this chapter I summarize the current state of research and show challenges that need to be addressed.

As discussed in Section 3.5, grasp sensing can be divided into three steps: capture, identify, and interpret.

Techniques for capturing grasps have received a lot of attention by researchers in the past. While implementation of grasp-sensitive surfaces is still challenging, the basic requirements and approaches are are well-understood. I have presented three prototyping techniques for grasp-sensitive surfaces. For mass-production, various technologies exist, with capacitive sensing being the most mature and versatile. Flexible PCBs with custom electrode patterns can be pre-shaped into many forms. Off-the-shelf chips for capacitive sensing are available now from many manufacturers, such as Analog Devices, Atmel, or Microchip.

For identifying grasps, several approaches have been proposed. It is necessary to unify existing approaches and address variability in grasps in the future. However, a simple, ad-hoc, general-purpose approach could be to reduce the dimensionality of grasp signatures using PCA and to classify them as one of a few grasp types using a machine-learning classifier such as SVM.

The greatest challenge is reliably interpreting grasps, however. This issue has not been solved, and has not even been investigated so far. While difficulties in correctly interpreting grasps are mentioned in passing in several papers, they are never addressed. Only once we are able to reliably interpret grasps, grasp interaction will become a viable interaction technique.

Being aware of the challenges in interpreting grasps is a necessary precondition to addressing these challenges. The following section illustrates what can - and will - get wrong when using a poorly conceived grasp-sensing artifact.

## 9.2 A More Realistic Scenario

(This section is a continuation and variation on the day-in-the-life scenario described in Section 1.2.)

> Tom wakes up to the sound of his mobile phone's wakeup alarm. It is Friday, the 13th. He slightly *squeezes* the phone, thereby silencing the alarm tone and activating the snooze function. Or so he thinks. In fact, Tom has sleepily pressed the phone's power button while squeezing it, turning off the phone. Tom sleeps a little bit longer.
>
> Tom wakes up to the sound of his landline telephone ringing. His co-worker, Sally, is calling, wondering if Tom forgot the meeting that is happening right now. She is a little bit upset.
>
> Tom apologizes, quickly showers, *picks up* his phone, powers it on, and runs to the subway. The phone's capacitive sensors are slightly confused by the wet hands but the classifier happily recognizes "Grasp #13" and accordingly launches the music player. Hurrying towards the subway station, Tom *fumbles* to shut off his phone which is loudly playing Death Metal. People are looking.

> In the subway Tom quickly checks his e-mail in order to find out in which meeting room he needs to be since thirty minutes ago. As line break handling is still an unsolved problem, the room number is cut off by the e-mail client. Tom *rotates the phone into landscape mode* so that he can read the whole line containing the room number. The phone recognizes this grasp and it automatically activates the camera application, allowing Tom to focus on taking a photo. As the subway is only dimly lit, the phone activates its flash LED. Other passengers are not amused.
>
> Tom embarrassedly enters the meeting room, apologizes, and sits down. He *puts the phone on the large steel table* which is covered by a thick tablecloth to mitigate noise. The phone can not reliably distinguish a metal plate from a human body and assumes that Tom has put it into his pocket. Therefore, it switches from acoustic to haptic notifications. Meanwhile, Sally urgently needs to contact Tom. She desperately calls his phone every few minutes but his phone's vibration motor is silenced by the thick tablecloth. Sally is really annoyed now.
>
> Soon it is Tom's turn to present the annual earnings forecast. As Tom's laptop is still on his desk, Tom opens the presentation slides on his phone, *handing it to the customer*. The biometrics subsystem is quite sure that Tom just handed his phone to his wife. It automatically opens the image gallery that Tom's wife had open when she last used Tom's phone. The customer does not really appreciate the insight into Tom's love life.
>
> Tom's phone detects a *power grasp being released* while the accelerometers indicate rapid horizontal acceleration. The phone emits an excited "Wheeeeeeeee!" until it crashes into a wall.

While this scenario may be slightly exaggerated, it shows how grasp interaction can go terribly wrong in lots of different ways. Misunderstandings between user and system are especially critical on mobile phones, as these often hold our personal data and connect us to friends, family, co-workers, customers, and the digital world.

However, incorrect interpretation of a grasp may also have dire consequences on other graspable artifacts: With a biometric gun, false positives and false negatives can have drastic effects. A grasp-sensitive dial on a radiation therapy machine might confuse a precision grip with a power grip in one of thousand uses and increase radiation power by 1000 units instead of 1 unit. Even in non-critical applications, incorrect recognition or interpretation of a grasp may lead to annoyed users, lost time, or even data loss.

With powerful technologies at our fingertips, doing something wrong is all to easy. But how can we do it right?

## 9.3 Getting Grasp Interaction Right

The previous scenario focused on problems caused by incorrect capturing, identification, or interpretation of grasps. Reducing such errors is one of the two major challenges in grasp interaction. To this end, more research is needed - especially regarding identification and interpretation of grasps.

The first step to solving a problem is understanding it (Polya 1957). However, the very first step to understanding a problem is being aware of it.

As the related work presented in Chapter 4 shows, researchers are rarely aware of the limited understanding we have of human grasping. Therefore, I present a study in Chapter 10 that demonstrates the variability and unpredictability of grasps in simple but realistic tasks.

In Chapter 11 I propose a descriptive model of meaning in human grasping - called GRASP - that can act as a framework for discussing and systematically analyzing use cases and limitations of grasp-sensing objects.

The other great challenge in grasp interaction is making it usable. Finding the right balance between usability and reliability is not trivial. For example, requiring the user to conduct a series of unusual grasps for even the simplest action - ideally with both hands at the same time - certainly reduces the probability of errors in capturing, identifying, and interpreting grasps. However, such requirements severely limit the usability of grasp interaction.

While previous work has focused on increasing reliability of grasp sensing, usability of grasp interaction has not found much attention so far. This final part of the dissertation offers some observations and thoughts on this problem.

In Chapter 12 I discuss the relationship of grasp interaction to other interaction paradigms and present preliminary guidelines for implementing usable grasp interaction.

Finally, I critically discuss my research presented in this thesis in Chapter 13 and offer an outlook into the brave new world of grasp interaction.

Chapter 10

# How Do People Grasp a Mobile Phone?

*We conducted a study with 20 participants, investigating how people grasp three different mobile phones for four different tasks. Our study shows that even for very simple tasks, and even if the phone provides affordances, people always grasp it in at least two distinctly different ways. While some combinations of task and phone model cause a majority of users to employ a certain grasp type, it is not yet clear whether there is a pattern to these combinations. Additionally, there will always be users who employ extremely exotic grasps.*

**Attribution:** The study presented in this chapter was designed by me and three students I supervised as part of a research course, Constantin Scheuermann, Marc Mühlbauer, and Nadezda Mikhaylova. The students conducted the study, documented it, and initially analyzed some of the data. They received course credit for their work. I annotated the captured photos, re-analyzed parts of the data, and discuss the implications of the study in the following. The results are published here for the first time.

## 10.1 Motivation

Mobile phones are ubiquitous, feature-rich artifacts that usually require the user to grasp them in order to interact with them. Therefore, they are very interesting test-beds for grasp interaction. As discussed in Section 4.4, several researchers have equipped mobile phones with grasp-sensitive surfaces (Kim et al. 2006; Wimmer and Boring 2009; Iso et al. 2012; Goel, Wobbrock, and Patel 2012). However, none of these publications discuss how people actually grasp mobile phones. Instead, researchers define grasp types which they associate with a certain action and try to recognize these grasps as precisely as possible. Due to this approach, these studies might have a low external validity.

Research on everyday grasps (see Chapter 2.4) so far has either focused on very constrained tasks, such as cylinder grasping, or has investigated a very broad range of different objects and grasps. None of the studies required participants to conduct different actions with each grasped object.

To complement the insights gained from these studies, we conducted a study in which we investigated how people grasp different mobile phones for different everyday tasks. This study was conducted in 2010. Our goal was to gather qualitative information about how variably users grasp mobile phones. To this end, we had 20 participants conduct four common tasks with three differently-shaped mobile phones. Static grasps were photographed and annotated.

## 10.2 Participants

For the study, we recruited 20 participants (15 male), age range 20-33, mean age 24. Five of the participants were left-handed (four male). Most participants were media informatics students at the University of Munich. All participants owned and used at least one mobile phone. None of the participants suffered from a limitation of hand function. Of the 20 participants, 8 primarily used a mobile phone with a standard numeric keypad, 3 used a mobile phone with a clamshell design and numeric keypad (Motorola Razr), 2 primarily used a mobile phone with a stylus, and 7 used a mobile phone with a touchscreen (4 iPhones). All participants were briefed about the study and agreed to be photographed. Participants received no compensation.

## 10.3 Phone Form Factors

In order to judge the effect of the phone's form factor on the employed grasp, we had each participant conduct all tasks on three different phones:

a) **Nokia XpressMusic 5800** - a Symbian OS smartphone with a resistive touchscreen which is operated via a stylus. Dimensions (HxWxD): 111 x 51,7 x 15,5 mm (Figure 10.1a).
b) **Motorola Razr2** - a flip/clamshell mobile phone with a numeric keypad and no touchscreen. Dimensions (HxWxD): 103 x 53 x 13 mm (Figure 10.1b).
c) **T-Mobile Pulse** - an Android smartphone with a capacitive touchscreen. Dimensions (HxWxD): 116 x 62.5 x 13.5 mm (Figure 10.1c).

These phones were chosen as they represented three prevalent classes of mobile phones in 2010 and required three different input techniques. We opted not to include a classic non-clamshell mobile phone as it is very similar to the Motorola Razr2. The major

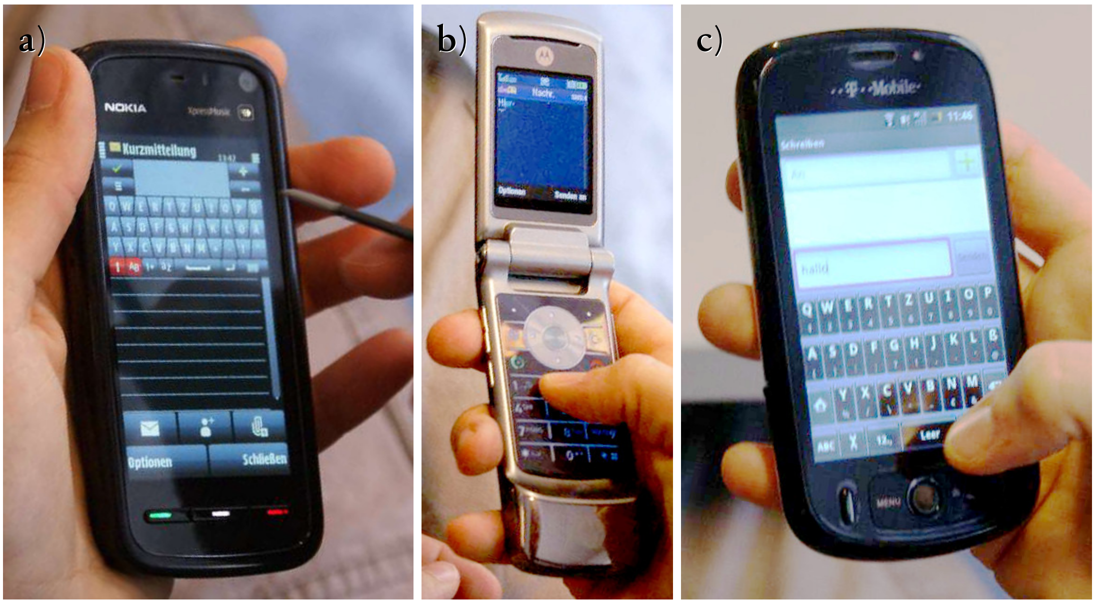


**Figure 10.1:** Three phone models were used in the study: Nokia XpressMusic 5800 (a), Motorola Razr2 (b), and T-Mobile Pulse (c).

difference is that a clamshell phone needs to be opened before using it. Once opened, it has a similar width and screen/keypad arrangement as a classic mobile phone but is slightly longer. We asked participants to always use the stylus when operating the Nokia XpressMusic 5800 phone.

## 10.4 Tasks

In order to simulate everyday usage of a mobile phone, we chose 4 short tasks that are both common and require different grasps. In all tasks the participant sits in front of a desk, picks up a phone lying on the desk, holds it, presses one or more buttons, and puts it back onto the desk:

1) **Make a phone call.** The phone lies in front of the participant on the desk, centered to the participant's position. The participant picks up the phone, dials a random number, holds it to his ear, waits a short time, presses the disconnect button on the phone, and finally puts the phone back onto the desk.
2) **Send a text message.** The phone lies on the desk; the messaging application has already been started by the experimenter. The participant picks up the phone, types the word "Hallo", presses a button to close the application, and puts the phone back onto the desk.
3) **Take a photo.** The phone lies on the desk; the camera application has already been started by the experimenter. A square piece of cardboard with a smiley face

drawn on it hangs in front of the participant in a distance of about 1 m. The participant picks up the phone, points the camera towards the smiley and presses the appropriate button to take a photo. Afterwards, she puts the phone back onto the desk.

4) **Answer the phone.** The phone lies on the desk. The user picks up the phone, presses the 'accept call' button, and puts the phone back onto the desk. This task is very similar to the first task. We included it to see whether the participant would pick up the phone in the same way. As the participants only had to press a single button instead of dialing a phone number, we wondered whether they might employ a slightly different grasp.

We chose to limit the experiment to these 4 tasks to keep the analysis straightforward. In these tasks we could easily control the participants' position with regard to the phone. While it might have been insightful to let participants pull out the phone from a pocket or purse in the beginning, the participants choice of clothing might have had a greater effect on the grasps than the tasks or phones.

## 10.5 Study Design

Participants were first informed about the goal and procedure of the study. As the study aimed to document how people intuitively grasp phones, we did not want the participants to feel being tested. Therefore, neither task completion times nor error rates were measured. Participants were told that any errors they made would not be relevant for this study. They were encouraged to ask questions at any time.

The phones were lying directly in front of the participants on a desk, centered with regard to the participants. An experimenter explained and demonstrated each task before the participant had to complete it. As we did not want to influence how the participants grasped the phones in any way, the experimenters did never pick up the phone themselves while showing the tasks. After the experimenter had explained the task, the user picked up the phone and completed the task. During each task, photos of stable grasps were taken from two angles - over the shoulder and from the side. In task 1, photos were taken of four grasps: pick-up, dialing a number, holding the phone to the ear, and putting the phone back onto the desk. In the other cases, only one stable grasp was captured: holding the phone while pressing a button.

Participants completed all tasks in the same order as presented above, first with the Nokia, then with the Motorola, and finally with the T-Mobile phone. We did not counterbalance or randomize task order and phone order, as we did not expect a learning effect. After all, the participants had been regularly using mobile phones before. We also did not repeat any of the action/phone combinations a second time. Keeping the same sequence of tasks across all sessions allowed the experimenters to keep all sessions very similar to each other, mitigating potential confounding variables.

Participants completed all twelve tasks (three phones x four tasks) within 10-15 minutes. Afterwards they were asked to fill out a short questionnaire. We also measured width and length of their preferred hand.

## 10.6 Results

While we expected participants to use slightly different tasks, we were surprised about the variety of grasps we encountered. We found great differences between grasps for all tasks and mobile phone types.

### 10.6.1 Picking Up the Phone

Generally, all participants picked up all phones using a *Prismatic 4 Finger* grasp (Figure 10.2a). However, in some cases individual fingers did not touch the phone. This was especially prominent with the small Razr2 phone whose size made it hard to use all five fingers for grasping. Five participants (5,8,12,16,18) alternated between this basic grasp and a modified version wherein they positioned their index finger at the top end of the phone. Instead of the tip of the middle finger, its side touched the phone (Figure 10.2b). There is no apparent pattern to the use of this modified grasp - it is used for all three phone types. Only one person always placed their index finger at the top end, the others alternated between both grasps. One person (14) employed an uncommon grasp when picking up the stylus-operated phone. She picked up the phone with thumb and middle finger being at the top and bottom end, stabilizing it with the ring finger (Figure 10.2c). This grasp can not be easily attributed to one of the grasp types in Feix' taxonomy. The participant only used this grasp once in the beginning and switched to the basic *Prismatic 4 Finger* grasp for the remainder of the experiment.

Of the 20 participants, 16 always picked up the phone with their preferred hand. One right-handed person always picked up the phone with the left hand. In addition, three persons (6,14,18) picked up the different phones with different hands. The type of phone had no apparent effect on this behavior.

### 10.6.2 Holding the Phone to the Ear

We found three predominant grasps for holding a mobile phone to the ear. In all three cases, the thumb is positioned on one side of the phone, and fingers 3-5 are positioned on the opposite side:


**Figure 10.2:** For the very simple and unconstrained task of picking up a mobile phone from a desk, most participants employed some variant of a *Prismatic 4 Finger* grasp (a). Some participants sometimes employed a modified grasp, placing the index finger at the top end of the phone (b). Additionally, one person employed a unique grasp when picking up the Nokia phone (c).

- a *Prismatic 4 Finger* grasp, where all fingers rest on the sides of the phone (in the following called "side grasp") (Figure 10.3a),
- a modified version of this grasp, where the index finger is positioned on the back of the phone ("back grasp"), stabilizing it (Figure 10.3b),
- an in-between state where the index finger rests on the edge between back and side of the phone ("edge grasp") (Figure 10.3c).

One participant (9) held the Nokia phone only between thumb and middle finger, pressing it to the ear with the index finger. The adducted ring finger laterally supported the phone. This is one of several grasps that are not included in Feix' comprehensive taxonomy. The participant used standard grasps for the other tasks and phones.

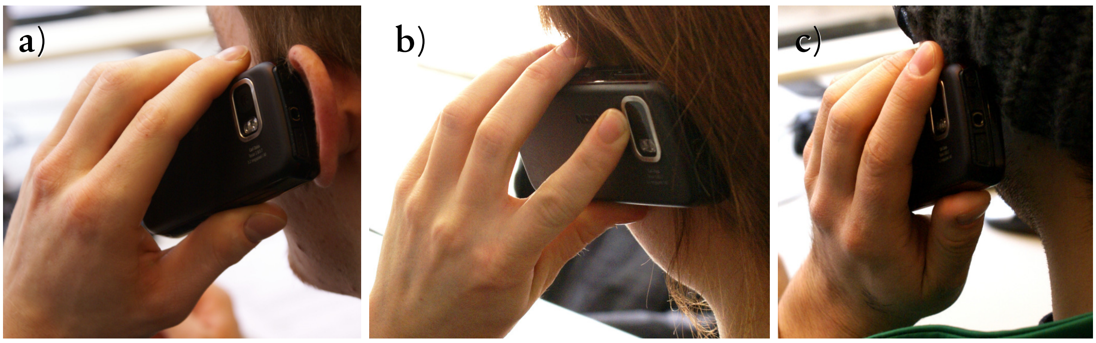


**Figure 10.3:** Most participants held the phone to the ear using a *Prismatic 4 Finger* grasp. However, the index finger was positioned differently - either on the side (a), the back (b), or the edge of the phone (c). The chosen posture was partially determined by hand size and phone width.

There were distinct differences between phones. The Nokia phone was mainly grasped using a back grasp (8 times) or edge grasp (9 times), rarely was a side grasp used (2 times). The Motorola Razr2 was almost equally often grasped with the index finger on the back (8), on the edge (6), or on the side (6). The T-Mobile Pulse, being 10 mm wider than the other two phones, was mainly grasped using a back grasp (16), only rarely was the index finger located on the edge (2) or side (2). There is an obvious but not coherent correlation between hand size and employed grasp. Participants with very short hands (< 160 mm in length) were more likely to employ back and edge grasps, whereas participants with long hands (> 190 mm) usually employed edge and side grasps. However, there was much variability. One participant (5) with a quite average hand length of 170 mm always put his index finger on the side of the phones. Another participant (18) with a hand length of 190 mm always employed a back grasp. Therefore, hand size alone does not completely explain the different grasps. Habits and flexibility of the fingers probably played a role, too, in choosing a grasp.

### 10.6.3 Taking a Photo

Mobile phones are generally used to take photos in either portrait orientation (phone upright) or landscape orientation (phone horizontal). We asked participants to take a photo of a square piece of cardboard. No further directions were given, and we did not indicate the expected orientation of the photo. Like in the other tasks, the phones were lying on the table in the standard vertical orientation. We therefore assumed that participants would grasp the phone in their preferred way.

The Nokia and T-Mobile phones were generally held in landscape orientation (17, resp. 15 of 20 cases), whereas the Motorola phone was mostly held in a portrait orientation (18 of 20 cases). This can probably be attributed to the special form factor of the Motorola Razr2.

Each respective orientation of the phone affords one or more appropriate grasps. In portrait orientation, most participants held the phone only in one hand using a *Fixed Hook* palmar grasp (Figure 10.4a). They used the thumb of this hand for pressing the shutter button. While most participants grasped the phone with their dominant hand, one left-handed participant (18) held a phone with their right hand, and two right-handed participants (19,20) held a phone in the left hand. For the Motorola Razr2 and for the T-Mobile Pulse there was one participant each (17, 20) who used both hands for stabilizing the phone (Figure 10.4b). One other participant (4) employed a quite exotic grasp when holding the Nokia and T-Mobile Phones (Figure 10.4c).

However, when holding the phone in landscape orientation, most participants employed both hands. Three distinct grasps could be observed. A majority held the phone in a *Palmar Pinch Grasp* between the tips of both thumbs and index fingers, grasping the phone on its four corners (Figure 10.5a). The Nokia phone was held almost exclusively in this way, and a majority held the T-Mobile phone in this way. Five participants held the T-Mobile Pulse in a modified grasp (Figure 10.5b). In this grasp, thumb and index finger implement a *Ring Grasp*, while the remaining fingers are placed on the back of the

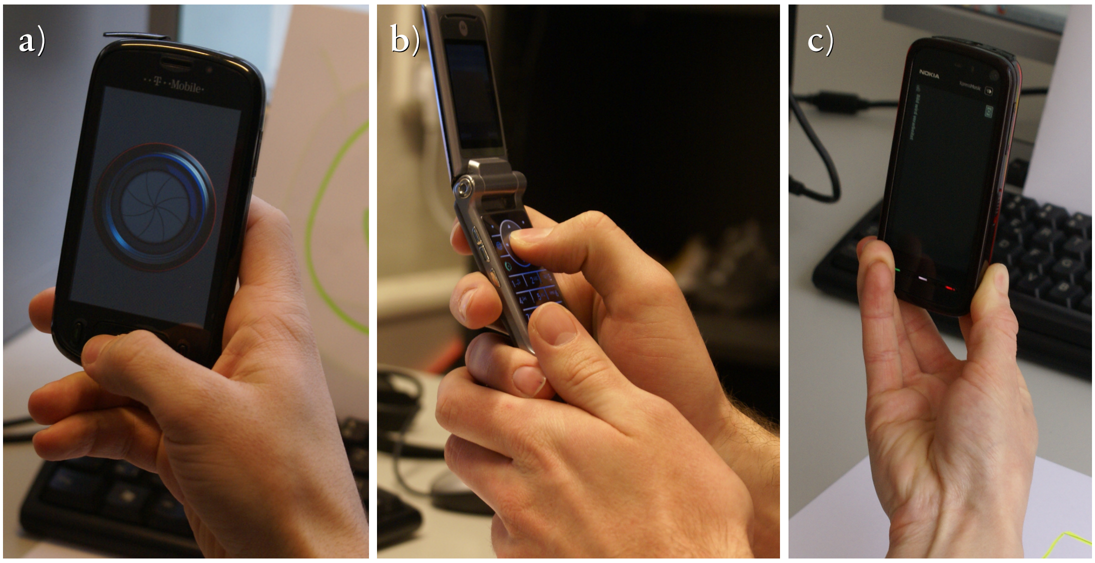


**Figure 10.4:** For taking a photo in portrait mode, most participants held the phone in a *Fixed Hook* palmar grasp (a), operating it with their thumb. Two participants additionally stabilized the phone with the other hand (b). One other participant employed a quite unique grasp for holding the phone (c).

phone to stabilize the grasp. The Motorola phone was used in landscape orientation in two cases. In those cases, the participants held the phone in the same *Fixed Hook* grasp as employed for portrait orientation but tilted their hand to the left (Figure 10.5b)

## 10.6.4 Dialing a Phone Number

For dialing a phone number, three different approaches were observed:

a) holding the phone in one hand and operating the buttons with the thumb of the holding hand.
b) holding the phone in one hand and operating the buttons with the index finger of the other hand.
c) holding the phone in both hands and operating the buttons with the thumbs of both hands.

All participants held the phone in portrait orientation. A majority of participants always used their dominant hand for operating the phone. However, six participants alternated the hand for different phones.

For the Nokia, we asked participants to use the stylus instead of a finger. Therefore, all participants except one switched the phone to the non-dominant hand, entering the

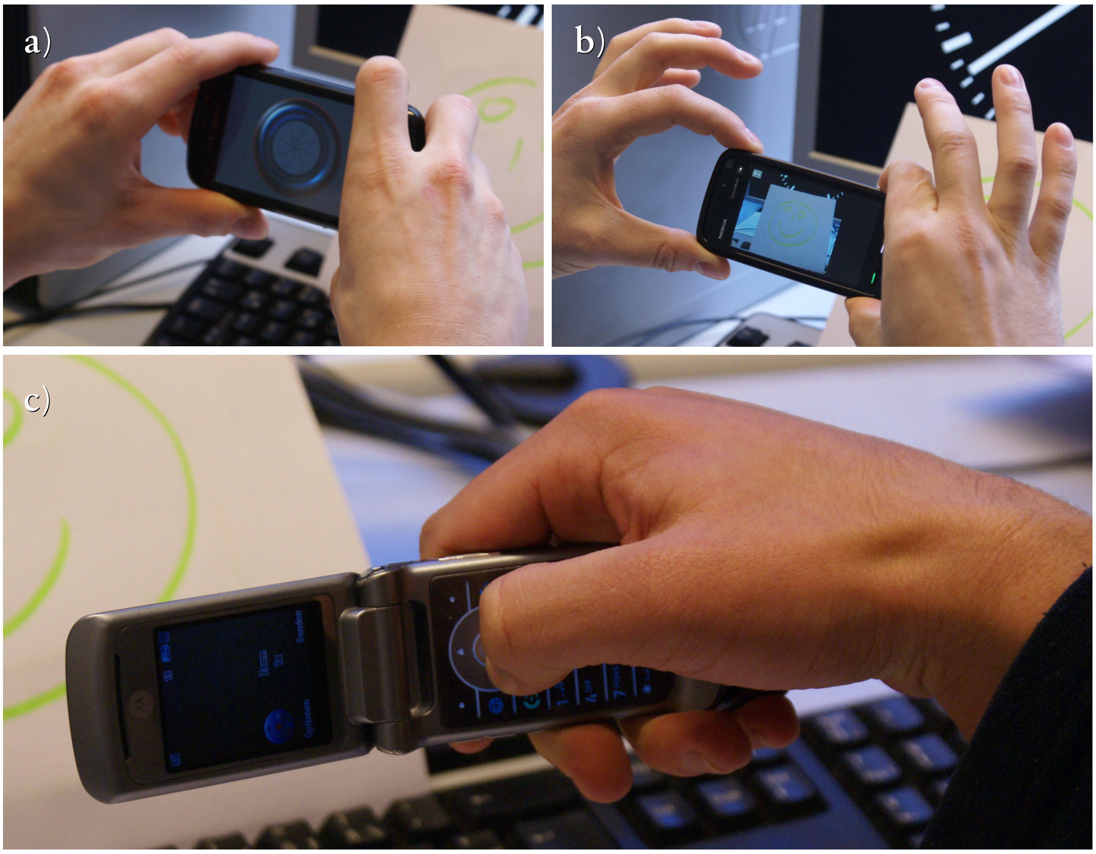


**Figure 10.5:** For taking a photo in landscape mode, most participants held the phone with both hands. They used either thumbs and index fingers (a), or the whole hand (b). Two participants held the Motorola phone in landscape orientation (c).

phone number with the stylus held in the dominant hand. Participant 8 ignored our request, holding the phone in his right hand and typing with the thumb.

For the other two phones, no obvious usage patterns could be observed. None of the participants used only one approach, and both Motorola Razr2 and T-Mobile Pulse were operated with all three approaches.

For typing the text message and accepting a call, participants used very similar grasps. Therefore, those tasks are not discussed in detail.

When typing a text message, most participants held the phone in portrait orientation. One participant held the Nokia phone, none held the Motorola phone, and seven participants held the T-Mobile phone in landscape orientation.

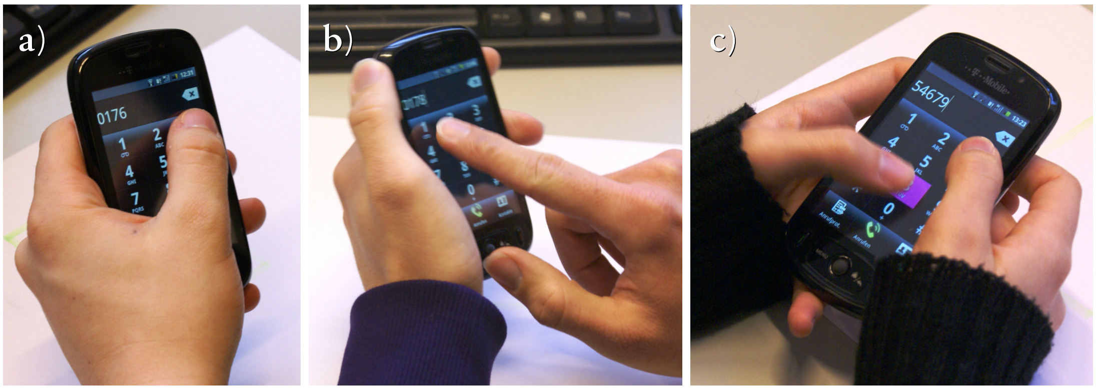


**Figure 10.6:** For dialing a phone number, participants employed three different grasps: holding the phone in one hand and operating it with the thumb of the holding hand (a), holding the phone in one hand and operating it with the index finger of the other hand (b), or holding the phone in both hands and operating it with the thumbs of both hands (c).

## 10.7 Discussion

Overall, we observed that participants employed hugely different grasps for the same tasks and phones. While most grasps could be attributed to two or three general grasp types for a task, we always observed at least one participant employing a completely different and unexpected grasp.

Picking up a mobile phone lying on a table is an extremely simple task with a limited number of sensible grasps. In our study, we also positioned the phone always in the same place and orientation. Not taking photos of the pick-up grasp for tasks 2-4 was an unfortunate omission. Therefore, we can not tell whether people picked up the same phone in different ways during the different tasks. It might be that every person always picks up their personal phone in the same way. However, even in the small sample of 20 participants, the exception turned out to be the rule. While we observed two predominant grasp types for picking up a phone, people regularly used different grasps. These inconsistencies show up in all tasks, for all phone types, and for many different participants.

It is important to better understand what can be inferred from specific grasps and what not. While a majority of participants picked up the phone with their preferred hand, nearly one third did not. On the other hand, people generally employ their dominant hand for entering text. Therefore, it would be dangerous to infer the handedness of a user from the way they pick up a mobile phone, but it might be inferred from the way they hold the phone during text entry.

In other cases, the employed grasp was determined less by the participants' handedness, but by the affordances of the mobile phones. Apparently participants were mostly used to take photos in landscape orientation - which they chose when using the T-Mobile or

Nokia phones. However, the shape of the Motorola Razr2 seems to have led most users to hold it in a portrait orientation.

We tried to control as many variables as possible in this study, including phone placement, phone types, and tasks. Participants could focus on their task without distractions or having to use their hands for a parallel task. Despite these measures we observed a huge variety in employed grasps. For real-life scenarios, one can expect even more variety in grasps.

The results of our study suggest that it is not feasible to associate a single grasp type with a certain action, as grasps vary tremendously for different users and even for a single user. Even for very simple tasks under carefully controlled conditions, and even though the grasped object provides affordances, users grasp in very different ways.

Instead of ignoring these differences, it is worthwhile to analyze the reasons.

# Chapter 11

# GRASP - a Descriptive Model of Meaning in Grasps

*GRASP is a descriptive model that aids in analyzing why users grasp objects in a certain way. The model comprises five aspects that determine a grasp: Goal (what the user wants to do), Relationship (assumptions about and emotions towards the object), Anatomy (neural and anatomical properties of the user), Setting (environmental factors and placement of the object), and Properties (intrinsic properties of the object). These aspects allow for systematically discussing issues in grasp interaction and analyzing existing and future applications.*

**Attribution:** This chapter is based on my paper "Grasp Sensing for Human-Computer Interaction" (Wimmer 2011a) presented at TEI 2011. Some definitions of the GRASP model have been reproduced verbatim or nearly verbatim to avoid conflicting descriptions in paper and dissertation. These are marked by explicit references to this paper.

## 11.1 Which Factors Determine a Grasp?

As shown in the previous chapter, people grasp objects in a variety of ways. With grasp interaction we usually want to support the users' implicit and explicit goals using grasp sensing. To this end, we need to know *why* the user grasped an object in a certain way. However, the user's goals are not the only factors determining a specific grasp.

Schlesinger (1919) argues that the shape of the grasped object affects the grasp. Napier (1956) explicitly describes object shape and task as (the only) factors. Cutkosky and Wright (1986) also propose distinguishing between those factors, arguing that "the choice of grasp is dictated less by the size and shape of objects than by the tasks they want to accomplish.". That people grasp differently for different tasks has also been shown by Steenbergen et al. (1997) and Ansuini et al. (2008)

However, there are several factors beside task and object shape. Jacobson-Sollerman and Sperling (1977) mention that the surrounding of an object also affects from which directions the hand can approach and which grasp can be employed. Eastough and Edwards (2007) show that knowledge of intrinsic object properties - such as weight - affects grasping. Siegel, Walker, and Stefanucci (2009) found that disgust changes the way people grasp objects. Veldhuis et al. observed and utilized that grasps vary between users. They also found that grasps change over time and that regularly grasping in a certain way results in more consistent grasps.

Despite these findings, none of the existing publications on grasp sensing (excluding my recent publications) explicitly acknowledge that multiple factors affect how an object is being grasped. Sometimes researchers implicitly acknowledge these factors, for example by controlling for a few of them.

A model that describes all factors determining a grasp allows researchers and designers to systematically analyze and incorporate these factors into grasp-sensing applications.

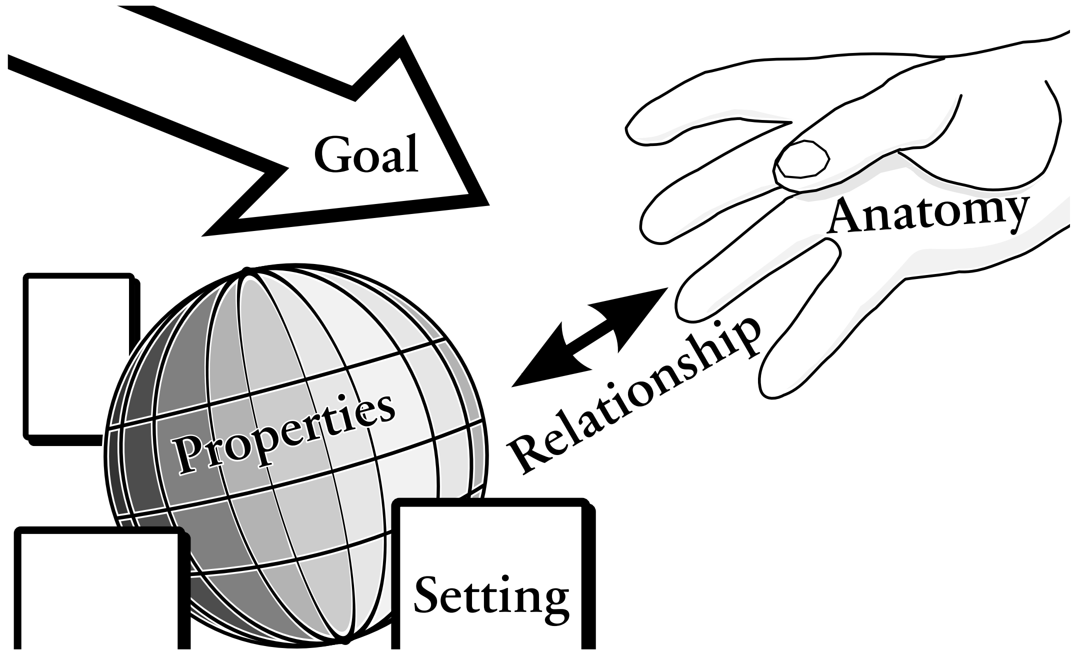


**Figure 11.1:** How one grasps an object conveys meaningful information. The GRASP model describes five distinct groups of meaningful factors that affect how humans grasp objects: The *Goal* the user wants to achieve, the *Relationship* between user and object, the user's *Anatomy*, the *Setting* in which the grasp takes place, and the intrinsic *Properties* of the grasped object.

MacKenzie and Iberall (1994 p.7) offer a rather mechanistic model of grasping. They describe a grasping person as a black-box system whose inputs are a goal and an object. This system then outputs a specific grasp. In addition to input and output, MacKenzie and Iberall define multiple constraints that also influence the employed grasp - such as placement and orientation of the object. MacKenzie and Iberall group these constraints into three classes:

- **High Level:** Social/Cultural, Motivational, Informational, Functional
- **Physical:** Object Properties, Biomechanical/Mechanical
- **Sensorimotor:** Neural, Perceptual, Anatomical/Physiological, Evolutionary/Developmental

However, MacKenzie and Iberall do not explain why and how they distinguished between the two *inputs* and the *constraints* although both affect a grasp.

In robotics the object to be grasped, the intended result of the grasp, and additional environmental factors are known. The challenge is to find a hand posture that allows for a stable grasp. This goal-centric approach offers only limited insight into actual human grasping. For grasp interaction, it is not important how an optimal grasp would look for a given object and goal. Instead, the challenge is the opposite: to take grasp information, infer all factors that contribute to this grasp, and use this information to support the user's explicit and implicit goals.

In order to analyze these contributing factors and extract meaningful information, we need to know them and understand how they contribute to a grasp.

A first step in this direction is my GRASP model (Wimmer 2011a), a descriptive[1] model of meaning in grasps.

> *Essentially, all models are wrong, but some are useful.*

George E. P. Box, Norman R. Draper (1987) Empirical Model-Building and Response Surfaces. p. 424

The primary goal when creating this model was to allow for a structured analysis of the design space for grasp interaction. It is intended as a tool for design and evaluation of grasp-sensing applications. For example, GRASP may support the design of a grasp-sensitive volume knob by providing a checklist of important aspects that need to be considered. GRASP can also be used to objectively find and discuss limitations of empirical research, for example, by identifying confounding variables that have not been taken into account in a study. Section 11.7 presents detailed examples.

[1] Unlike a *predictive* model, a *descriptive* model does not predict outcomes but structures relevant factors of a concept.

The GRASP model comprises five major aspects[2]:

- **Goal** - what the user wants to do with the object
- **Relationship** - the personal relationship between user and object
- **Anatomy** - sensorimotor properties of the user
- **Setting** - environmental conditions and location of the object
- **Properties** - all intrinsic properties of the object, including shape and texture

This division groups all factors contributing to a grasp into individually controllable and observable aspects. For example, hardware designers have control over the *properties* of the object but not over the other four aspects. For biometry, the users' *anatomy* should be detected, while the influence of the other aspects should not have an effect on the recognition result.

In the following, I describe and discuss the individual aspects, subjectively ordered from most to least important. Examples for each of the aspects are shown in Figure 11.2.

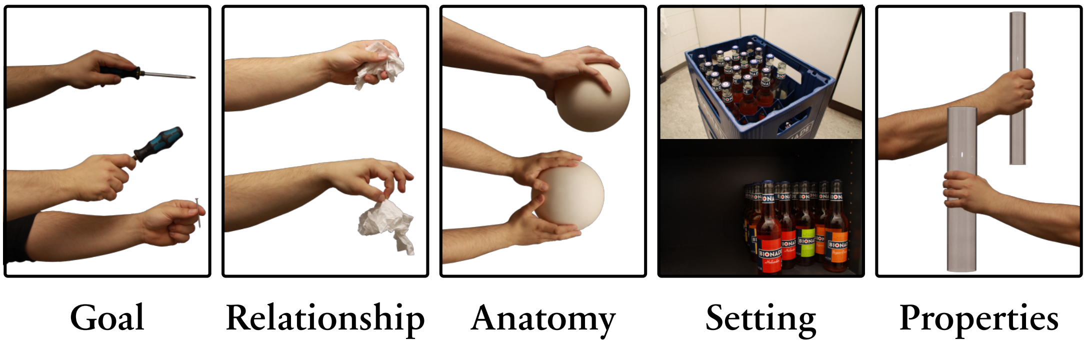


**Figure 11.2:** Examples of how each of the GRASP factors affect the grasp: A screwdriver is held differently, depending on the *Goal* that the user wants to achieve. When picking up a paper towel, a person's *Relationship* to the towel affects the type of grasp, depending on whether the towel belongs to the person or not. Even if two persons have the same goal and relationship to the object, their *Anatomy* - such as hand size - determines which grasp they need to apply. In different *Setting*s - a bottle standing in a crate or on a shelf - different grasps need to be applied. Finally, the intrinsic *Properties* of an object - such as size or shape - require according grasps.

[2] The first letters obviously form the initialism "GRASP". However, the division into these five aspects was not driven by the need for a good acronym. Originally, the "properties" aspect was called "object" and "anatomy" was called "user". As "GRUSO" bore some semblance to "GRASP", I decided to rename those aspects. As it turned out, this also significantly helps others in remembering all of the aspects.

# 11.2 Properties

The *properties* aspect includes all intrinsic properties of the graspable object. They have a major influence on how the user grasps the object. For example, a thin rod can be grasped with a precision grasp while a wide tube requires a power grasp - independent of any goal the user may have (Figure 11.2, right).

The intrinsic properties include:

- shape
- size
- deformability
- color, texture, softness, temperature of the surface
- weight and weight distribution
- variations of these properties over time

Not included are extrinsic properties such as position and orientation of the object, or the wetness of its surface. Thus, all factors within the *properties* aspect are under the control of the hardware designer or manufacturer and may be customized for different tasks.

Many of the intrinsic properties listed above define actual and perceived *grasp affordances*. The term *grasp affordances* has previously been defined in the context of robotics as "the different ways to place a hand or a gripper near an object so that closing the gripper will produce a stable grip." (Detry et al. 2009). However, *affordances* have also been a central concept in human-computer interaction for some time (Gibson 1977; Norman 1988; Kaptelinin 2013).

According to Norman (1988), "*the term affordance refers to the perceived and actual properties of the thing, primarily those fundamental properties that determine just how the thing could possibly be used.*" Norman later clarified that he actually only wanted to cover *perceived affordances* (Norman 2004) which are all ways of grasping an object that are obvious to the user. Nevertheless, his definition of (actual) affordances is also precise and useful.

Combining the views of HCI and robotics, I define the *(actual) grasp affordances* of an object as all effective ways of grasping this object. Depending on the shape of the object, only a small number of grasp types will allow for holding it safely. The primary function of the object further limits the number of different useful grasps. For example, a screwdriver affords only a small number of significantly different grasps, such as a power grasp and a precision grasp. When grasping a computer mouse, most users will want to place their index finger on a mouse button. These requirements strongly decrease the number of applicable grasp types.

Furthermore, I define *perceived grasp affordances* as all ways of grasping the object that are obvious to the user, following Norman's characterization. Neither actual nor perceived grasp affordances are strict subsets of the other one: some effective ways of grasping an object may not be obvious to the user, whereas some seemingly obvious ways of grasping an object may not be effective. For example, it may take a few futile attempts for a toddler to realize that soap bubbles can not be effectively grasped in a force-closure grasp.

Designers of grasp interaction need to be aware of all grasp affordances the object offers. However, they can reduce the number of perceived grasp affordances, for instance by adding dents to object where fingers should be placed.

Creating perceived grasp affordances allows for strategic placement of grasp-sensitive areas on the object's surface. HandSense, presented in Chapter 5, utilizes this fact for achieving good grasp recognition with a small number of sensors.

As Eric Rademacher noted in his Master's Thesis (Rademacher 2010), touchscreens in mobile devices are placed at locations that are usually not covered by the hand holding the device. In contrast, grasp-sensitive surfaces should ideally be located at places on the object's surface that are covered by the hand in normal operation.

In practice, all of the other four GRASP aspects also affect grasp affordances and perceived grasp affordances. For example, a bottle on a shelf offers different grasp affordances than a bottle in a crate.

## 11.3 Goal

The *goal* aspect includes all factors that cause a grasp to be initiated. It describes what the user wants to achieve by grasping the object. Goals may be comprised of sub-goals.

Obviously, the user's goal has a major effect on the employed grasp. For example, a pencil is held in different ways for line-drawing and cross-hatching. As shown in Chapter 10, mobile phones are grasped differently depending on the task.

Within the set of all goals, several distinctions can be made:

- intrinsically versus extrinsically motivated grasps, i.e., grasps chosen by the user versus grasps defined by an experimenter,
- explicit versus implicit grasp interaction, i.e., whether the user knows that their grasp will evoke a reaction of the system,
- exerting versus supporting grasps, i.e., whether a grasp is intended for manipulating the object of for fixating it in order to allow interaction by the other hand.

These different facets of the *goal* aspect are discussed in the following. Table 11.1 shows examples for combinations of explicit/implicit grasp interaction and exerting/supporting grasps.

### 11.3.1 Extrinsically and Intrinsically Selected Grasps

It is helpful to distinguish between grasps employed due to an intrinsic decision and grasps employed due to an extrinsic requirement. In general, intrinsically selected grasps are described by their function whereas extrinsically selected grasps are defined by their form.

For example, a study participant might be told to pick up a box from the floor. The grasp they choose is intrinsically selected, i.e. can be arbitrarily selected from the set of functionally effective grasps.

In contrast, a study participant might be told to pick up the box using a left-handed precision grasp involving all fingers. In this case, the choice of grasp is extrinsically determined - the participant does not choose a functionally effective grasp but tries to match the formal definition of the grasp. When the user is explicitly asked to employ a certain grasp type, other GRASP factors obviously have less influence than if the user selects an appropriate grasp on their own.

Both types of grasp selection have been used in scientific research. Some researchers define the exact grasp type to be used (Kim et al. 2006; Wimmer and Boring 2009), whereas others require users to conduct certain tasks, letting them choose the most appropriate grasp themselves (Taylor and Bove 2009).

When designing grasp interaction it is helpful to be aware of this distinction.

We can not control intrinsically selected grasps. When given a choice, users will choose the grasp they see as the most appropriate one. As shown in Chapter 10, people use a variety of different grasps even for very simple tasks.

However, we can design grasp interaction to contain extrinsically selected grasps. This can be done by designing grasp affordances that require a certain grasp type to be employed. Alternatively, the user can be trained to choose only certain grasp types when interacting with a grasp-sensing artifact.

In practice, most grasps are intrinsically selected. In some cases extrinsically selected grasps may be warranted, however. For example, requiring a certain grasp type may increase the accuracy of biometric authentication; it may also be prudent to require a specific exotic grasp for triggering irreversible operations in order to avoid accidental activation.

### 11.3.2 Explicit and Implicit Grasp Interaction

It is important to realize that grasping can be used both as an explicit and an implicit input channel (Wimmer and Boring 2009; Wimmer 2010a; Wimmer 2011a).

*Implicit grasp interaction* means that the user chooses grasps based on the object's grasp affordances and does not intend to effect a certain reaction of the object (Wimmer 2010a). For example, a user might pick up their mobile phone in order to call someone. The phone would automatically switch on the display without the user necessarily intending or even noticing this.

*Explicit grasp interaction* means that the user knows the *meaning* that is associated with a certain grasp type and grasps the object accordingly to effect a certain reaction (Wimmer 2010a). For example, a user might hold their phone in landscape orientation because they know that this activates the camera application.

For implicit grasp interaction, the user does not need to know or take into account that an object is grasp-sensitive.

Schmidt (Schmidt 2000) defines implicit interaction as: "an action, performed by the user[,] that is not primarily aimed to interact with a computerized system but which such a system understands as input."

For grasp interaction, *implicit* means that the primary function of the grasp is not to evoke a reaction of the grasp-sensing system, but to hold or manipulate the object. While the user does not aim to evoke a reaction of the system, they may expect it, however.

For example, a police officer pulling her grasp-sensitive gun primarily intends to move the gun into a suitable position. She knows that the gun automatically authenticates her and authorizes her to fire it. However, authentication is not her primary motivation for grasping the gun.

When the user knows about the implicit grasp interaction, and it fails, they may try to explicitly effect the results of originally implicit grasp interaction. Implicit grasps may convey information about the user's intention. However, the intention is not necessarily aimed at interacting with the object. For example, a user might pick up their mobile phone in order to put it somewhere else. They might be surprised or even annoyed by the phone reacting to their grasp (Wimmer 2011a).

For explicit grasp interaction, the user needs to be aware that they handle a grasp-sensitive object. In this case, a grasp *always* conveys information about the user's goal. Furthermore, the user expects a system response. Therefore, it is essential for effective grasp interaction to determine whether a grasp is implicit or explicit. This can be done, for example, by choosing grasp types for explicit interaction that are uncommon in everyday usage.

### 11.3.3 Exerting and Supporting Grasps

A final distinction is between exerting and supporting grasps.[3]

[3] These two aspects were called "primary" and "supportive" grasps in the original publication (Wimmer 2011a).

An *exerting grasp* is a grasp that is executed with the intention of manipulating the grasped object (Wimmer 2011a). This includes picking up an object in order to put it somewhere else, grasping a computer mouse, or holding a spoon.

Exerting grasps are often employed when interacting with tangible user interfaces, pen-based user interfaces, or hand tools. As they are used for manipulating the object in a certain way, they can provide much information on the user's goals.

A *supporting grasp* serves to fixate or position an object in order to interact with it (Wimmer 2011a). This includes holding a mobile phone with one hand while typing a message with the other hand, clutching a bottle so that one can twist of the cap, or cradling a phone between neck and shoulders in order to speak into it. Holding a mobile phone while typing with the thumb of the same hand should also be seen as a supporting grasp, as the primary interaction could also be achieved if the phone lay on a table.

Supporting grasps are often employed in mobile interaction to allow input on a touch-screen with the other hand while on the move. As they are not directly involved in active manipulation of object or data, they provide little information about the primary goals of the user. However, supporting grasps often precede interaction, allowing the system to anticipate it. Furthermore, supporting grasps may provide information about the primary interaction. For example, recognizing with which hand the user holds a mobile phone allows the system to optimize the user interface for input with the other hand.

Supporting grasps usually can not be easily changed during interaction. Therefore, a change in a supporting grasp provides valuable information. For example, a user shifting their grasp while thumb-typing on a mobile phone might indicate that the user can not easily reach certain areas on the screen. As shown by Noor et al. (2014), slight changes in the supporting grasp may also be used for predicting touch input.

Supporting grasps also allow for haptic feedback (Fukumoto and Sugimura 2001).

Exerting or supporting grasps need not be limited to a single hand. For example, two-thumb typing uses both hands for supporting grasps. Bi-manual interaction with tangible artifacts - for instance on the Reactable - requires both hands to conduct exerting grasps.

One may also simultaneously conduct supporting and exerting grasps with a single hand, for instance by holding a bottle in a power grasp between palm and fingers 3-5 while twisting the cap with thumb and index finger using a precision grasp.

In summary, the user's goals have a major effect on the specific grasp used. While previous models, e.g., by (MacKenzie and Iberall 1994) describe the task or goal as a monolithic factor, motivations for human grasping are complex and diverse. Therefore, deconstructing the user's goal into extrinsic and intrinsic motivations, explicit and implicit interactions, and exerting and supporting grasps allows for a more nuanced discussion of grasping and possibly for more accurate predictions of grasps.

| | implicit interaction | explicit interaction |
|---|---|---|
| exerting grasp | pick up phone in order to put it on the table | hold phone horizontally in order to invoke photo mode |
| supporting grasp | hold phone in one hand while typing with the other hand | hold phone in one hand and pan a map with the index finger of the other hand. Squeeze phone with holding hand in order to zoom |

Table 11.1: Examples of exerting and supporting grasps in implicit and explicit grasp interaction.

## 11.4 Anatomy

The *anatomy* aspect includes all factors that are inherent to the grasping person's body (Wimmer 2011a).

These include:

- hand size and shape
- body posture
- sensorimotor control of grasping
- habits
- vision
- changes in these properties over time

For example, a small child will need to clutch a ball with both hands, whereas an adult may easily hold it in one hand.

In most cases, differences in *anatomy* can not be controlled by designers of grasp interaction. Therefore, applications need to be robust against user-specific differences in grasping. This may be achieved by training the system with a variety of users or by choosing wide thresholds. Overall, it is impossible to reliably distinguish between semantically different grasps (a low false acceptance rate) and at the same time recognize semantically similar grasps by different users (a low false rejection rate). As discussed in Section 3.5, a trade-off between both is required.

Sometimes, sensing such differences is the main objective. For instance, biometric authentication may be based on such differences between users. A power tool might detect that the user is not holding it tightly enough and warn her.

## 11.5 Setting

The *setting* aspect includes all factors pertaining to the environment in which the grasp takes place and to the physical relationship between user and object.

These include:

- extrinsic properties of the object, such as position, orientation, visibility, and wetness
- accessibility of the object and barriers obstructing access to it
- extrinsic properties of the grasping person, such as position and orientation
- environmental conditions, such as lighting and temperature

For example, pulling a bottle out of a crate requires a different grasp than taking it from a shelf. In this case, precision grasp (crate) and power grasp (shelf) are functionally equivalent - which one is used only depends on the setting, not on the task.

A well-designed system should not attribute these differences to different goals but recognize that they are caused by differences in setting. This is actually very hard. Designers of grasp interaction rarely have control over the setting for grasp interaction. For many graspable objects, e.g., for mobile phones, it is very difficult to anticipate in which settings users will grasp their device. Therefore, user studies in real-world settings are indispensable.

## 11.6 Relationship

The *relationship* aspect includes non-physical factors that are unique to a user-object combination. These are properties of the object that we can not perceive but know about or feel towards it.

This includes:

- feelings, e.g. disgust, fear, anger directed towards the object
- assumptions about an object's intrinsic properties, such as its weight
- assumptions about the worth or dangerousness of an object
- social values

For example, when picking up a used paper towel most people will try to have as little contact with it as possible and therefore choose a precision grasp. However, if they know that it is their own paper towel that they dropped earlier, they may also employ a power grasp.

Siegel, Walker, and Stefanucci (2009) found that partially covering a tool's handle with a disgusting substance led participants to use a different grasp for picking it up. As another example, Eastough and Edwards (2007) found that knowledge about the mass of an object already affects hand posture during the reaching phase - even if that knowledge is incorrect.

One might argue that these factors should belong either to the *anatomy* or the *properties*, or even the *goal* aspect. However, the relationship depends on both user and object. For example, an arachnophobic person will pick up a box labeled "Spiders!" in a different way than someone who loves spiders. The same arachnophobic person will pick up the same box differently if it is labeled "Beetles!". And this person will again be very hesitant to pick up a box labeled "Beetles!" if they know that actually spiders are inside.

All these factors which are partially determining a grasp can not easily be observed or controlled. Collecting such factors in a separate aspect allows the other aspects to stay independent of each other.

The *relationship* aspect has received little attention so far.

## 11.7 Applying GRASP to Practical Problems

In the original publication I have shown that the GRASP model meets formal requirements for a "good" model (Wimmer 2011a). However, formally valid models are not necessarily also useful in practice.

In the following I show how the GRASP model can applied as a tool for formative and summative evaluations of grasp-sensing user interfaces. Furthermore, GRASP allows for structuring current and future research.

**GRASP as a Tool in the Design Process** GRASP may be employed in formative evaluations, i.e., informing design decisions. There it may act as a framework for analyzing requirements and limitations and provide a checklist of issues to consider. Like other heuristics and checklists, GRASP does not replace knowledge and experience of a product designer or interaction designer but is another tool in the designer's toolbox. It also may help non-designers in understanding and systematically approaching the complexities in grasp interaction.

For example, an engineer might want to design a grasp-sensing rotary knob for a stationary music player device. Depending on the number of fingers used for grasping it, different parameters (volume, current track, bass boost) should be controlled by the rotary knob. GRASP aids the designer in considering important aspects of the design:

- **Goal**: The user will turn the rotary knob clockwise or counter-clockwise. It can be assumed that the user will only grasp the rotary knob when they want to change

parameters. As the device is stationary, users will probably not grasp the knob by accident or when holding the device. If the designer is unsure about this, they might want to conduct a small observational user study. Interaction will be intrinsically motivated, explicit, and exerting. Therefore, it is valid - and certainly helpful - to somehow indicate the recognizable grasp types on the rotary knob.
- **Relationship**: Rotary knobs are not known for evoking emotional responses. Therefore, this factor is probably not relevant for the current application.
- **Anatomy**: As users need to employ different numbers of fingers for turning the knob, the knob should be easy to turn with two fingers. As hand sizes of users vary significantly, care must be taken when sensing how many fingers are in contact with the knob. Wrist rotation is limited to an angle of 270 degree. It might be annoying for users to loosen and fasten an unnatural grasp while turning the knob. Therefore, the knob's transmission ratio should probably be set in a way that reduces re-gripping the knob. Furthermore, ergonomic rotation angles differ for left-handed and right-handed interaction Ideally, the designer should conduct a user study to find comfortable ratios and to find out whether re-gripping may cause users to grasp differently every time.
- **Setting**: As the device is stationary, mobile use does not need to be considered. However, users might want to operate the knob in the dark or without looking at it. Therefore, care should be taken not to mistake feeling for the knob for interacting with it. For example, a machine-learning classifier should be trained to reject grasp signatures effected by feeling for the knob.
- **Properties**: Obviously, the knob needs to afford grasps with a different number of fingers. This limits its minimum and maximum diameter. Furthermore, the surface needs to provide enough friction to allow turning the knob with two fingers. Given the aforementioned ergonomic limitations, adding grasp affordances - e.g., finger-wide dents along the rim - might guide users to grasp the knob in a way that allows turning it sufficiently far both in clockwise and counter-clockwise directions without having to re-grip.

There are certainly more aspects to consider in the design of grasp-sensing artifacts. And the issues described here would also be found by an experienced industrial designer or by trial-and-error. However, GRASP makes it harder to miss important issues and offers a common language for all involved parties.

**Analyzing Limitations of Existing Solutions** GRASP may also be employed in summative evaluations, i.e. analyzing existing applications.

The following example illustrates how GRASP can be used to describe existing applications and find problems that have not been accounted for[4]. GRASP allows for identifying which of the five meaningful factors have been controlled or measured in the study

[4] This and another example have been previously presented in my original publication (Wimmer 2011a).

- and which factors have not been considered. Furthermore, external validity of a study can be analyzed by comparing study setup and real-world applications for each of the five factors.

Veldhuis et al. (Veldhuis et al. 2004) propose using grasp signatures - in this instance: pressure patterns - for authenticating legit users of a gun. GRASP shows how different aspects of grasping have been handled by the researchers:

- **Goal**: The goal is given by the study setup. Grasp interaction is implicit and employing an exerting grasp. In the actual application, the user's goal would be to shoot the gun. However, in the studies, participants only grasped the guns and did not fire them.
- **Relationship**: The relationship between owner and weapon is ignored in this research. It is not clear whether the relationship has an effect on the employed grasp in practice.
- **Anatomy**: Anatomical features are recognized and used for authentication.
- **Setting**: The setting is controlled in the laboratory study. However, in real-life scenarios, setting (gun location, environmental conditions) will vary significantly.
- **Properties**: The properties of the gun butt are kept constant throughout the studies. Weight, weight distribution, and surface texture are different from an actual gun, however.

Comparing the implicit assumptions made by the researchers with real-life usage reveals three limitations of the study[5]:

The goals in the study and in real-life applications are not identical, limiting external validity of the study. Furthermore, users may have a number of different reasons for grasping the gun - e.g., they might want to clean it or put it away. The system should not treat such grasps as unauthorized attempts to fire the gun.

A gun may need to be fired in various different settings - e.g., while running or while it is raining. The system needs to correctly authenticate users in such conditions. Again, external validity of the controlled laboratory experiment is limited.

Finally, grasps may significantly differ between gun prototype and actual guns. Therefore, actual recognition rate may be better or worse than the reported values.

The issues identified here may be useful when planning subsequent projects, comparing different studies, or composing a survey of research on grasp interaction.

[5] Despite its limitations, the research by Veldhuis et al. represents one of the most thorough and extensive investigations of grasp interaction.

# 11.8 Discussion of GRASP

In this chapter I presented GRASP, a descriptive model of meaning in grasping. Its main purpose is to offer a common language and mental framework for discussing grasp interaction. The model is non-exclusive and constrained to meaningful human grasping.

A major open question for me is whether *Relationship* is really a factor of its own. As the relationship between user and object depends on both, issues like disgust could be attributed to either *Properties* or *Anatomy* instead of being put into a separate *Relationship* aspect. One might also argue that such issues are actually *Goals*. For example, disgust might be described as the goal of avoiding contact with an object.

However, this would conflict with the initial definition of *Goals*. Having *Relationship* as a separate aspect allows limiting the other four aspects to factors that can be defined or measured.

Another important question is whether affordances should be attributed to the *Properties*, *User*, or even the *Goals*. It seems reasonable to define affordances as properties of a graspable object - after all, creating affordances requires modifying the object. However, affordances are also dependent on the intended users: an ergonomic computer mouse provides affordances for right-handed users but not for left-handed users. Furthermore, designing affordances requires the designer to make assumptions about the users' goals.

Therefore, the relationship between grasp interaction and grasp affordances is not yet perfectly clear to me. For the sake of simplicity I have decided to discuss affordances as part of the *Properties* aspect.

According to Google Scholar, the GRASP model has been cited in eleven publications as of October 2014. While some researchers explicitly refer to my GRASP model (Wolf et al. 2012; Kyota and Saito 2012), it has not been scrutinized, confirmed, or refuted so far. Therefore, GRASP should be seen as a basis for further refinement and discussion instead of a fixed rule set.

# Chapter 12

# Grasp Interaction in Context

*Grasp interaction is an interaction technique of its own. However, it complements other paradigms, such as tangible interaction or gestural interaction. While tangible interaction focuses on physical objects being manipulated, grasp interaction is concerned with the hand manipulating them. Gestural interaction comprises two-dimensional and three-dimensional movement of body parts for explicitly signaling intentions, whereas grasp interaction requires little movement and also conveys implicit information. Both tangible interaction and grasp interaction can be augmented and supported by grasp sensing. In general, grasp information should not be treated as reliable input but as contextual information that may be combined with other information to better describe the context of an interaction. Effective grasp interaction may be facilitated by creating affordances, conducting laboratory and real-life studies, lowering latency, and combining grasp information with other context. As there is no intuitive and effective "undo" feature for grasp interaction, designers need to be careful not to let users accidentally trigger dangerous actions. It is tempting to throw machine-learning algorithms at grasp recognition. However, manually designing and refining heuristics gives the designer deeper insight into the design space and its challenges.*

**Attribution:** This chapter contains only original content. Some of the thoughts expressed in this chapter may have been mentioned in previous publications and have certainly been inspired by my previous work.

While this dissertation deals with *grasp-sensitive surfaces* - techniques for capturing grasp signatures - I have also discussed challenges for grasp interaction in the previous chapters. I have shown that correctly interpreting grasps is difficult and that users employ a wide variety of grasps for similar tasks. Furthermore, I have proposed a descriptive model of meaningful aspects affecting how a user grasps an object.

The relationship between grasp interaction and touch interaction has already been described in Section 4.4.2. However, grasp interaction is also related to many other areas of research, most notably *tangible interaction*, *gestural interaction*, *intrabody near-field communication*, and *context-sensitive applications*. These relationships are discussed in the following sections. I conclude this chapter with a list of (yet untested) design suggestions for grasp-sensing artifacts.

## 12.1 Relationship to Tangible User Interfaces

Despite the similarity of their names, grasp interaction and tangible interaction are not overlapping but complementary concepts. Grasp-sensitive surfaces can be used to enrich tangible interaction.

With Tangible User Interfaces (TUIs), digital data is given a physical representation. This physical representation is uni- or bidirectionally coupled to the underlying data. Users may access and manipulate this data through manual manipulation of tangible artifacts. In most cases, such manipulation consists of moving, rotating, or deforming the tangible artifacts.

The concept of TUIs was first mentioned in the seminal paper "Bricks: laying the foundations for graspable user interfaces" (Fitzmaurice, Ishii, and Buxton 1995). In this paper and Fitzmaurice's subsequent PhD thesis, grasping was viewed as a means for changing the spatial arrangement of physical objects; *how* the user grasped the object was not of interest. Consequently, the term "graspable user interfaces" was soon replaced with the broader "tangible user interfaces" in subsequent publications (Ishii and Ullmer 1997).

In contrast, grasp-sensitive user interfaces are not only graspable but also use the information provided by a grasp for interaction. Whereas tangible interaction focuses on the physical object being manipulated, grasp interaction focuses on the hand manipulating the object.

## 12.2 Relationship to Gestural User Interfaces

Gestural interaction comprises two-dimensional and three-dimensional movement of fingers, arms, and other body parts as indicators of the user's intentions.

As grasp interaction does not necessarily require a temporal change of the grasp during interaction (see Section 3.2), grasp interaction is clearly not a subset of gestural interaction. However, certain ways of grasp interaction - e.g., squeezing an object - involve changing the grasp over time.

While such changes in grasps can be seen as gestures, they are usually spatially constrained both in amplitude and degrees of freedom. Therefore, approaches from gesture recognition and gestural interaction can not easily be applied to grasp interaction.

However, grasp sensing can also be used to support gestural interaction. Katrin Wolf and colleagues have recently explored interaction with mobile devices through micro-gestures while grasping the devices (Wolf et al. 2012; Wolf 2012a; Wolf 2012b; Wolf et al. 2013). In these cases, grasping is a prerequisite for gestural interaction, not a part of it: Functionally effective grasps fixate an object while fingers not participating in the grasp conduct small gestures.

Everyday grasps and everyday gestures may be utilized for implicit interaction. In these cases, it is necessary to reliably distinguish between intentional and unintentional grasps or gestures. I would assume that findings from implicit gestural interaction can inform implicit grasp interaction and vice versa.

## 12.3 Relationship to Intrabody Near-Field Communication

Intrabody near-field communication (NFC) encompasses methods for transmitting digital information via the user's body, usually by capacitive coupling between sender, body, and receiver (Zimmerman 1996; Grosse-Puppendahl et al. 2014). This also allows for transmitting information from the user to an object that they touch or grasp and vice versa. Therefore, intrabody NFC may be employed as an alternative or supplement to grasp interaction:

Similar to grasp interaction, information is only transmitted when the user touches or grasp an object. However, the information transmitted via the user's hand is independent of the user's grasp. Information may also be transmitted when the user inadvertently touches the object.

While intrabody NFC requires the user to wear a transceiver circuit on their body - possibly limiting social acceptance - it can be a robust alternative to grasp sensing for applications where it is necessary to identify the user touching a device, even if they are not grasping it. Furthermore, both approaches may be combined to increase robustness. For example, a system might only enable an intrabody NFC circuit when a certain grasp is detected, avoiding a Midas touch effect. In addition, data transmitted via intrabody NFC may provide further information about the user's intentions while grasping an object.

## 12.4 Grasp information as context

As argued throughout this dissertation, grasp information alone does not reliably allow predicting a user's intentions. Several other aspects beside the user's goal influence how an object is grasped. Therefore, grasp signatures collected by grasp-sensitive surfaces should be seen as *contextual information* (or simply: *context*) that is combined with other contextual information to form a better view of the user's intentions and constraints.

Abowd et al. (1999) define *context* as "any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves."

In this sense, grasp information may be used as context for other input modalities, such as touch input, text input, or speech input.

Furthermore, grasp information should be combined with context from other sensors, such as accelerometers, user identification, or GUI state, to improve the reliability and expressivity of grasp interaction.

## 12.5 Suggestions for Designing Grasp-Sensitive User Interfaces

No dissertation in human-computer interaction is complete without a set of design guidelines. The following suggestions summarize subjective thoughts of mine on important considerations when designing grasp-sensing user interfaces. In addition and in contrast to the requirements defined by Tsukamoto, Yuta, and Okada (2014) (Section 4.4.19), my suggestions do not define necessary properties of grasp-sensitive artifacts but offer hints and guidance for designers of grasp interaction. As no large-scale real-life deployments of grasp-sensitive objects have happened or been evaluated yet, some of the suggestions may turn out to be wrong.

- Create affordances to guide grasps. This forces different users to grasp an object in the same way, mitigating the effects of anatomy or setting. The more pronounced these affordances are, the more likely users will grasp the object in the intended way. However, limiting the user to only one or a few comfortable grasps also reduces the expressivity of grasp interaction.
- If you know exactly how users will grasp an object - e.g., due to pronounced affordances - it is acceptable to put only a few sensors at pre-defined spots on the surface. If it is not clear how users will grasp - and this is almost always the case - it is necessary to make the whole surface grasp-sensitive.
- Try to learn more about how your users grasp by conducting laboratory studies and observing real-life usage.
- Combine grasp information with other contextual information in order to increase the reliability of grasp identification and interpretation.

- An *undo* feature for grasp interaction is not trivial. For explicit interaction, one might define a certain *undo* grasp. However, this would need to be communicated to users. For implicit interaction, the user usually does not even know that they just invoked a command. Therefore, potentially dangerous operations should never be activated just by picking up an object or inspecting it. For example, one might require a sequence of unusual grasps for triggering such operations. 'Inverse' affordances - physical features of the object that make it very hard to grasp it in a certain way - might also be used for mitigating the risk of a user inadvertently triggering dangerous operations.
- Speed up grasp sensing. Acceptable response times for grasp sensing certainly vary for different applications. While this topic has not yet been explored, I would assume that users expect nearly instant responses in many cases. This requires a high sampling rate and significantly limits the amount of pre-processing and classification.
- Try understanding why and how users grasp. Machine-learning classifiers are easy to use and allow for quite accurate grasp identification. However, they may give a false sense of reliability as they can fail spectacularly for grasp signatures that are not in the training set. Blindly throwing machine learning at problems may also prevent us from thinking hard enough about the underlying mechanisms, ultimately limiting scientific progress. Considering meaningful aspects of a grasp before implementing a classifier allows for more robust and extensible grasp sensing and may generate ideas for new applications and features. My GRASP model may be of use for systematically finding expressive or confusable grasps.

In summary, grasp interaction extends, augments, and sometimes replaces a variety of other interaction techniques. Rarely, grasp information on its own provides sufficient information for deducing the users' intentions. Therefore, grasp information should be combined with other contextual information.

Despite being related to touch interaction, tangible interaction, and gestural interaction, grasp interaction has its own set of properties and design criteria.

Chapter 13

# Summary, Discussion and Outlook

*Grasp sensing enables and enhances many applications, e.g., by identifying users, providing additional input modalities, predicting a user's actions or making hand-held tools more expressive. Research on the three main challenges in grasp sensing - capturing, identification, and interpretation of grasps - has progressed with different speeds. Grasp capturing has received much attention so far because it is the first step needed for every research project. Therefore, basic technical challenges have been solved. The major challenge now is to make prototyping and manufacturing grasp-sensitive surfaces cheaper and easier. Grasp identification has received some attention so far. However, partially due to a lack of robust grasp classifications, research is still fragmented. Robust, general classification of everyday grasps has neither been attempted nor accomplished. Grasp interpretation has received very little attention so far. This will only change once grasp identification gets more robust. Nevertheless, meanwhile it is reasonable to learn more about meaningful factors in everyday grasping. The present dissertation has a few limitations. For instance, the three prototyping techniques lack real-life evaluations and real-life impact so far. Future work should focus on grasp identification and grasp interpretation. For identifying grasps, corpora and models of everyday grasping need to be developed. For interpreting grasps, a corpus of meaning in grasps is necessary. Furthermore, we need to investigate practical, social, and ethical implications of grasp sensing before actually implementing them.*

**Attribution:** This chapter contains a summary of all previous chapters. It was written specifically for this dissertation.

In this final chapter of the dissertation, I summarize the previous chapters, discuss salient questions raised in the dissertation, and give an outlook on important open questions and future work.

## 13.1 Summary

This dissertation is divided in three parts.

In the first part I motivated the topic of the dissertation and gave an overview of human grasping and related work.

Sensing how a user grasps a grasp-sensitive artifact - often a hand-held tool - allows for supporting their explicit and implicit goals. Thus, grasp interaction holds the potential to make our interaction with the world more effective, efficient, safe, and enjoyable.

The human hand and its functions have inspired research in a multitude of areas - most importantly in anatomy, neuroscience, prosthetics, and robotics. Each of these research areas has developed own perspectives, definitions, and classifications of human grasping. A survey of relevant literature indicates that grasps vary significantly between users, for different tasks, and different settings. Research on grasping specifically for human-computer interaction is scarce. Therefore, I suggest definitions and approaches for *grasp interaction*. Most notably, my interaction-centric definition of human grasping focuses on the function of a grasp rather than its form or formation.

In order to facilitate grasp interaction, it is necessary to first sense grasps. Grasp sensing comprises three steps: information about a grasp needs to be captured, the grasp type needs to be identified, and the meaning of the grasp has to be interpreted. A comprehensive survey of research on grasp sensing techniques shows a wide variety of applications that benefit from grasp sensing. However, in most research projects previous insight from other research areas is ignored. In many cases, researchers pre-define the meaning of certain grasp types, thereby reducing external validity of their research. Due to a lack of prototyping techniques, most researchers spent significant effort on implementing grasp-sensitive surfaces.

In the second part of this dissertation I presented three novel prototyping techniques for grasp-sensitive surfaces.

In order to advance grasp interaction, we need to gain a better understanding of how and why humans grasp, and how other factors - e.g., object shape - affect grasps. Rapid, iterative prototyping of grasp-sensitive user interfaces allows researchers and designers to quickly implement and evaluate novel concepts.

To support research on grasp interaction, I developed three quite different prototyping techniques.

HandSense employs few, high-resolution, capacitive sensors for detecting touch and proximity. It builds on CapToolKit, a custom open-source capacitive sensing platform which I had developed previously. HandSense allows for versatile placement of sensors within objects that should be made grasp-sensitive and offers some unique features. For example, the HandSense prototype can reliably distinguish between left-handed and right-handed grasp by sensing the thickness of touching tissue instead of tracking individual finger positions.. With a user study I showed that heuristic classification can be as accurate as machine-learning classifiers when only a few common grasps need to be distinguished.

FlyEye employs bundles of optical fiber to capture touch locations on non-planar surfaces. To this end, one end of each fiber is inserted into a small hole drilled into the object's surface. The other fiber ends are bundled together and attached to an infrared camera. In order to make touch and proximity sensing more robust, a second set of fibers, interspersed with the 'camera' fibers, emits infrared light from the surface of the object. This light gets reflected back into the 'camera' fibers by a finger or hand in close proximity to or contact with the surface. In order to make it easier to map changes in the captured image to touch locations on the surface, I developed a semi-automatic *relative mapping* technique.

TDRtouch builds on time-domain reflectometry to implement a multi-touch-sensitive cable that may be wrapped around objects of arbitrary shape. Touching an unshielded cable changes its characteristic impedance at the touch location. A time-domain reflectometer can locate the position of these changes (discontinuities) by injecting an electric pulse into the cable and measuring the run-time of the echoes reflected back at each discontinuity. In our research we quantified the performance of TDR-based touch sensing, developed guidelines for designing sensor layouts, and extended the principle to non-planar, deformable, and modular sensors.

In comparison, HandSense, FlyEye, and TDRtouch exhibit complementary advantages and limitations. While HandSense offers only low spatial resolution, it is fast and versatile. FlyEye requires significant effort to cover large surfaces and requires modifications to the object. In turn, it offers high resolution and does not require knowledge of electronic circuits or soldering. TDRtouch - in its current implementation - requires large and expensive hardware. However, it is very versatile and offers reasonably high spatial resolution.

Together, these three prototyping techniques cover many use cases and have inspired subsequent own and other research.

In the third part, I discussed challenges for grasp interaction and presented a theoretical framework for grasp interaction.

How users grasp in real-life interaction scenarios has not yet been explored in sufficient detail. Most research on grasp interaction was conducted in laboratory settings and/or focused on clearly defined use cases and grasps. This may lead to the development of unintuitive, unergonomic, and unresponsive grasp-sensitive artifacts.

To demonstrate the variety of grasps used even for trivial tasks, we conducted a study where participants had to grasp different mobile phones for different tasks. In this study participants employed at least two different grasp types for the same task. Furthermore, a number of exotic grasps was observed that are very different from the grasps used by a majority of participants.

GRASP is a descriptive model of meaning in human grasping that may act as a framework for discussing grasp interaction. By grouping all factors affecting the outcome of a grasp process into five aspects - Goal, Relationship, Anatomy, Setting, and Properties - GRASP aids in analyzing applications for grasp interaction.

Grasp interaction is related to tangible interaction and gestural interaction. For most real-life applications, grasp information should be combined with other information as part of the interaction context.

## 13.2 Discussion

In the following I highlight and discuss salient issues documented in this dissertation.

### 13.2.1 Applications for Grasp Sensing

A wide variety of applications for grasp-sensitive user interfaces have been described in this dissertation. Important application areas for grasp sensing are explicit and implicit user authentication, explicit selection of operating modes, implicit generation of context information, and implicit support of other device interactions.

As shown by Veldhuis et al. (2004), R. Chen, She, et al. (2011a), and Iso et al. (2012), grasp sensing allows for identifying users with high accuracy based on how they grasp a gun, a steering wheel, or a mobile phone. This can be used for increasing safety and security when interacting with hand-held devices, and allows for automatically adjusting the interface to different users.

Grasp sensing also opens up additional input modalities. As shown in several publications (Kim et al. 2006; Wimmer and Boring 2009; Taylor and Bove 2009), grasping an object in a certain way may explicitly or implicitly be used for triggering actions. I described how grasp information can be used as context information in ubiquitous computing (Wimmer and Boring 2009).

Grasp sensing can also be used to predict probable user actions (Noor et al. 2014), increasing input speed. Furthermore, grasp sensing also makes tool use more expressive (Song et al. 2011). Overall, these examples offer compelling evidence that grasp sensing can enhance interaction with graspable devices.

Further application areas for grasp sensing are, for example, manipulation of virtual objects (Kry and Pai 2006a) and dynamic adaptation of user interfaces (Goel, Wobbrock, and Patel 2012; L.-P. Cheng et al. 2012a; Cheng et al. 2013).

### 13.2.2 State of Research

I have proposed a generic grasp sensing workflow consisting of three steps in Section 3.5. These three steps - capture, identify, and interpret - are also helpful for investigating the state of research on grasp sensing and grasp interaction.

Capturing grasps has received much attention in related work because most researchers had to design and build their own grasp-sensitive surfaces. However, research on grasp-sensitive surfaces may become less important and less prevalent in the future. The prototyping techniques I developed allow for quick implementation of prototypes and low-volume products. Furthermore, recent developments in the fields of flexible electronics and capacitive sensing solutions may allow for printing grasp-sensitive surfaces in arbitrary shapes in the near future. For many simple applications, few, cheap, individually connected sensor electrodes - similar to those used in HandSense - will be the best choice. Due to their high cost, non-planar multi-layer sensor matrices might only be commercially feasible for simple shapes and/or expensive products. Therefore, the versatile and cheap approach employed by TDRtouch might be worth further exploration and could scale from tiny to huge grasp-sensitive objects. It will certainly take more research and a few years until grasp-sensitive surfaces become commercially available. Nevertheless, I think that *capturing grasps* is on its way to being a solved problem for researchers and will become an *engineering* challenge.

Identifying grasps, i.e., deriving grasp type or finger placement from the captured grasp signature, has received quite some attention by researchers, too. However, reliably identifying arbitrary grasps in real-life scenarios is not yet possible. Research prototypes have only distinguished between ten and twenty pre-defined grasp types so far. As described throughout this dissertation, real grasping is much more diverse. One reason for the lack of progress in this area is certainly the lack of off-the-shelf hardware for capturing grasps, requiring researchers to invest effort and money into hardware design. A second reason is the lack of a common, robust framework for describing grasps in various degrees of abstraction, and the lack of statistical evidence regarding the variety and prevalence of different grasps in real life. As the description of previous studies of human grasping in Chapter 2 shows, different classifications and descriptions for the same specific grasp make it impossible to compare different approaches. Without reliable data about real-life grasping, research may have small external validity, for example if researchers choose grasps that are easy to recognize, instead of prevalent ones for their studies. Without a common language, research will remain insular. Overall, identifying grasp is work in progress where more research is necessary.

Interpreting grasps, i.e. reacting to a grasp that has been correctly recognized, is a challenge that has not really been approached so far. In many prototypes, different modes can be activated by grasping in a certain way. The focus of previous research is clearly on recognizing grasps, not on the actions or modes associated with them. These are often only attached to motivate the research on hardware and classification algorithms. None of the related work presented in this dissertation has investigated what users actually intend when grasping an object. I assume that interpretation of grasps will not be tackled in the near future. First, the challenges of capturing and identifying grasps need to be solved. Nevertheless, preliminary empirical research on meaningful factors in human grasping is both feasible and sensible in the near term.

The implications of the current state of research for future work is discussed in the next section.

### 13.2.3 Limitations of This Dissertation

Throughout this dissertation I have mentioned and discussed shortcomings and omissions in related work. Some of the criticism also applies to my own work.

For example, I was not aware of much of the related work when working on HandSense and FlyEye. Thus, the simple application scenario for HandSense reinforced the notion that only a small set of distinct grasps need to be distinguished.

While the accuracy of HandSense was evaluated in a user study, no user studies were conducted for FlyEye or TDRtouch. Therefore, actual utility in real-life scenarios has not been proven. For none of the prototyping techniques, their suitability for actual prototyping of user interfaces was evaluated in a user study.

Finally, while my papers have been cited several times and apparently have inspired the work of other researchers, neither prototyping techniques nor GRASP framework have been used for actual product development yet. Therefore, practical validity and utility of my research are low. This fate seems to be shared with the majority of HCI research, however.

## 13.3 Future Work

As discussed in the previous section, grasp sensing and grasp interaction are still far from being relevant in everyday life. In the following I describe future work on grasp-sensitive surfaces, identification of grasps, and interpretation of grasps.

### 13.3.1 Grasp-Sensitive Surfaces

Regarding grasp-sensitive surfaces, it is necessary to reduce price and increase flexibility of sensor substrates. Except for a few special applications, optical grasp-sensitive surfaces - as presented with FlyEye - might not provide significant benefits over capacitive sensors.

Single capacitive sensors strategically arranged at places on the surface which offer grasp affordances - as presented with HandSense - offer cheap and simple but limited grasp sensing. While several sensing toolkits support multiple high-resolution touch sensors (CapToolKit, OpenCapSense, Kitronyx Snowboard), further research into software support for grasp sensing is needed. It would also be interesting to evaluate whether swept-spectrum capacitive sensing (Sato, Poupyrev, and Harrison 2012) offers a real advantage over traditional capacitive sensing.

As described in detail in Chapter 7, TDRtouch offers many opportunities for making the technique smaller, cheaper, and more robust. A cheap, small, fast, and precise TDR circuit would significantly simplify implementation of grasp-sensitive objects.

Furthermore, it would be interesting to evaluate how well each sensing technique supports ideation and rapid prototyping of grasp-sensitive applications.

### 13.3.2 Grasp Identification

In order to improve identification of different grasps, three issues need to be addressed.

First, we need to understand better how people grasp in everyday life, so that we can build better corpora and taxonomies of typical and exotic grasps. A thorough investigation of everyday grasping will also highlight shortcomings and gaps of existing classifications. Having a large corpus of grasps, their typical occurrences, and associated grasp signatures would also help in training machine-learning classifiers.

Second, a multi-level descriptive model of grasps is necessary for discussing grasp sensing and mapping grasp signatures to grasps in a consistent, reusable manner. Such a model would allow for describing grasps by high-level features (e.g., *power grasp*), low-level features (e.g., angles of joints, touch locations), and intermediate features (e.g., virtual fingers). It would allow interaction designers to specify high-level grasps that can be directly mapped to a set of low-level features for classification algorithms. By including typical standard deviations for all low-level features, the model would also allow deriving similarity metrics for comparing different grasps. The research presented in this dissertation, especially the GRASP model (Chapter 11)

Third, analyzing time-dependent changes in grasp signatures instead of identifying only static grasps may improve accuracy and reduce latency of grasp identification.

### 13.3.3 Grasp Interpretation

Little research has been conducted so far on correctly interpreting a user's grasp. Without having solved capturing and identification of grasps, concrete suggestions for future research on grasp interpretation may be premature. Nevertheless, I see two overall challenges in the near future.

Similar to a corpus of everyday grasps, we also need to develop a corpus of meaning in grasps. While it is certainly not possible to create general direct mappings between grasps and users' intentions, knowing which intentions are typically associated with certain grasps limits the potential for misinterpretations.

Furthermore, we need to investigate practical, social, and ethical implications of grasp-sensitive objects before actually implementing them. Many usability challenges that are well known to GUI designers also apply to grasp interaction - e.g., discoverability of actions, undo, or operating modes. Novel grasp-sensitive artifacts might confuse users or lead to awkward interactions. Pervasive grasp sensing might also evoke a Midas-touch feeling in users who want to just pick up objects without causing some additional action. Finally, grasp sensing might be employed for surveillance or to limit the freedom of users; for example, a manufacturer's digital rights management (DRM) might only allow registered users to operate a power drill.

Discussing these issues before we actually build grasp-sensitive environments seems like a good idea.

---

The ability to grasp tools has allowed humans to replace the slow progress of genetic evolution with the rapid advances of culture and engineering. Tools have made the human hand faster, more precise, and more powerful. As Schlesinger noted at the beginning of his seminal treatment of human grasping, human hand and hand-held tool merge to one whole, allowing humans to shape the world around them. It is hoped that the research presented in this dissertation helps making the world a little bit better in some ways, and not worse in any way.

# IV

# Bibliography

Abowd, Gregory D, Anind K Dey, Peter J Brown, Nigel Davies, Mark Smith, and Pete Steggles. 1999. "Towards a Better Understanding of Context and Context-Awareness." In *Handheld and Ubiquitous Computing*, 304–7. Springer.

acam messelectronic GmbH. 2000. "Application Note 1: Laser Distance Measurement with TDC's." `http://www.acam.de/fileadmin/Download/pdf/English/AN001_e.pdf`.

AEA Technology Inc. 2013. "Application Note 201: Step Vs. Pulse TDR Technology." `http://www.aeatechnology.com/uploads/application_notes/file_93965c103c_AN201%20Step%20vs%20Pulse%20TDR%20Technology.pdf`.

Agashe, H.A., and J.L. Contreras-Vidal. 2011. "Reconstructing Hand Kinematics During Reach to Grasp Movements from Electroencephalographic Signals." In *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE*, 5444–47. IEEE.

Anderson, Glen, Rina Doherty, and Subhashini Ganapathy. 2011. "User Perception of Touch Screen Latency." In *Design, User Experience, and Usability. Theory, Methods, Tools and Practice*, 195–202. Springer.

Ansuini, Caterina, Livia Giosa, Luca Turella, Gianmarco Altoè, and Umberto Castiello. 2008. "An Object for an Action, the Same Object for Other Actions: Effects on Hand Shaping." *Experimental Brain Research* 185 (1). Springer: 111–19.

Arbib, M.A., T. Iberall, and D.M. Lyons. 1985. "Coordinated Control Programs for Movements of the Hand." In *Hand Function and the Neocortex*, edited by A.W. Goodwin and I. Darian-Smith. Berlin: Springer-Verlag.

Atmel Corporation. 2000. *Touch Sensors Design Guide*. 10620D-AT42-04/09.

Barrett, Gary, and Ryomei Omote. 2010. "Projected-Capacitive Touch Technology." *Information Display* 26 (3): 16–21.

Bartindale, Tom, Daniel Jackson, and Patrick Olivier. 2009. *FiberSense – Multi-Touch Fibre Optic Sensing on Non-Planar Surfaces*. Technical Report, Department of Computing Science, Newcastle University.

Bartling, Jim. 2009. "Low-Cost, High-Resolution Time-Measurement Application." *ECN Magazine*. Rockaway, NJ, USA: Advantage Media. `http://www.ecnmag.com/articles/2009/07/low-cost-high-resolution-time-measurement-application`.

Baudisch, Patrick, Mike Sinclair, and Andrew Wilson. 2007. "Soap: a Pointing and Gaming Device for the Living Room and Anywhere Else." In *ACM SIGGRAPH 2007 Emerging Technologies*. SIGGRAPH '07. New York, NY, USA: ACM. doi:10.1145/1278280.1278298.

Baur, Dominikus, Otmar Hilliges, and Andreas Butz. 2008. "Flux: Enhancing Photo Organization Through Interaction and Automation." In *Proceedings of the 9th International Symposium on Smart Graphics*, 216–23. SG '08. Berlin, Heidelberg: Springer-Verlag. doi:10.1007/978-3-540-85412-8_20.

Baxter, L.K. 1997. *Capacitive Sensors: Design and Applications*. IEEE Press.

Baxter, Larry K. 2000. *Capacitive Sensors*. `http://capsense.com/capsense-wp.pdf`.

Beck, F., and B. Stumpe. 1973. *Two Devices for Operator Interaction in the Central Control of the New CERN Accelerator*. CERN Technical Report 73-6.

Bell, Charles. 1833. "The Hand, Its Mechanisms and Vital Endowments, as Evincing Design." W. Pickering.

Benko, Hrvoje, Andrew D. Wilson, and Ravin Balakrishnan. 2008. "Sphere: Multi-Touch Interactions on a Spherical Display." In *Proceedings of the 21st Annual ACM Symposium on User Interface Software and Technology*, 77–86. UIST '08. New York, NY, USA: ACM. doi:10.1145/1449715.1449729.

Berlin, Eugen, Jun Liu, Kristof van Laerhoven, and Bernt Schiele. 2010. "Coming to Grips with the Objects We Grasp: Detecting Interactions with Efficient Wrist-Worn Sensors." In *TEI '10: Proceedings of the Fourth International Conference on Tangible, Embedded, and Embodied Interaction*, 57–64. New York, NY, USA: ACM. doi:http://doi.acm.org/10.1145/1709886.1709898.

Bicchi, A. 1995. "On the Closure Properties of Robotic Grasping." *The International Journal of Robotics Research* 14 (4). Sage Publications: 319–34.

Böhme, Christoph. 2011. "The Selective Attention for Action Model (SAAM)." PhD thesis, University of Birmingham.

Bonnet, Robert. 1915. *Die Hand Und Ihr Ersatz. 26. Vaterländischer Und Kriegsvortrag Gehalten Am 10. Und 20. März 1915*. Leopold Voss.

Buhan, I. R., J. M. Doumen, P. H. Hartel, and R. N. J. Veldhuis. 2006. *Feeling Is Believing: a Location Limited Channel Based on Grip Pattern Biometrics and Cryptanalysis*. Technical Report TR-CTIT-06-29. Enschede: Centre for Telematics; Information Technology University of Twente; http://eprints.eemcs.utwente.nl/5694/; Centre for Telematics; Information Technology University of Twente.

Bullock, Ian M., Joshua Z. Zheng, Sara De La Rosa, Charlotte Guertler, and Aaron M. Dollar. 2013. "Grasp Frequency and Usage in Daily Household and Machine Shop Tasks." *IEEE Transactions on Haptics* 99 (PrePrints). Los Alamitos, CA, USA: IEEE Computer Society: 1. doi:http://doi.ieeecomputersociety.org/10.1109/TOH.2013.6.

Butler, Alex, Shahram Izadi, and Steve Hodges. 2008. "SideSight: Multi-'Touch' Interaction Around Small Devices." In *Proceedings of UIST '08*, 201–4. New York, NY, USA: ACM. doi:http://doi.acm.org/10.1145/1449715.1449746.

Buxton, Bill. 2007. "Multi-Touch Systems That I Have Known and Loved." `http://www.billbuxton.com/multitouchOverview.html`.

Chang, Chih-Chung, and Chih-Jen Lin. 2011. "LIBSVM: A Library for Support Vector Machines." *ACM Trans. Intell. Syst. Technol.* 2 (3). New York, NY, USA: ACM: 27:1–27:27. doi:10.1145/1961189.1961199.

Chen, Ronghua, M. She, Jin Wang, Xiangping Sun, and Lingxue Kong. 2011a. "Driver Verification Based on Handgrip Recognition on Steering Wheel." In *Systems, Man, and Cybernetics (SMC), 2011 IEEE International Conference on*, 1645–51. doi:10.1109/ICSMC.2011.6083907.

Chen, Ronghua, Mary Fenghua She, Xiangping Sun, Lingxue Kong, and Yuchuan Wu. 2011b. "Driver Recognition Based on Dynamic Handgrip Pattern on Steeling Wheel." *2010 11th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing* 0. Los Alamitos, CA, USA: IEEE Computer Society: 107–12. doi:http://doi.ieeecomputersociety.org/10.1109/SNPD.2011.33.

Cheng, Lung-Pan, Fang-I Hsiao, Yen-Ting Liu, and Mike Y. Chen. 2012a. "iRotate grasp: Automatic Screen Rotation Based on Grasp of Mobile Devices." In *Adjunct Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology*, 15–16. UIST Adjunct Proceedings '12. New York, NY, USA: ACM. doi:10.1145/2380296.2380305.

———. 2012b. "iRotate: Automatic Screen Rotation Based on Face Orientation." In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2203–10. CHI '12. New York, NY, USA: ACM. doi:10.1145/2207676.2208374.

Cheng, Lung-Pan, Hsiang-Sheng Liang, Che-Yang Wu, and Mike Y. Chen. 2013. "IGrasp: Grasp-Based Adaptive Keyboard for Mobile Devices." In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 3037–46. CHI '13. New York, NY, USA: ACM. doi:10.1145/2470654.2481422.

Clarke, Brooke. 2003. "Tektronix 1502 Metallic Time Domain Reflectometer." `http://www.prc68.com/I/Tek1502.shtml`.

Collins, Ryan V. 2003. "Methods and Apparatus for Receiving User Input via Time Domain Reflectometry." Google Patents.

Cremer, Max. 1907. "Ueber Die Registrierung Mechanischer Vorgänge Auf Electrischem Wege, Speziell Mit Hilfe Des Saitengalvanometers Und Saitenelectrometers." *Münchner Medizinische Wochenschrift* 54 (August): 1629–30.

Cutkosky, M., and P. Wright. 1986. "Modeling Manufacturing Grips and Correlations with the Design of Robotic Hands." In *Robotics and Automation. Proceedings. 1986 IEEE International Conference on*, 3:1533–39. IEEE.

Cutkosky, M.R. 1989. "On grasp choice, grasp models, and the design of hands for manufacturing tasks." *IEEE Transactions on Robotics and Automation* 5 (3): 269–79.

Deakin, RE, and DG Kildea. 1999. "A Note on Standard Deviation and RMS." *Australian Surveyor* 44 (1). Taylor & Francis: 74–79.

Detry, Renaud, Emre Baseski, Mila Popovic, Younes Touati, N Kruger, Oliver Kroemer, Jan Peters, and Justus Piater. 2009. "Learning Object-Specific Grasp Affordance Densities." In *Development and Learning, 2009. ICDL 2009. IEEE 8th International Conference on*, 1–7. IEEE.

Dreyfuss, Henry. 1966. "The Measure of Man - Human Factors in Design."

Eastough, Daniel, and Martin G Edwards. 2007. "Movement Kinematics in Prehension Are Affected by Grasping Objects of Different Mass." *Experimental Brain Research* 176 (1). Springer: 193–98.

Echtler, Florian, Tobias Sielhorst, Manuel Huber, and Gudrun Klinker. 2009. "A Short Guide to Modulated Light." In *TEI '09: Proceedings of the 3rd International Conference on Tangible and Embedded Interaction*, 393–96. New York, NY, USA: ACM. doi:10.1145/1517664.1517742.

Ekvall, Staffan, and Danica Kragic. 2005. "Grasp Recognition for Programming by Demonstration." In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, 748–53. IEEE.

Evett, Steven R. 2000. *Chapter 8: Tektronix 1502 Modification for Waveform Output*. USDA Time Domain Reflectometry (TDR) System Manual. `http://www.cprl.ars.usda.gov/TDR/TDR_S8.PDF`.

Feix, Thomas. 2011. "Anthropomorphic Hand Optimization Based on a Latent Space Analysis." PhD thesis, Technische Universität Wien.

Feix, Thomas, and Daniel Jaworski. 2009. *Data Analysis of Test Measurement Video Analysis (Unpublished)*. Technische Universität Wien.

Feix, Thomas, Roland Pawlik, Heinz-Bodo Schmiedmayer, Javier Romero, and Danica Kragic. 2009a. *The Generation of a Comprehensive Grasp Taxonomy*. TechNote. `http://www.csc.kth.se/grasp/taxonomyGRASP.pdf`.

———. 2009b. "A Comprehensive Grasp Taxonomy." In *Robotics, Science and Systems Conference: Workshop on Understanding the Human Hand for Advancing Robotic Manipulation, Poster Presentation*.

Fifer, M.S., M. Mollazadeh, S. Acharya, NV Thakor, and NE Crone. 2011. "Asynchronous Decoding of Grasp Aperture from Human ECoG During a Reach-to-Grasp Task." In *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE*, 4584–87. IEEE.

Fishkin, K.P., M. Philipose, and A. Rea. 2005. "Hands-on RFID: Wireless Wearables for Detecting Use of Objects." In *Wearable Computers, 2005. Proceedings. Ninth IEEE International Symposium on*, 38–41. IEEE.

Fitzmaurice, George W., Hiroshi Ishii, and William A. S. Buxton. 1995. "Bricks: Laying the Foundations for Graspable User Interfaces." In *CHI '95: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 442–49. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co. doi:10.1145/223904.223964.

Fruchterman, Thomas M. J., and Edward M. Reingold. 1991. "Graph Drawing by Force-Directed Placement." *Software: Practice and Experience* 21 (11). Department of Computer Science, University of Illinois at Urbana-Champaign, 1304 W. Springfield Avenue, Urbana, IL 61801-2987, U.S.A.: John Wiley & Sons, Inc.: 1129–64. doi:10.1002/spe.4380211102.

Fukumoto, Masaaki, and Yasuhito Suenaga. 1994. "FingeRing: a Full-Time Wearable Interface." In *Conference Companion on Human Factors in Computing Systems*, 81–82. CHI '94. New York, NY, USA: ACM. doi:10.1145/259963.260056.

Fukumoto, Masaaki, and Toshiaki Sugimura. 2001. "Active Click: Tactile Feedback for Touch Panels." In *CHI '01 Extended Abstracts on Human Factors in Computing Systems*, 121–22. CHI EA '01. New York, NY, USA: ACM. doi:10.1145/634067.634141.

Gibson, James J. 1977. "The Theory of Affordances." In *Perceiving, Acting, and Knowing : Toward an Ecological Psychology*, edited by Robert Shaw and John Bransford. Hillsdale, NJ, USA: Lawrence Erlbaum Associates.

Goel, Mayank, Jacob Wobbrock, and Shwetak Patel. 2012. "GripSense: Using Built-in Sensors to Detect Hand Posture and Pressure on Commodity Mobile Phones." In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology*, 545–54. UIST '12. New York, NY, USA: ACM. doi:10.1145/2380116.2380184.

Gong, Nan-Wei, Jürgen Steimle, Simon Olberding, Steve Hodges, Nicholas Edward Gillian, Yoshihiro Kawahara, and Joseph A. Paradiso. 2014. "PrintSense: A Versatile Sensing Technique to Support Multimodal Flexible Surface Interaction." In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 1407–10. CHI '14. New York, NY, USA: ACM. doi:10.1145/2556288.2557173.

Große-Puppendahl, Tobias, Yannick Berghöfer, Andreas Braun, Raphael Wimmer, and Arjan Kuijper. 2013. "OpenCapSense: A Rapid Prototyping Toolkit for Pervasive Interaction Using Capacitive Sensing." In *IEEE International Conference on Pervasive Computing and Communications (PerCom)*, 151–58. `http://epub.uni-regensburg.de/27975/`.

Grosse-Puppendahl, Tobias, Sebastian Herber, Raphael Wimmer, Frank Englert, Sebastian Beck, Julian von Wilmsdorff, Reiner Wichert, and Arjan Kuijper. 2014. "Capacitive Near-Field Communication for Ubiquitous Interaction and Perception." In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, 231–42. UbiComp '14 Adjunct. New York, NY, USA: ACM. doi:10.1145/2632048.2632053.

Han, Dong-Ho, Bao-Shu Xu, Myoung J Choi, Jiangqi He, Scott Gardiner, and Cliff Lee. 2005. "Realization of Ultra-Wideband, High-Resolution TDR for Chip-Carrier Packages." In *ASME 2005 Pacific Rim Technical Conference and Exhibition on Integration and Packaging of MEMS, NEMS, and Electronic Systems Collocated with the ASME 2005 Heat Transfer Summer Conference*, 1499–1503. American Society of Mechanical Engineers.

Han, Jefferson Y. 2005. "Low-Cost Multi-Touch Sensing Through Frustrated Total Internal Reflection." In *UIST '05: Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology*, 115–18. New York, NY, USA: ACM Press. doi:http://doi.acm.org/10.1145/1095034.1095054.

Harrison, Beverly L., Kenneth P. Fishkin, Anuj Gujar, Carlos Mochon, and Roy Want. 1998. “Squeeze Me, Hold Me, Tilt Me! An Exploration of Manipulative User Interfaces.” In *Proceedings of CHI ’98*, 17–24. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co. doi:http://doi.acm.org/10.1145/274644.274647.

Heumer, G., H.B. Amor, M. Weber, and B. Jung. 2007. “Grasp recognition with uncalibrated data gloves-a comparison of classification methods.” *VR ’07*. IEEE.

Hilbert, David. 1891. “Ueber Die Stetige Abbildung Einer Line Auf Ein Flächenstück.” *Mathematische Annalen* 38 (3). Springer: 459–60.

Hinckley, K., J. Pierce, M. Sinclair, and E. Horvitz. 2000. “Sensing techniques for mobile interaction.” In *Proceedings of UIST ’00*, 91–100. ACM New York, NY, USA.

Hinckley, Ken, and Mike Sinclair. 1999. “Touch-Sensing Input Devices.” In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: the CHI Is the Limit*, 223–30. CHI ’99. New York, NY, USA: ACM. doi:10.1145/302979.303045.

Hinckley, Ken, Michel Pahud, Hrvoje Benko, Pourang Irani, François Guimbretière, Marcel Gavriliu, Xiang ’Anthony’ Chen, Fabrice Matulic, William Buxton, and Andrew Wilson. 2014. “Sensing Techniques for Tablet+Stylus Interaction.” In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*, 605–14. UIST ’14. New York, NY, USA: ACM. doi:10.1145/2642918.2647379.

Holman, David, and Roel Vertegaal. 2011. “TactileTape: Low-Cost Touch Sensing on Curved Surfaces.” In *Proceedings of the 24th Annual ACM Symposium Adjunct on User Interface Software and Technology*, 17–18. UIST ’11 Adjunct. New York, NY, USA: ACM. doi:10.1145/2046396.2046406.

Holman, David, Nicholas Fellion, and Roel Vertegaal. 2014. “Sensing Touch Using Resistive Graphs.” In *Proceedings of the 2014 Conference on Designing Interactive Systems*, 195–98. DIS ’14. New York, NY, USA: ACM. doi:10.1145/2598510.2598552.

Holz, Christian, and Patrick Baudisch. 2011. “Understanding Touch.” In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2501–10. CHI ’11. New York, NY, USA: ACM. doi:10.1145/1978942.1979308.

Holz, Christian, Tovi Grossman, George Fitzmaurice, and Anne Agur. 2012. “Implanted User Interfaces.” In *Proceedings of the 2012 ACM Annual Conference on Human Factors in Computing Systems*, 503–12. CHI ’12. New York, NY, USA: ACM. doi:10.1145/2207676.2207745.

Hotelling, Steve Porter, and Wayne Carl Westerman. 2012. “Flexible Touch Sensing Circuits.” Patent. `http://www.patentlens.net/patentlens/patent/US_8144129/en/`.

Howarth, Barry, Mark Coates, and Lee Renforth. 2006. “Fault Location Techniques for One of the World’s Longest AC Interconnector Cables.” In *Proceedings of the 8th IEE International Conference on AC and DC Power Transmission*, 14–18. ACDC ’06. London, UK: IET.

Hrabia, Christopher-Eyk, Katrin Wolf, and Mathias Wilhelm. 2013. "Whole Hand Modeling Using 8 Wearable Sensors: Biomechanics for Hand Pose Prediction." In *Proceedings of the 4th Augmented Human International Conference*, 21–28. AH '13. New York, NY, USA: ACM. doi:10.1145/2459236.2459241.

Huang, Chi-Fang, Yu-Ching Hung, and Chia-Lin Liu. 2009. "P-53: Precise Location of Touch Panel by Employing the Time-Domain Reflectometry." *SID Symposium Digest of Technical Papers* 40 (1). SID: 1291–93. doi:10.1889/1.3256532.

Hudson, Scott E., and Jennifer Mankoff. 2006. "Rapid Construction of Functioning Physical Interfaces from Cardboard, Thumbtacks, Tin Foil and Masking Tape." In *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology*, 289–98. UIST '06. New York, NY, USA: ACM. doi:10.1145/1166253.1166299.

Hudson, Scott E., Chris Harrison, Beverly L. Harrison, and Anthony LaMarca. 2010. "Whack Gestures: Inexact and Inattentive Interaction with Mobile Devices." In *Proceedings of the Fourth International Conference on Tangible, Embedded, and Embodied Interaction*, 109–12. TEI '10. New York, NY, USA: ACM. doi:10.1145/1709886.1709906.

Imrhan, Sheik N. 2000. "Comparison of Hand Dimensions Among Various Nationalities." *Human Factors and Ergonomics Society Annual Meeting Proceedings* 44 (33): 284–86.

Ingram, James N, Konrad P Körding, Ian S Howard, and Daniel M Wolpert. 2008. "The Statistics of Natural Hand Movements." *Experimental Brain Research* 188 (2). Springer: 223–36.

Ishii, Hiroshi, and Brygg Ullmer. 1997. "Tangible Bits: Towards Seamless Interfaces Between People, Bits and Atoms." In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, 234–41. CHI '97. New York, NY, USA: ACM. doi:10.1145/258549.258715.

Iso, Toshiki, Tsutomu Horikoshi, Masakatsu Tsukamoto, and Takeshi Higuchi. 2012. "Personal Feature Extraction via Grip Force Sensors Mounted on a Mobile Phone: Authenticating the User During Key-Operation." In *Proceedings of the 11th International Conference on Mobile and Ubiquitous Multimedia*, 30:1–:4. MUM '12. New York, NY, USA: ACM. doi:10.1145/2406367.2406404.

Jackson, Daniel, Tom Bartindale, and Patrick Olivier. 2009. "FiberBoard: Compact Multi-Touch Display Using Channeled Light." In *ITS*, 25–28.

Jackson, Edwin Ward. 1865. *Jackson's Gymnastics for the Fingers and Wrist*. London, UK: N. Trübner and Co. `https://archive.org/details/jacksonsgymnast00jackgoog`.

Jacobson-Sollerman, C, and L Sperling. 1977. "Grip Function of the Healthy Hand in a Standardized Hand Function Test. a Study of the Rancho Los Amigos Test." *Scandinavian Journal of Rehabilitation Medicine* 9 (3): 123–29.

Jeannerod, M. 1986. "The Formation of Finger Grip During Prehension. a Cortically Mediated Visuomotor Pattern." *Behavioural Brain Research* 19 (2). Elsevier: 99–116.

Jones, L.A., and S.J. Lederman. 2006. *Human Hand Function*. Oxford University Press, USA.

Jota, Ricardo, Albert Ng, Paul Dietz, and Daniel Wigdor. 2013. "How Fast Is Fast Enough?: a Study of the Effects of Latency in Direct-Touch Pointing Tasks." In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2291–2300. ACM.

Ju, Peter, Leslie Pack Kaelbling, and Yoram Singer. 2000. "State-Based Classification of Finger Gestures from Electromyographic Signals." In *Proceedings of the Seventeenth International Conference on Machine Learning*, 439–46. ICML '00. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. http://dl.acm.org/citation.cfm?id=645529.657806.

Kamakura, Noriko, Michiko Matsuo, Harumi Ishii, Fumiko Mitsuboshi, and Yoriko Miura. 1980. "Patterns of Static Prehension in Normal Hands." *The American Journal of Occupational Therapy* 34 (7). The American Occupational Association, Inc.: 437–45.

Kamp, Cornelis van de, and Frank T. Zaal. 2007. "Prehension Is Really Reaching and Grasping." *Exp Brain Res*, May. Center for Human Movement Sciences, University of Groningen, P. O. Box 196, 9700, AD Groningen, The Netherlands, f.t.j.m.zaal@rug.nl. doi:10.1007/s00221-007-0968-2.

Kang, S.B., and K. Ikeuchi. 1994. "Grasp Recognition and Manipulative Motion Characterization from Human Hand Motion Sequences." In *Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on*, 1759–64. IEEE.

Kang, Sing Bing, and Katsushi Ikeuchi. 1991. *A Framework for Recognizing Grasps*. CMU-RI-TR-91-24. Pittsburgh, PA: CMU Robotics Institute.

Kaptelinin, Victor. 2013. "Affordances." In *The Encyclopedia of Human-Computer Interaction, 2nd Ed.*, edited by Mads Soegaard and Rikke Friis Dam. Aarhus, Denmark: The Interaction Design Foundation. https://www.interaction-design.org/encyclopedia/affordances_and_design.html.

Kauffman, J.A., A.M. Bazen, S.H. Gerez, and R.N.J. Veldhuis. 2003. "Grip-Pattern Recognition for Smart Guns." In *Proceedings of the 14th ProRISC Workshop, Veldhoven, the Netherlands, November 26-27, 2003*, 379–84. STW Technology Foundation. http://doc.utwente.nl/46699/.

Kerr, Duncan R., Steve Hotelling, and Brian Q. Huppi. 2006. "Hand Held Electronic Device with Multiple Touch Sensing Devices." Patent Application. http://www.patentlens.net/patentlens/patent/US_2006_0197750_A1/en/.

Kim, K. 2011. "Time Domain Reflectometer Touch Screen Sensor." Patent. http://www.google.com/patents/US8736582.

Kim, K., W. Chang, S. Cho, J. Shim, H. Lee, J. Park, Y. Lee, and S. Kim. 2006. "Hand Grip Pattern Recognition for Mobile User Interfaces." In *In Proc. AAAI '06*, 21:1789. 2.

Kinoshita, Hiroshi, Tomohiko Murase, and Takao Bandou. 1996. "Grip Posture and

Forces During Holding Cylindrical Objects with Circular Grips." *Ergonomics* 39 (9). Taylor & Francis: 1163–76.

Kozik, A.F., and D.M. Taylor. 1992. "Apparatus to Determine Coordinates." Patent. `http://www.google.com/patents/US5149918`.

Kroemer, K.H. 1986. "Coupling the Hand with the Handle: An Improved Notation of Touch, Grip, and Grasp." *Human Factors*. Human Factors & Ergonomics Society.

Kry, Paul G., and D. K. Pai. 2006a. "Grasp recognition and manipulation with the tango." In *International Symposium on Experimental Robotics*. Vol. 10. 9. Springer.

Kry, Paul G., and Dinesh K. Pai. 2006b. "Interaction Capture and Synthesis." In *ACM SIGGRAPH 2006 Papers*, 872–80. SIGGRAPH '06. New York, NY, USA: ACM. doi:10.1145/1179352.1141969.

Kuo, Ye-Sheng, Sonal Verma, Thomas Schmid, and Prabal Dutta. 2010. "Hijacking Power and Bandwidth from the Mobile Phone's Audio Interface." In *Proceedings of the First ACM Symposium on Computing for Development*, 24:1–24:10. ACM DEV '10. New York, NY, USA: ACM. doi:10.1145/1926180.1926210.

Kyota, Fumihito, and Suguru Saito. 2012. "Fast Grasp Synthesis for Various Shaped Objects." *Computer Graphics Forum* 31 (2pt4). Blackwell Publishing Ltd: 765–74. doi:10.1111/j.1467-8659.2012.03035.x.

Landsmeer, JMF. 1962. "Power Grip and Precision Handling." *Annals of the Rheumatic Diseases* 21 (2). BMJ Publishing Group Ltd; European League Against Rheumatism: 164–70.

Lee, Donghwan, Jinho Sung, and Jaehong Park. 2003. "A 16ps-Resolution Random Equivalent Sampling Circuit for TDR Utilizing a Vernier Time Delay Generation." In *Nuclear Science Symposium Conference Record, 2003 IEEE*, 2:1219–23. IEEE.

Lee, Hyunjeong, Wook Chang, Joonah Park, and Junghyun Shim. 2009. "New Mobile UI with Hand-Grip Recognition." In *Proceedings of the 27th International Conference Extended Abstracts on Human Factors in Computing Systems*, 3521–22. CHI EA '09. New York, NY, USA: ACM. doi:10.1145/1520340.1520522.

Lee, SK, William Buxton, and KC Smith. 1985. "A Multi-Touch Three Dimensional Touch-Sensitive Tablet." In *ACM SIGCHI Bulletin*, 16:21–25. 4. ACM.

Liu, Shenwei, and François Guimbretière. 2012. "FlexAura: a Flexible Near-Surface Range Sensor." In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology*, 327–30. UIST '12. New York, NY, USA: ACM. doi:10.1145/2380116.2380158.

Lobo, Jorge, Pedro Trindade, and Jorge Dias. 2011. "Observing Hand Grasp Type and Contact Points Using Hand Distributed Accelerometers and Instrumented Objects." In *Procedings of the IEEE/ICRA 2011 Workshop on Autonomous Grasping*. Shanghai, China.

Mackay, Wendy E., and Anne-Laure Fayard. 1997. "HCI, Natural Science and Design: A Framework for Triangulation Across Disciplines." In *Proceedings of the 2Nd Conference on Designing Interactive Systems: Processes, Practices, Methods, and Techniques*, 223–34. DIS '97. New York, NY, USA: ACM. doi:10.1145/263552.263612.

MacKenzie, C.L., and T. Iberall. 1994. *The grasping hand*. North Holland.

Malicki, MA, R Plagge, M Renger, and RT Walczak. 1992. "Application of Time-Domain Reflectometry (TDR) Soil Moisture Miniprobe for the Determination of Unsaturated Soil Water Characteristics from Undisturbed Soil Cores." *Irrigation Science* 13 (2). Springer: 65–72.

Mäntyjärvi, J., K. Nybergh, J. Himberg, and K. Hjelt. 2004. "Touch Detection System for Mobile Terminals." In *Proceedings of MobileHCI '05*. Springer.

Marshall, J., T. Pridmore, M. Pound, S. Benford, and B. Koleva. 2008. "Pressing the Flesh: Sensing Multiple Touch and Finger Pressure on Arbitrary Surfaces." *Pervasive Computing*. Springer, 38–55.

Mascaro, S., and H.H. Asada. 2000. "Fingernail Touch Sensors: Spatially Distributed Measurement and Hemodynamic Modeling." In *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, 4:3422–27. IEEE.

———. 2001. "Finger Posture and Shear Force Measurement Using Fingernail Sensors: Initial Experimentation." In *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, 2:1857–62. IEEE.

Mascaro, S., K.W. Chang, and H.H. Asada. 1999. "Photo-Plethysmograph Nail Sensors: for Measuring Finger Forces Without Haptic Obstruction: Modeling and Experimentation." In *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, 2:962–67. IEEE.

Mascaro, S.A., and H.H. Asada. 2004. "Measurement of finger posture and three-axis fingertip touch force using fingernail sensors." *IEEE Transactions on Robotics and Automation* 20 (1). New York, NY: Institute of Electrical; Electronics Engineers, c1989-c2004.: 26–35.

Mason, Matthew T, Alberto Rodriguez, Siddhartha S Srinivasa, and Andrés S Vazquez. 2012. "Autonomous Manipulation with a General-Purpose Simple Hand." *The International Journal of Robotics Research* 31 (5). SAGE Publications: 688–703.

McPherron, S.P., Z. Alemseged, C.W. Marean, J.G. Wynn, D. Reed, D. Geraads, R. Bobe, and H.A. Béarat. 2010. "Evidence for Stone-Tool-Assisted Consumption of Animal Tissues Before 3.39 Million Years Ago at Dikika, Ethiopia." *Nature* 466 (7308). Nature Publishing Group: 857–60.

Mifflin, Houghton. 2011. *The American Heritage Dictionary of the English Language, Fifth Edition*. Houghton Mifflin.

Miller, Robert B. 1968. "Response Time in Man-Computer Conversational Transactions." In *Proceedings of the December 9-11, 1968, Fall Joint Computer Conference, Part I*, 267–77. ACM.

Mistry, P., and P. Maes. 2009. "SixthSense: a Wearable Gestural Interface." In *ACM SIGGRAPH ASIA 2009 Sketches*, 11. ACM.

Miyaki, Takashi, and Jun Rekimoto. 2009. "GraspZoom: Zooming and Scrolling Control Model for Single-Handed Mobile Interaction." In *Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services*, 11:1–:4. MobileHCI '09. New York, NY, USA: ACM. doi:10.1145/1613858.1613872.

Mohr LLC. 2009. "Press Release: U.S. Navy Selects Mohr CT100 TDR to Replace Tektronix(R) 1502C." `http://www.mohr-engineering.com/pr120209a.php`.

———. 2012. "Application Note: TDR for Microwave/RF and Digital Cables." `http://www.mohr-engineering.com/TDR_for_Microwave_RF_and_Digital_Cables-A.php`.

Mortensen, Ditte Hvas, and Klaus B. Bærentsen. 2013. "Playing with Nonverbal Communication: Using Grasp and Facial Direction to Create Adaptive Interaction in a Game." *Interacting with Computers*. doi:10.1093/iwc/iwt023.

Munroe, Randall. 2011. "XKCD #927: Standards." http://xkcd.com/927/.

Nakajima, Kosuke, Yuichi Itoh, Takayuki Tsukitani, Kazuyuki Fujita, Kazuki Takashima, Yoshifumi Kitamura, and Fumio Kishino. 2011. "FuSA Touch Display: a Furry and Scalable Multi-Touch Display." In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, 35–44. ITS '11. New York, NY, USA: ACM. doi:10.1145/2076354.2076361.

Napier, JR. 1956. "The prehensile movements of the human hand." *Journal of Bone & Joint Surgery, British Volume* 38 (4). JBJS (Br): 902.

Ng, Albert, Julian Lepinski, Daniel Wigdor, Steven Sanders, and Paul Dietz. 2012. "Designing for Low-Latency Direct-Touch Input." In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology*, 453–64. UIST '12. New York, NY, USA: ACM. doi:10.1145/2380116.2380174.

Nierop, O.A. van, A. van der Helm, K.J. Overbeeke, and T.J.P. Djajadiningrat. 2008. "A Natural Human Hand Model." *The Visual Computer* 24 (1). Springer: 31–44.

Nogami, Glenda Y. 1982. "Good-Fast-Cheap: Pick Any Two: Dilemmas About the Value of Applicable Research." *Journal of Applied Social Psychology* 12 (5). Wiley Online Library: 343–48.

Noor, Mohd, Mohammad Faizuddin, Andrew Ramsay, Stephen Hughes, Simon Rogers, John Williamson, and Roderick Murray-Smith. 2014. "28 Frames Later: Predicting Screen Touches from Back-of-Device Grip Changes." In *Proceedings of the 32Nd Annual ACM Conference on Human Factors in Computing Systems*, 2005–8. CHI '14. New York, NY, USA: ACM. doi:10.1145/2556288.2557148.

Norman, D. A. 1988. *The Psychology of Everyday Things*. Basic books.

Norman, Donald. 2004. "Affordances and Design." http://www.jnd.org/dn.mss/affordances_and.html.

Oikonomidis, Iason, Nikolaos Kyriazis, and Antonis Argyros. 2011. "Efficient Model-Based 3D Tracking of Hand Articulations Using Kinect." In *Proceedings of the British Machine Vision Conference*, 101.1–101.11. BMVA Press.

Oikonomidis, Iasonas, Nikolaos Kyriazis, and Antonis A. Argyros. 2011. "Full DOF Tracking of a Hand Interacting with an Object by Modeling Occlusions and Physical Constraints." In *ICCV*, edited by Dimitris N. Metaxas, Long Quan, Alberto Sanfeliu, and Luc J. Van Gool, 2088–95. IEEE.

Olberding, Simon, Nan-Wei Gong, John Tiab, Joseph A. Paradiso, and Jürgen Steimle. 2013. "A Cuttable Multi-Touch Sensor." In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*, 245–54. UIST '13. New York, NY, USA: ACM. doi:10.1145/2501988.2502048.

Oliver, B.M. 1964. "Time Domain Reflectometry." *HP Journal* 15 (6): 1–8. http://www.hpl.hp.com/hpjournal/pdfs/IssuePDFs/1964-02.pdf.

Pai, Dinesh K., Eric W. Vanderloo, Subarna Sadhukhan, and Paul G. Kry. 2005. "The Tango: A Tangible Tangoreceptive Whole-Hand Human Interface." *World Haptics Conference* 0. Los Alamitos, CA, USA: IEEE Computer Society: 141–47. doi:10.1109/WHC.2005.136.

Polhemus. 2006. "AC & Pulsed DC Motion Trackers." http://www.polhemus.com/polhemus_editor/assets/AC-DCwebPaper1.pdf.

Polya, George. 1957. *How to Solve It: a New Aspect of Mathematical Method, Ed*. London, UK: Penguin.

Ponce, J., S. Sullivan, A. Sudsang, J.D. Boissonnat, and J.P. Merlet. 1997. "On computing four-finger equilibrium and force-closure grasps of polyhedral objects." *International Journal of Robotics Research* 16 (1): 11–46.

Rademacher, Eric. 2010. "DigitSense: Grasp Recognition Using Fingerprint Sensors." Master's thesis, University of Munich.

Reddy, N.P., and V. Gupta. 2007. "Toward Direct Biocontrol Using Surface EMG Signals: Control of Finger and Wrist Joint Models." *Medical Engineering & Physics* 29 (3). Elsevier: 398–403.

Rekimoto, J. 2001. "GestureWrist and GesturePad: unobtrusive wearable interaction devices." *Proceedings Fifth International Symposium on Wearable Computers*. IEEE Comput. Soc, 21–27. doi:10.1109/ISWC.2001.962092.

Rogers, John A, Takao Someya, and Yonggang Huang. 2010. "Materials and Mechanics for Stretchable Electronics." *Science* 327 (5973). American Association for the Advancement of Science: 1603–7.

Roudaut, A., H. Pohl, and P. Baudisch. 2011. "Touch Input on Curved Surfaces." In *Proceedings of the 2011 Annual Conference on Human Factors in Computing Systems*, 1011–20. ACM.

Rudeck, Frederik, and Patrick Baudisch. 2012. "Rock-Paper-Fibers: Bringing Physical Affordance to Mobile Touch Devices." In *CHI*, 1929–32.

Santello, Marco, Martha Flanders, and John F. Soechting. 1998. "Postural Hand Synergies for Tool Use." *The Journal of Neuroscience*.

Saponas, T.S., C. Harrison, and H. Benko. 2011. "PocketTouch: Through-Fabric Capacitive Touch Input." In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, 303–8. ACM.

Saponas, T.S., D.S. Tan, D. Morris, and R. Balakrishnan. 2008. "Demonstrating the feasibility of using forearm electromyography for muscle-computer interfaces." *Proceeding of the Twenty-Sixth Annual SIGCHI Conference on Human Factors in Computing Systems*. ACM.

Saponas, T.S., D.S. Tan, D. Morris, R. Balakrishnan, J. Turner, and J.A. Landay. 2009. "Enabling always-available input with muscle-computer interfaces." *Proceedings of the 22nd Annual ACM Symposium on User Interface Software and Technology*. ACM.

Saponas, T.S., D.S. Tan, D. Morris, J. Turner, and J.A. Landay. 2010. "Making muscle-computer interfaces more practical." *Proceedings of the 28th International Conference on Human Factors in Computing Systems*. ACM.

Sato, Munehiko, Ivan Poupyrev, and Chris Harrison. 2012. "Touché: Enhancing Touch Interaction on Humans, Screens, Liquids, and Everyday Objects." In *Proceedings of the 2012 ACM Annual Conference on Human Factors in Computing Systems*, 483–92. CHI '12. New York, NY, USA: ACM. doi:10.1145/2207676.2207743.

Savage, Valkyrie, Xiaohan Zhang, and Björn Hartmann. 2012. "Midas: Fabricating Custom Capacitive Touch Sensors to Prototype Interactive Objects." In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology*, 579–88. UIST '12. New York, NY, USA: ACM. doi:10.1145/2380116.2380189.

Schlesinger, G. 1919. "Der Mechanische Aufbau der Künstlichen Glieder." In *Ersatzglieder und Arbeitshilfen*, edited by Borchardt, M. and Hartmann, K. and Leymann, H. nd Radike, R. and Schlesinger, G. and Schwiening, H. Berlin: Springer.

Schmidt, A. 2000. "Implicit Human Computer Interaction Through Context." *Personal and Ubiquitous Computing* 4 (2). Springer: 191–99.

Schöning, Johannes, Peter Brandl, Florian Daiber, Florian Echtler, Otmar Hilliges, Jonathan Hook, Markus Löchtefeld, et al. 2008. *Multi-Touch Surfaces: A Technical Guide*. Technical report. `http://drehscheibe.in.tum.de/forschung/pub/reports/2008/TUM-I0833.pdf.gz`.

Shang, X., and R. N. J. Veldhuis. 2007. "Local Absolute Binary Patterns as Image Preprocessing for Grip-Pattern Recognition in Smart Gun." In *IEEE Conference on Biometrics: Theory, Applications and Systems, Washington DC*, edited by K. W. Bowyer, P. J. Flynn, V. Govindaraju, and N. Ratha, 1–6. Washington DC: http://eprints.eemcs.utwente.nl/11334/; University of Notre Dame.

———. 2008. "Grip-Pattern Verification for Smart Gun Based on Maximum-Pairwise Comparison and Mean-Template Comparison." In *Biometrics: Theory, Applications and BTAS 2008. 2nd IEEE International Conference on Biometrics: Theory Applications and Systems, 2008., Washington, USA*, 1–5. Los Alamitos: http://eprints.eemcs.utwente.nl/14613/; IEEE Computer Society.

Shang, X., K. Kooi, R. N. J. Veldhuis, and W. P. T. Ganzevoort. 2006. "Restoration of Missing Lines in Grip Patterns for Biometrics Authentication on a Smart Gun." In *Proceedings of SPS-DARTS 2006 the Second Annual IEEE Benelux/DSP Valley Signal Processing Symposium, Antwerp, Belgium*, 95–98. Antwerp: http://eprints.eemcs.utwente.nl/11341/; IEEE Benelux Signal Processing Chapter.

Shang, Xiaoxin. 2008. "Grip-Pattern Recognition: Applied to a Smart Gun." PhD thesis, Enschede: University of Twente. `http://doc.utwente.nl/60347/`.

Shih, RH, EM Vasarhelyi, A. Dubrowski, and H. Carnahan. 2001. "The Effects of Latex Gloves on the Kinetics of Grasping." *International Journal of Industrial Ergonomics* 28 (5). Elsevier: 265–73.

Shrirao, Nikhil, Narender Reddy, and Durga Kosuri. 2009. "Neural Network Committees for Finger Joint Angle Estimation from Surface EMG Signals." *BioMedical Engineering OnLine* 8 (1): 2. doi:10.1186/1475-925X-8-2.

Siegel, Erika, Phillip Walker, and Jeanine Stefanucci. 2009. "So Close and yet so Far Away: An Effect of Disgust on Distance Perception and Graspability." *Journal of Vision* 9 (8). Association for Research in Vision; Ophthalmology: 1096–96.

Skierucha, Wojciech, Andrzej Wilczek, Agnieszka Szypłowska, Cezary Sławiński, and Krzysztof Lamorski. 2012. "A TDR-Based Soil Moisture Monitoring System with Simultaneous Measurement of Soil Temperature and Electrical Conductivity." *Sensors* 12 (10). Molecular Diversity Preservation International: 13545–66.

Slocum, Donald B, and Donald R Pratt. 1946. "Disability Evaluation for the Hand." *The Journal of Bone & Joint Surgery* 28 (3). The Journal of Bone; Joint Surgery: 491–95.

Smeets, J.B.J., and E. Brenner. 1999. "A New View on Grasping." *Motor Control* 3 (3). Human Kinetics: 237–71.

Smith, Joshua. 1999. "Electric Field Imaging." PhD thesis, Massachusetts Institute of Technology. `http://web.media.mit.edu/~jrs/phd.pdf`.

Smith, Joshua, Tom White, and Dodge. 1998. "Electric Field Sensing For Graphical Interfaces." *Computer Graphics and Applications* 18 (3): 54–61. `http://www.media.mit`

.edu/physics/publications/papers/98.02.CGA/_Final.pdf.

Smith, Paul, Cynthia Furse, and Jacob Gunther. 2005. "Analysis of Spread Spectrum Time Domain Reflectometry for Wire Fault Location." *Sensors Journal, IEEE* 5 (6). IEEE: 1469–78.

Smith, R.J., F. Tenore, D. Huberdeau, R. Etienne-Cummings, and N.V. Thakor. 2008. "Continuous Decoding of Finger Position from Surface EMG Signals for the Control of Powered Prostheses." In *Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE*, 197–200. IEEE.

Smith, Steven W, and others. 1997. "The Scientist and Engineer's Guide to Digital Signal Processing." California Technical Pub. San Diego. http://www.dspguide.com.

Song, Hyunyoung, Hrvoje Benko, Francois Guimbretiere, Shahram Izadi, Xiang Cao, and Ken Hinckley. 2011. "Grips and Gestures on a Multi-Touch Pen." In *Proceedings of the 2011 Annual Conference on Human Factors in Computing Systems*, 1323–32. CHI '11. New York, NY, USA: ACM. doi:10.1145/1978942.1979138.

Sperling, L, and C Jacobson-Sollerman. 1977. "The Grip Pattern of the Healthy Hand During Eating." *Scandinavian Journal of Rehabilitation Medicine* 9 (3): 115–21.

Starecki, Tomasz, and Stefan Misiaszek. 2006. "Low Cost Programmable Pulse Generator with Very Short Rise/fall Time." In *Proc. SPIE*, 6347:63472J–2J–5. doi:10.1117/12.714752.

Steenbergen, B, J Van der Kamp, AW Smithsman, and RG Carson. 1997. "Spoon Handling in Two-to-Four-Year-Old Children." *Ecological Psychology* 9 (2). Taylor & Francis: 113–29.

Sun, Shishuang, David J Pommerenke, James L Drewniak, Genda Chen, Liang Xue, Michael A Brower, and Marina Y Koledintseva. 2009. "A Novel TDR-Based Coaxial Cable Sensor for Crack/strain Sensing in Reinforced Concrete Structures." *Instrumentation and Measurement, IEEE Transactions on* 58 (8). IEEE: 2714–25.

Sun, Y., J.M. Hollerbach, and S.A. Mascaro. 2006a. "Measuring Fingertip Forces by Imaging the Fingernail." In *Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2006 14th Symposium on*, 125–31. IEEE.

———. 2006b. "Dynamic Features and Prediction Model for Imaging the Fingernail to Measure Fingertip Forces." In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, 2813–18. IEEE.

———. 2007. "Finger Force Direction Recognition by Principal Component Analysis of Fingernail Coloration Pattern." In *EuroHaptics Conference, 2007 and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. World Haptics 2007. Second Joint*, 90–95. IEEE.

———. 2008. "Predicting Fingertip Forces by Imaging Coloration Changes in the Fingernail and Surrounding Skin." *Biomedical Engineering, IEEE Transactions on* 55 (10).

IEEE: 2363–71.

———. 2009. “Estimation of Fingertip Force Direction with Computer Vision.” *Robotics, IEEE Transactions on* 25 (6). IEEE: 1356–69.

Taylor, Brandon. 2008. “Graspables: Grasp Recognition as a User Interface.” Master’s thesis, Massachusetts Institute of Technology.

Taylor, Brandon T., and Michael V. Bove. 2009. “Graspables: Grasp-Recognition as a User Interface.” In *In Proc. CHI ’09*, 917–26. New York, NY, USA: ACM. doi:10.1145/1518701.1518842.

Taylor, Craig L, and Robert J Schwarz. 1955. “The Anatomy and Mechanics of the Human Hand.” *Artificial Limbs* 2 (2). Hemisphere: 22–35.

Tektronix. 1986. “1502 Time Domain Reflectometer Service Manual, REV JAN 1986.”

Tsukamoto, Masakatsu, Higuchi Yuta, and Takashi Okada. 2014. “User Interface Using Natural Gripping Features - Grip UI.” *NTT DOCOMO Technical Journal* 15 (3): 17–24.

Veldhuis, Raymond N. J., Asker M. Bazen, Joost A. Kauffman, and Pieter H. Hartel. 2004. “Biometric Verification Based on Grip-Pattern Recognition.” In *Security, Steganography, and Watermarking of Multimedia Contents*, edited by Edward J. Delp and Ping W. Wong, 5306:634–41. Proceedings of SPIE. SPIE.

Wang, Robert Y., and Jovan Popović. 2009. “Real-time hand-tracking with a color glove.” *ACM Transactions on Graphics* 28 (3).

Wikipedia. 2013. “Plastic Optical Fiber — Wikipedia, the Free Encyclopedia.” `http://en.wikipedia.org/w/index.php?title=Plastic_optical_fiber&oldid=560667620`.

Willis, Karl, Eric Brockmeyer, Scott Hudson, and Ivan Poupyrev. 2012. “Printed Optics: 3D Printing of Embedded Optical Elements for Interactive Devices.” In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology*, 589–98. UIST ’12. New York, NY, USA: ACM. doi:10.1145/2380116.2380190.

Wimmer, Raphael. 2010a. “FlyEye: Grasp-Sensitive Surfaces Using Optical Fiber.” In *In Proc. TEI ’10*.

———. 2010b. “Some Thoughts on a Model of Touch-Sensitive Surfaces.” In *ACM International Conference on Interactive Tabletops and Surfaces*, 293–94. ITS ’10. New York, NY, USA: ACM. doi:10.1145/1936652.1936722.

———. 2011a. “Grasp Sensing for Human-Computer Interaction.” In *In Proc. TEI ’11*.

———. 2011b. “Grasp Interaction Using Physiological Sensor Data.” In *Workshop Brain and Body Interfaces at CHI 2011*.

———. 2011c. “Capacitive Sensors for Whole Body Interaction.” In *In: David England (Ed.): Whole Body Interaction. Springer, Berlin, Germany, 2011. ISBN 978-0-85729-432-6*.

Wimmer, Raphael, and Patrick Baudisch. 2011. "Modular and Deformable Touch-Sensitive Surfaces Based on Time Domain Reflectometry." In *UIST*. New York, NY, USA: ACM.

Wimmer, Raphael, and Sebastian Boring. 2009. "HandSense - Discriminating Different Ways of Grasping and Holding a Tangible User Interface." In *In Proc. TEI '09*.

Wimmer, Raphael, Fabian Hennecke, Florian Schulz, Sebastian Boring, Andreas Butz, and Heinrich Hußmann. 2010. "Curve: Revisiting the Digital Desk." In *NordiCHI 2010: Proceedings of the 6th Nordic Conference on Human-Computer Interaction*. New York, NY, USA: ACM.

Wimmer, Raphael, Matthias Kranz, Sebastian Boring, and Albrecht Schmidt. 2007. "A Capacitive Sensing Toolkit for Pervasive Activity Detection and Recognition." In *Per-Com '07*.

Wolf, Katrin. 2012a. "What I Grasp Is What I Control: Interacting Through Grasp Releases." In *Proceedings of the Sixth International Conference on Tangible, Embedded and Embodied Interaction*, 389–90. TEI '12. New York, NY, USA: ACM. doi:10.1145/2148131.2148228.

———. 2012b. "Design Space for Finger Gestures with Hand-Held Tablets." In *Proceedings of the 14th ACM International Conference on Multimodal Interaction*, 325–28. ICMI '12. New York, NY, USA: ACM. doi:10.1145/2388676.2388748.

Wolf, Katrin, Christian Müller-Tomfelde, Kelvin Cheng, and Ina Wechsung. 2012. "PinchPad: Performance of Touch-Based Gestures While Grasping Devices." In *Proceedings of the Sixth International Conference on Tangible, Embedded and Embodied Interaction*, 103–10. TEI '12. New York, NY, USA: ACM. doi:10.1145/2148131.2148155.

Wolf, Katrin, Robert Schleicher, Sven Kratz, and Michael Rohs. 2013. "Tickle: A Surface-Independent Interaction Technique for Grasp Interfaces." In *Proceedings of the 7th International Conference on Tangible, Embedded and Embodied Interaction*, 185–92. TEI '13. New York, NY, USA: ACM. doi:10.1145/2460625.2460654.

Worzyk, Thomas. 2009. *Submarine Power Cables: Design, Installation, Repair, Environmental Aspects*. Springer.

Zimmerman, TG. 1996. "Personal Area Networks: Near-field intrabody communication." *IBM Systems Journal* 35 (3/4): 610.

Zimmerman, Thomas G., Jaron Lanier, Chuck Blanchard, Steve Bryson, and Young Harvill. 1987. "A Hand Gesture Interface Device." In *Proceedings of the SIGCHI/GI Conference on Human Factors in Computing Systems and Graphics Interface*, 189–92. CHI '87. New York, NY, USA: ACM. doi:10.1145/29933.275628.

Zimmerman, Thomas G., Joshua R. Smith, Joseph A. Paradiso, David Allport, and Neil Gershenfeld. 1995. "Applying Electric Field Sensing to Human-Computer Interfaces." In *CHI '95*, 280–87.

# Erklärung zur Urheberschaft

Hiermit erkläre ich an Eides statt, dass die vorliegende Dissertation von mir selbständig und ohne unerlaubte Beihilfe angefertigt wurde.

München, den 19. Dezember 2014

Raphael Wimmer