LUDWIG-
MAXIMILIANS-
UNIVERSITÄT
MÜNCHEN

Bringing the Physical to the Digital: A New Model for Tabletop Interaction

Otmar Hilliges

München 2009

Bringing the Physical to the Digital:
A New Model for Tabletop Interaction

Otmar Hilliges

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

vorgelegt von
Otmar Hilliges
aus München

München, den 19. Januar 2009

Erstgutachter:	Prof. Dr. Andreas Butz
Zweitgutachter:	Prof. Dr. Sheelagh Carpendale
Drittgutachter:	Dr. Shahram Izadi
Viertgutachter:	Dr. Andrew D. Wilson

Tag der mündlichen Prüfung: 22. Juli 2009

Abstract

This dissertation describes an exploration of digital tabletop interaction styles, with the ultimate goal of informing the design of a new model for tabletop interaction. In the context of this thesis the term digital tabletop refers to an emerging class of devices that afford many novel ways of interaction with the digital. Allowing users to directly touch information presented on large, horizontal displays. Being a relatively young field, many developments are in flux; hardware and software change at a fast pace and many interesting alternative approaches are available at the same time. In our research we are especially interested in systems that are capable of sensing multiple contacts (e.g., fingers) and richer information such as the outline of whole hands or other physical objects. New sensor hardware enable new ways to interact with the digital. When embarking into the research for this thesis, the question which interaction styles could be appropriate for this new class of devices was a open question, with many equally promising answers.

Many everyday activities rely on our hands ability to skillfully control and manipulate physical objects. We seek to open up different possibilities to exploit our manual dexterity and provide users with richer interaction possibilities. This could be achieved through the use of physical objects as input mediators or through virtual interfaces that behave in a more realistic fashion.

In order to gain a better understanding of the underlying design space we choose an approach organized into two phases. First, two different prototypes, each representing a specific interaction style – namely gesture-based interaction and tangible interaction – have been implemented. The *flexibility* of use afforded by the interface and the level of *physicality* afforded by the interface elements are introduced as criteria for evaluation. Each approaches' suitability to support the highly dynamic and often unstructured interactions typical for digital tabletops is analyzed based on these criteria.

In a second stage the learnings from these initial explorations are applied to inform the design of a novel model for digital tabletop interaction. This model is based on the combination of rich multi-touch sensing and a three dimensional environment enriched by a gaming physics simulation. The proposed approach enables users to interact with the virtual through richer quantities such as collision and friction. Enabling a variety of fine-grained interactions using multiple fingers, whole hands and physical objects.

Our model makes digital tabletop interaction even more "natural". However, because the interaction – the sensed input and the displayed output – is still bound to the surface, there is a fundamental limitation in manipulating objects using the third dimension. To address this issue, we present a technique that allows users to – conceptually – pick objects off the surface and control their position in 3D. Our goal has been to define a technique that completes our model for on-surface interaction and allows for "as-direct-as possible" interactions. We also present two hardware prototypes capable of sensing the users' interactions beyond the table's surface. Finally, we present visual feedback mechanisms to give the users the sense that they are actually lifting the objects off the surface.

This thesis contributes on various levels. We present several novel prototypes that we built and evaluated. We use these prototypes to systematically explore the design space of digital tabletop interaction. The flexibility of use afforded by the interaction style is introduced as criterion alongside the user interface elements' physicality. Each approaches' suitability to support the highly dynamic and often unstructured interactions typical for digital tabletops are analyzed. We present a new model for tabletop interaction that increases the fidelity of interaction possible in such settings. Finally, we extend this model so to enable as direct as possible interactions with 3D data, interacting from above the table's surface.

Zusammenfassung

Das Thema dieser Dissertation ist die Erforschung von Interaktionsstilen für digitale Tabletop-Computer. Das ultimative Ziel ist ein neues Interaktionsmodel für die Tabletopinteraktion. Im Rahmen dieser Arbeit steht der Begriff 'digital tabletop' für eine neue, aufstrebende Klasse von Geräten die viele neue Arten mit digitalen Inhalten zu interagieren ermöglicht. Eine Klasse von Geräten, die es Benutzern erlauben direkt mit digitaler Information zu interagieren die auf großen, horizontalen Bildschirmen angezeigt wird. Da es sich um ein relativ junges Forschungsgebiet handelt, sind viele Entwicklungen im Fluss. Hardware und Software entwickeln sich mit hoher Geschwindigkeit und zurzeit gibt es viele unterschiedliche, teilweise konkurrierende Ansätze. Von zentralem Interesse für unsere Forschung sind Geräte die in der Lage sind mehrere Kontaktpunkte (z.B. Fingerspitzen) aber auch weitergehende Informationen wie Umrisse von ganzen Händen oder anderen Objekten zu erfassen.

Wenn wir Objekte in der physikalischen Welt manipulieren, profitieren wir dabei von der Geschicklichkeit unserer Hände. In dieser Arbeit wird versucht diese händische Geschicklichkeit auszunutzen um dem Benutzer vielfältigere Interaktionsmöglichkeiten zu eröffnen. Dies ließe sich zum Beispiel durch physikalische Objekte als Interaktionsmedium erreichen, oder durch die Verwendung von virtuellen Objekten, deren Verhalten stärker dem realer Objekte ähnelt.

Wir haben einen Forschungsansatz gewählt der sich in zwei Phasen einteilen lässt, um ein besseres Verständnis der Materie zu erlangen. In einer ersten Phase wurden zwei Interaktionsstile anhand von Prototypen untersucht. Einerseits gestenbasierte Interaktion und andererseits Interaktion mit Hilfe von physikalischen Objekten ('tangible interaction'). Die Flexibilität und Physikalität dieser Lösungsansätze wird als Kriterium definiert um die Interaktionsstile zu bewerten. Darauffolgend werden die beiden Paradigmen auf Ihre Tauglichkeit als generelle Interaktionsmodelle hin untersucht.

In der zweiten Phase werden die gewonnenen Erkenntnisse als Grundlage für die Entwicklung eines neuen Modells für Tabletop-Interaktion verwendet. Dieses Modell kombiniert fortgeschrittene multi-touch Sensortechnik mit einer virtuellen 3D-Welt, deren Objekte von einer Physiksimulation aus dem Computerspielebereich kontrolliert werden. Der vorgeschlagene Ansatz erlaubt es Benutzern mit virtuellen Objekten zu interagieren in dem Konzepte die aus der realen Welt bekannt sind, wie z.B. Kollisionen und Reibung, angewendet werden. Dadurch werden eine Reihe von komplexen Interaktionen ermöglicht, zum Beispiel Interaktionen mit mehreren Fingern gleichzeitig, mit der ganzen Hand oder durch Verwendung von physikalischen Objekten.

Das Modell stellt einen Schritt in Richtung noch natürlicherer und intuitiverer Tabletop-Interaktionen dar. Allerdings sind die vom System wahrgenommenen Benutzeraktionen und die angezeigten Informationen nach wie vor an die Displayoberfläche gebunden. Dadurch ergibt sich eine tiefgreifende Einschränkung bei der Manipulation von dreidimensionalen (3D) Objekten. Um diese Problem zu adressieren, wird eine Technik vorgestellt die es erlaubt Objekte - konzeptionell – vom Tisch aufzuheben und daraufhin deren Position im 3D Raum zu bestimmen. Hierbei war unser Ziel das ursprüngliche Modell zu ergänzen und eine Interaktionstechnik zu entwerfen, die es erlaubt so direkt wie nur möglich mit virtuellen Objekten zu interagieren. Um dies zu bewerkstelligen wurden zwei neue Hardwareprototypen entwickelt, die es ermöglichen

Benutzerinteraktion zu messen die in größerer Entfernung von der Displayoberfläche stattfindet. Darüber hinaus werden visuelle Feedbackmechanismen vorgestellt die die Illusion erzeugen sollen, dass der Benutzer die virtuellen Objekte tatsächlich von der Oberfläche aufhebt.

Diese Dissertationsschrift leistet auf mehreren Ebenen einen wissenschaftlichen Beitrag. Mehrere neuartige Prototypen werden vorgestellt die während den Forschungsarbeiten erstellt und evaluiert wurden. Diese Prototypen werden verwendet um systematisch den Designspace zu erforschen. Dabei werden die Flexibiltät in der Benutzung und die vermittelte Physikalität als Kriterien verwendet. Die betrachteten Interaktionsstile werden daraufhin untersucht, ob sie die hoch dynamischen und oft unstrukturierten Interaktionen unterstützen die typisch sind für Tabletop-Interaktionen. In einem weiteren Schritt wird ein neues Modell für die Tabletop-Interaktion vorgestellt, das die Qualität dieser Interaktionen steigert. Abschließend wird eine Erweiterung dieses Modells vorgestellt die es ermöglicht so direkt wie nur möglich mit virtuellen 3D Objekten zu interagieren. Dabei werden Objekte auf dem (zwei-dimensionalen) Display angezeigt während die Interaktion im Raum drüber stattfindet.

Preface

This dissertation has been written and submitted in partial fulfillment of the requirements for the Degree of Doctor of Philosophy in the Department of Mathematics, Informatics and Statistics at the Ludwig-Maximilians Universität München.

I was a graduate student at the chair for Media-Informatics from June 2005 to July 2009 where I also conducted a lot of the research discussed in this dissertation. I also had the opportunity to complete two research internships at Microsoft Research in Cambridge, UK. That time was an invaluable experience for me, and the projects that I was part of during my internships form the basis for some of the central aspects of this thesis. Due to the different locations at which the research has been conducted and in consequence due to the different ideas I was exposed to on a day to day basis, my own understanding and convictions about my research topics changed significantly over time. The resulting thesis, hopefully, benefits from this evolution of ideas as it does not describe a single strand of research but rather describes a selection of related techniques. Personally I think each project I did during this timespan fed into the design of the next one even if none of the prototypes discussed in here is a direct successor to the previous one.

Another consequence of working in different places and with many different collaborators is that it became impossible for me to separate which aspects of the respective projects were my contribution and which have been contributed by others. Therefore I chose to write the majority of the text in this document using the more inclusive scientific plural. In an attempt to clarify my role in each of the individual projects I added a contribution statement to each of the respective chapters.

Acknowledgments

I would like to thank many people whom for their support and encouragement. The following list is certainly not exhaustive, and I apologize to anyone whom I've failed to mention.

First of all, I want to thank my supervisor Andreas Butz who made it possible for me to even start this endeavor. He provided me with an income, equipment and lab space. More importantly, he always had an open ear for my troubles, ideas and concerns and he tolerated my going off on wild tangents (including changing my topic altogether, twice).

Shahram Izadi has played an equally important role, providing me with the opportunity to intern with Microsoft Research, twice, and subsequently helping me to learn something new everyday. He never stopped encouraging me and challenging me to always give my best. Without his ideas and enthusiasm I wouldn't be the same researcher today. In between some late-night hacking and last-minute paper writing, we have also shared a few hot curries and cold drinks and become friends in the process.

Two other people to whom special thanks are due are: Sheelagh Carpendale and Andy Wilson. Sheelagh for her willingness to be on my committee even though she hardly knew me. Furthermore, I want to thank Sheelagh for investing so much of her valuable time and energy into helping me come up with and refine the structure for this thesis. Andy has shown incredible generosity in the two projects we did together. Thanks Andy for sharing not only your time and ideas but even granting access to source code and other resources without ever having met me.

I was very lucky to have been able to work in two friendly, fruitful and fun environments. In Munich: Prof. Hußmann, all the current and previous PhD students, and especially the members of the Fluidum research group have made the daily grind feel more like a piece of cake (sometimes – especially on birthdays). I'd like to thank Dominkus Baur for enduring my supervision and still being brave enough to share an office with me. Sebastian Boring who has always been the right address to discuss all things football even though he is clearly rooting for the wrong team. I also want to thank all the students who have contributed to my research. In addition I'd like to acknowledge the support and friendship of Andreas Pleuss, Richard Atterer and Paul Holleis. Special thanks go to Alexander De Luca for keeping me entertained with his unique sense of humor, feeding my brain with abstruse trivia facts, keeping me company at lunch and his shared appreciation of the simple pleasures in life.

At Microsoft Research in Cambridge I have been fortunate to be allowed to work with a great group of people who create an incredibly stimulating environment: Steve Hodges, Alex Butler, James Scott, Stuart Taylor, David Kirk, Nicolas Villar, Alex Taylor and Sian Lindley have made my time in Cambridge a very exciting one and I can't wait to go back there. Special thanks belong to Armando Garcia-Mendoza who has been helpful and kind beyond belief – foreach line of code; thank you.

Of course none of this would have been possible without the support of my family. My parents, who never told me "to stop fooling around with computers (...) and become a radiologist" [SC94] but instead allowed me to choose my own path. My sisters for egging me on (unintentionally). By the way I guess I owe Rita beer.

And above all I want to thank my wonderful wife Amy who has been with me from the beginning of this endeavor. She has given me support when I needed it most, distraction when I needed to get my head off work and love – all the time. I hope she will continue to be my friend and partner for the rest of my life.

Contents

List of Figures

Part I

Introduction and Motivation

The following main body of this dissertation is structured into four parts. Each part builds on the previous ones and hopefully the chosen structure helps the reader to follow the evolution of the authors thinking throughout the timespan covered in this dissertation. Overall this structure mirrors the research approach taken. We discuss a variety of interaction techniques that have been developed and studied. Although these techniques are not direct successors to each other, clearly the findings from our evaluations have fed into the design of following projects.

Part I lays the foundation to our own work by introducing (Chapter 1) the reader to the general field of digital tabletop computing. In this part we also briefly touch upon common approaches to designing interaction techniques for this new class of computing devices. We argue that a new, improved model for tabletop interaction is necessary and specify requirements for such an interaction model. The second half of this part of the dissertation is made of a discussion of related work and recent developments both on the hardware side of things as well as in terms of interaction techniques (Chapter 2).

Part II then discusses our own initial explorations into tabletop computing based on two prototypes that we have built and evaluated. Here we discuss two common interaction paradigms and analyze them towards their suitability as general interaction model for digital tabletop computing. We discuss each approaches' strength but also its shortcomings based on the criteria of *flexibility* and *physicality*.

Part III then builds on the insights gathered from our analysis and proposes a new model for tabletop interaction that aims to combine the important aspects of *physicality* and *flexibility* in a coherent way. A model that increases the interaction fidelity of direct-touch interfaces beyond the "finger-as-cursor model" while maintaining a maximum of flexibility. We also discuss an iteration of the presented model that addresses issues caused by the mismatch of input and output fidelity in Chapter 9.

Finally Part IV summarizes the work presented in this dissertation and draws conclusions across all previous parts (Chapter 10). This part also hosts the index and bibliography of this dissertation.

Chapter 1

Introduction

Digital tabletops are an emerging class of computing interfaces characterized by large, horizontal and interactive displays built into tabletop form factors. These systems defy easy categorization and both tabletop hardware and software are changing at a fast pace, making these exciting times for digital tabletop research. When I started the research that has led to this dissertation, hardly any interactive tabletops were commercially available, the exception being the DiamondTouch [DL01] from Mitsubishi Electronics, but units were sparse and costly (approx. $ 10,000). Yet, not four years later, new tabletop products from Microsoft, SMART Technologies, Phillips and the like, in addition to various research prototypes are continuously changing the landscape we operate on. From an human computer interaction (HCI) research perspective, one appealing characteristic of digital tabletops is the possibility to directly interact with digital information by touching on-screen elements with fingers or a stylus. Precisely how this interaction happens varies greatly across different devices and proposed applications. Finding a new and more comprehensive model for tabletop interaction become the impetus for my research and is consequentially the main goal of this dissertation.

When we interact with the digital through graphical interfaces based on the traditional *W*indows *I*cons *M*enus *P*ointers (WIMP) paradigm, we are stripped of our manual dexterity; input happens through one single point of contact. In comparison to the flexibility that we possess when interacting with the real world, this single-pointer model can be limiting. Imagine unscrewing the lid from a water bottle with just the index finger of one hand. If we are restricted from using our non-dominant hand to position and hold in place a piece of paper, many of us would also have difficulties writing on paper. In addition, WIMP software assumes that only one person interacts with the system at a time and input happens sequentially, in a discrete fashion (e.g., pointing to a thumbnail, clicking for selection, *Ctrl-C* for copy). Our aptitude to carry out complex interactions with the real world is largely dependent on the parallel use of multiple fingers and both hands. The muscle groups of our fingers are controlled by a disproportionally large area of the motor cortex [CMR98]. Balakrishnan and MacKenzie [BM97] argue that this coordinated effort of many small muscle groups results in a high-input bandwidth. It would therefore stand to reason that unlocking the full interaction vocabulary of our hands may increase the fidelity of input in tabletop computing systems. This logic plays an important role in my research which makes the assumption that a more manual and natural style of interaction, which more closely resembles the fluidity of real-world interactions, will allow users to seamlessly engage in digital activities using tabletop computers. Allowing users to interact with such systems without interrupting the activities they were engaging in. That is interacting *socially* with another user, without being encumbered or absorbed by the need to also interact *with the computer*.

While hardware becomes more readily available and matures as a platform for home and office applications, it is also apparent that the degree to which these devices support multi-touch input varies greatly. Sometimes input is limited to one or two styli; other systems may provide recognition for several fingertips simultaneously; while yet another might be able to sense fingers and objects in parallel. As the hardware arena continues to develop, a variety of interaction styles have been proposed.

In order to gain a better understanding of the design space, we decided to take a closer look at some of these interaction techniques. First, we explored several interaction styles using prototypical implementations. These prototypes were then evaluated using *flexibility* and *physicality* as criteria. *Flexibility* refers to the openness of the interaction: how well does the interface lend itself to experimentation by the users? How easy is it to discover functionality through simple experimentation and the application of real world knowledge? How well can systems based on this interaction style be designed for appropriation? *Physicality* refers primarily to the level of realism in the behavior of interface elements. It also refers to how well the interactions exploit our fine-grained motor skills and manual dexterity.

Based on the findings from these initial explorations, we propose a new model for tabletop interaction aimed at improving the fidelity of interaction with such systems. A model that affords a variety of fine-grained ways to manipulate virtual objects akin to manipulating real objects. This approach looks at the intersection of emerging multi-touch hardware and game physics simulations. The presented model enables richer, more open interaction than previously possible by removing application logic and scripted behavior from the system. In consequence, this interaction model enables users to apply manipulation strategies from the real world to virtual objects through the exertion of friction and collision forces, thus allowing for experimentation and appropriation.

One major issue uncovered by studying our proposed model was the mismatch of input and output dimensionality. While virtual objects are three dimensional and live in a 3D simulation, input only happens in two dimensions. Imagine interacting with 3D objects through a thin sheet of transparent film that allows you to push objects around but does not allow you to grab objects nor does it allow you to pick objects up. This causes severe problems if users were to try to position objects on top of other objects.

In an attempt to counteract this issue, we began to explore depth-based interaction on interactive surfaces. We present the results of this exploration; a new interaction technique that allows users to – conceptually – lift 3D objects off the 2D surface, almost as if they were interacting directly with the virtual object. The technique also allows for fine-grained control over the object's position in 3D, enabling for example precise stacking of objects.

1.1 Background and Motivation

(a)

(b)

Figure 1.1: a) A group meets around a traditional table using paper. b) Various objects found on a typical office desk. Objects are loosely stacked, sometimes organized into piles. Interaction happens ad hoc.

Tables can be found in practically every environment of our daily lives. We can find them in work settings as conference tables or as desks, or at home, as dining or coffee tables. Their form factor supports a variety of uses: for sitting or gathering around in order to interact with others (see Figure 1.1 a); for working on with pen and paper as well as computers, mice and keyboards. Objects can be stacked and organized into piles (Figure 1.1 b) on their surface. Finally desks allow a surface for us to rest our hands and forearms on while carrying out work tasks or engaging in social activities with friends and family.

Emerging digital tabletops promise to combine the qualities they inherit from their non-technologically enhanced ancestors combined with the capability to display and interact with digital information. In work settings this could enhance the ability to display and share digital work artifacts.In private settings, horizontal *interactive* surfaces could be used to comfortably share digital photo or music collections with others while affording eye contact and other social cues that are so important to face-to-face communication. Of course, computing power brings other benefits such as permanent storage or replication of data, visualization of large data sets and the ability to efficiently sort or search for specific content.

In recent years these interactive surfaces have become more and more widespread in research and as commercial products. Users of these systems often comment on the unique user experience. Being able to directly touch digital information in combination with rich interactive applications – providing direct coupling of input and output – is often described as intuitive or natural. On traditional tables, people also regularly manipulate a variety of physical objects. Digital tables likewise offer a supporting surface, allowing for physical objects to be used as tangible input devices. This offers the possibility of intuitive manipulation and a potentially better integration between digital and physical interaction spaces.

The many positive properties accredited to digital tabletops have led to much interest among researchers and commercial companies. Many compelling scenarios have been proposed, such as browsing and sharing of digital photographs, interacting with maps and other geographical information and strategic planning applications. Most state-of-the-art interaction techniques however, are typically limited to 2D movements and rotations within the surface plane.

Both single-user [WPR+07] and multi-user [MPWS06] scenarios have been investigated on interactive tabletops with one or several discrete input points from each person, typically simulating pointing with mice. Others have explored using multiple contact points together (multiple fingers or both hands) to enrich interaction, for example scaling and rotating objects [KCST05, MH06], or enabling precise selection [BWB06]. Many of these proposed interaction techniques rely heavily on two-dimensional pointing and selecting similar to mouse-based interaction in WIMP interfaces. We call this the "finger-as-cursor" model. Recently hardware has become available that allows for much richer sensing than simply identifying multiple fingertips as cursors. These interactive surfaces can sense whole hands and physical objects placed on them. This allows for the creation of much richer and more expressive interaction techniques unlocking the users' manual dexterity and motor skills accrued through lifelong experiences with the real world.

1.2 Problem Statement

The hardware arena is changing at a rapid pace. Simultaneously, many different approaches to interaction design in this space have been proposed including pen-based, multi-touch and tangible interaction. Many of these interaction styles have similar properties (e.g., directly coupled input and output, physical interaction quality, support for bi-manualism). However, interaction techniques have often been designed in an ad-hoc manner and studied in isolation. This thesis offers an early exploration into a variety of these interaction styles in order to gain a better understanding of the design space as a whole.

One aspect that appears to contribute to the popularity of digital tabletops is a physical interaction quality that simulates real-world interactions. Not surprisingly many interface designs feature some sort of physical behavior. This can be roughly grouped into two categories: giving virtual objects momentum or physical extent. Examples of the former are flicking – or tossing – based techniques [RGS+06] or the common rotate and translate, where virtual objects behave analogous to a sheet of paper dragged over a flat surface with one or more fingers [KCST05, LPS+06]. The latter is formed by techniques such as the data mountain [RCL+98], where objects avoid other objects or push them aside as if they have actual physical volume. Many of these techniques may feel real or natural, but their behavior is often preprogrammed or scripted. For example, while a photo rotates and translates with seeming naturalness, under the hood there still is interaction logic and heuristics that are being executed. This approach may break down once users interact with the system in ways unanticipated by the developer.

The central goal of this thesis is to develop an interaction model that increases the interaction fidelity of digital tabletop computing, while maintaining maximum interaction flexibility, thus allowing virtual objects to feel more real and enabling users to interact with these objects by applying strategies from the real world. In order to tackle this problem, first we had to posit the important question:

What is the Interface? Real-world interactions benefit from tactile, haptic and other rich sensory feedback. They also benefit from our aptitude to manipulate objects in various ways. This physicality is only poorly represented by most direct-touch interfaces. This thesis is an exploration into different possibilities to provide richer interactions, including tangible interfaces and virtual objects that behave in a more realistic way.

In summary, the goal is to understand the individual properties of various interaction styles that have been proposed for tabletop interaction. The knowledge gathered from building and evaluating different prototypes then serves as input to the design of a new model for tabletop interaction. This new model aims at providing both *flexibility* and *physicality*. *Flexibility* refers to the possibility to interact with the system through various strategies, or in other words, unrestricted by specific, ad-hoc design decisions on the developers side. Physicality refers to both the behavior of the interface (i.e., realism), as well as the extend to which an interface allows users to apply their full set of motor skills and manual dexterity.

During the development of our new interaction model a a secondary goal became apparent. In order to be able to approximate the rich ways in which we interact on regular tables it is necessary to open up the space directly above the table for interaction. On real tables we routinely pick objects up flip them around, put them into containers or stack several objects. In consequence the final parts of this dissertation deal with the following question:

Where is the Interface? Tabletops have planar two-dimensional displays, therefore it is often assumed that applications and interactions have to be two-dimensional as well. Only recently have researchers begun to look into the possibility of interaction in three dimensions. Be it through 3D graphics displayed on a tabletop display, or richer sensing that allows for interaction in the volume above the tabletop. When interacting with 3D content using 2D input many actions, that are a matter of course in the real world, become difficult. For example, picking up an object in order to place it on top of another one.

Here the goal is to provide users with interaction techniques to lift objects off the plane and control their position in 3D. For example, one could imagine a virtual book with pages that behave just like paper and an interaction technique that allows us to leaf through the book, as effortless as in the real world. To achieve this goal we have to deal with two subsequent problems. First, most tabletop devices only detect input that happens directly on the surface. In order to enable 3D interactions in a seamless fashion we need to extend the currently available sensing capabilities. Second, when interacting with objects displayed *on* the surface from *above* the surface one of the primary challenges is the lack of feedback during in-the-air interactions. We aim to provide techniques that compensate for this issue.

1.3 Thesis Overview

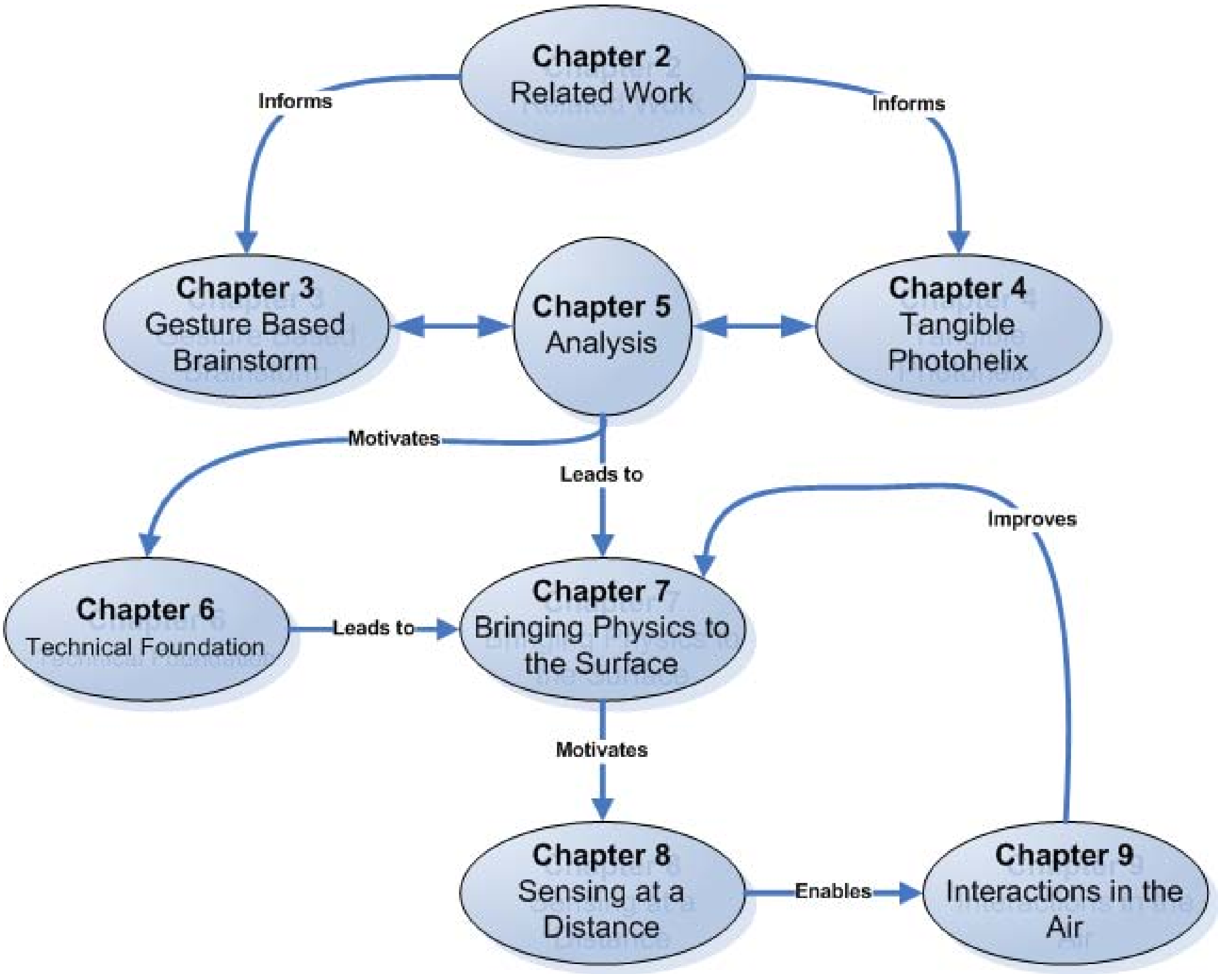


Figure 1.2: Thesis Roadmap

This thesis has an unusual structure due to the explorative nature of the research. The document (and indeed research approach) can be organized into four parts. The first two parts introduce to the general field and help gather an understanding of the tabletop interaction design space through literature review. Then we built several application prototypes, which serve as study object for direct-touch, gesture-based interaction and tangible interaction in combination with touch input respectively (Figure 1.3). Both approaches are then analyzed toward their fulfillment of the requirements for *flexibility* and *physicality*. In the third part a new model for tabletop interaction is introduced, enabling fluid and flexible interactions with the virtual, resembling the rich interactions that we are used to from the real-world. In response to findings from lab-based user studies, another iteration of this approach is presented enabling users to interact in the space above the table. The fourth and final part draws conclusions from all previous parts. The thesis can be broken down into chapters as follows (see also Figure 1.2):

Chapter 2: Surveys relevant related work in the fields of tabletop interaction. A brief overview of tabletop hardware is given, followed by a discussion of gesture-based interaction techniques and tangible user interfaces on tabletops.

Chapter 3 and 4: Explore early application prototypes (Figure 1.3). The prototypes are discussed in detail, with specific attention to their interaction styles. Both prototypes stand for one of the typical interaction styles common among many tabletop projects: gesture-based interaction (Chapter 3) and tangible user interfaces in combination with direct-touch input (Chapter 4).

Chapter 5: Explores the respective qualities of the prototypes detailed in Chapter 3 and Chapter 4. Results from lab-based user studies and observations from many hours of system usage are presented. The chapter concludes with a discussion of the qualities and limitations of each approach. We use *flexibility* and *physicality* as criteria.

Chapter 6: Discusses various approaches to sensing multiple fingers, hands and objects. We also introduce our own interactive surface prototype based on liquid displacement sensing (Figure 1.4 a). We discuss various design considerations and trade-offs.

Chapter 7: Debuts our innovative model for interaction on digital tabletops. The model combines rich input from interactive surface hardware, is capable of sensing rich surface input with a 3D environment, and is powered by a gaming physics simulation (Figure 1.4 b). We describe various iterations of the model and discuss trade-offs between implementation alternatives. Finally, we discuss findings from an initial lab-based user study.

Chapter 8: Discusses novel hardware developments which enable the projection of an image on a display surface while simultaneously capturing the image of users' hands and other objects at a distance.

Chapter 9: Offers an initial exploration of depth-based interaction on digital tabletops, which enable users to – conceptually – pick virtual objects up from the tabletop and manipulate their 3D positions before placing or dropping them back onto the surface (Figure 1.4 c). We discuss two rear projection-vision prototypes. Both use special projection-screen materials to allow sensing at significant depths beyond the display surface. We present a novel shadow-based technique to help alleviate feedback issues when interacting with objects displayed *on* the surface from *above* the surface.

Chapter 10: Concludes with a discussion of the presented interactive systems and their interaction styles. We discuss qualities of the proposed solution and its limitations. We also present areas for further exploration.

1.4 Contribution

This thesis contributes on various levels. We present several novel prototypes that we have built and evaluated. We use these prototypes to systematically explore the design space of digital table-

(a) Brainstorm (b) Photohelix

Figure 1.3: Two initial explorations: a) Gesture based interaction in the Brainstorm system. Virtual objects borrow their appearance and behavior from the real world to increase discoverability and learnability. b) Combining tangible and direct-touch interaction for collocated sharing of digital photos. Using a physical handle to control interface parameters frees cognitive resources by exploiting motor skills and rich interaction vocabulary of hands and fingers.

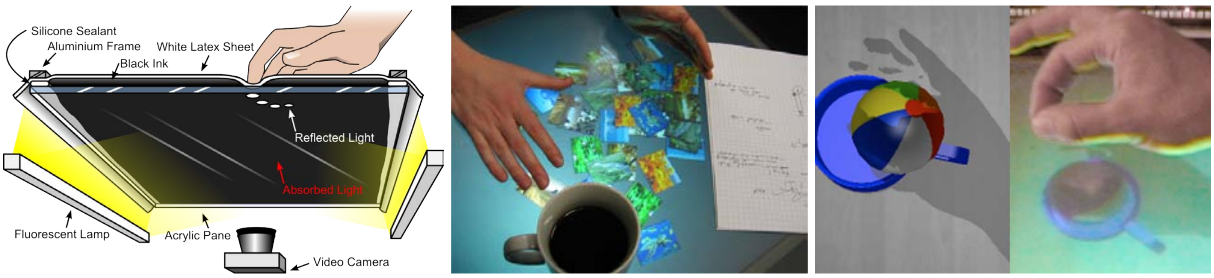
top interaction. In order to analyze and compare interaction techniques we introduce *flexibility* and *physicality* as criteria. We also present a new model for tabletop interaction that increases the fidelity of interaction possible in such settings. Furthermore, we study this model based on two prototypes. Finally, we present a third prototype as initial exploration into the most important issues uncovered by our evaluations. The major contributions are:

Analysis of tabletop interaction. Tabletops afford many ways to interact with them. In order to understand the design space better we undertook an initial exploration into different interaction styles. A broad survey of literature on tabletop interaction is presented in Chapter 2. Here we discuss interaction techniques and high-level interaction styles, such as gesture-based or tangible interaction. In Chapter 3 we present a prototype that explores how large direct-touch enabled displays embedded into the environment can facilitate face-to-face communication and creative group work. We also present results from a user study comparing our system with paper-based brainstorming. We then present Photohelix (Chapter 4), an exploration into hybrid interfaces – combining tangible objects and direct-touch input in one application design for co-located browsing and sharing of pictures. We also discuss results from a quantitative user study.

New criterion for evaluation. As means to position the two presented prototypes and their respective interaction styles within the broader tabletop design space, we use the concepts of *flexibility* and *physicality* as criteria. We show in Chapter 5 how both prototypes fulfill some of the requirements of these criteria and highlight conceptual problems rooted in the respective interaction styles.

New model for tabletop interaction. Based on our learnings from the initial explorations we propose a new model for tabletop interaction in order to achieve increased openness and naturalness in the interaction. To provide an in-depth understanding of the technical foundations (Chapter 6) underlying our interaction model. We discuss emerging tabletop hardware as discussed in the literature (Section 6.1) and propose our own alternative approach to interactive surface sensing (Section 6.2). Our model (Chapter 7) is based on virtual objects that behave more akin to the real world and rich input using fingers, hands and tangible objects. This model allows users to apply their interaction strategies perfected through everyday manipulation of physical objects. This model also contributes on a different level as it hides many of the complexities when developing rich tabletop applications, augmented by a physics gaming engine. Our techniques are transferable to different hardware devices.

Enabling 3D interaction. The most pressing issue with our model is the lack of 3D control over objects. For example, since it is not possible to lift objects off the surface, it is difficult for users to place objects on top of, or into other objects for storage. In Chapter 9 we begin to address this problem by introducing sensing and interaction techniques that enable interaction in the space above the table's surface. Our technique affords fine grained 3D control over virtual objects. Interacting above the surface in this way opens up many interesting challenges that we describe in Chapter 9.



(a) Liquid Displacement Sensing (b) Surface Physics (c) Depth Based Interaction

Figure 1.4: A new model for tabletop interaction: a) A novel approach to sensing rich user input. b) Interacting with virtual objects controlled by a gaming physics engine. Interaction is possible using whole hands, multiple fingertips and tangible objects. c) 3D Interaction above the table. It is possible to – conceptually – lift objects off the surface.

Chapter 2

Related Work

> "Curiosity is the essence of the scientific mind."
>
> **– Bill Watterson**

This thesis aims at developing new ways to interact with digital tabletops. In order to do so this chapter starts with a brief overview of various proposed tabletop sensing approaches. Many of the projects only briefly touched upon in this chapter are discussed in much more detail in Chapters 6 and 8. Some of the hardware discussed here, has been designed in order to support a particular application or interaction style, but mostly hardware development happens independent from interaction research (due to the complexities of both). In contrast, our own model for tabletop interaction is tightly interwoven with and sometimes driven by novel hardware developments. Therefore we chose to discuss the relevant projects closer to the description of our proposed interaction techniques in Chapters 7 and 9.

To gain a better understanding of the current state of the art in tabletop interaction techniques we discuss many interaction techniques proposed in the literature in depth in this chapter (cf. 2.2). We furthermore classify various projects into two main interaction styles: gesture-based (Section 2.2.2) and tangible interaction (Section 2.2.3). The discussion of these interaction styles functions as basis for our own explorations into these interaction paradigms in Part II. There exist a raft of direct-touch interaction techniques designed for devices that only support the detection of a single contact point. A thorough discussion of these interaction techniques would go beyond the scope of this thesis. One of the appealing qualities of tabletop computing is the possibility to interact with multiple fingers simultaneously and the focus of this thesis are interaction techniques for multi-touch enabled tabletops. Therefore this discussion of related work focuses on interaction techniques designed for multi-touch devices.

2.1 Tabletop Hardware Overview

Sensing fingers and other objects on an interactive surface is a fairly non-trivial task. This has led researchers to experiment with various techniques. Common approaches to sensing can be loosely grouped into two categories: 1) camera-based sensing and 2) sensor electronics integrated into the surface. In the latter, many techniques have been proposed including ones based on capacitive [App08, DL01, Wes99], resistive [Jaz08], or IR sensing [HIB+07, IBH+08]. In the former, numerous systems have been built using the camera as the sensor, coupled with some form of illumination scheme, be it FTIR [Han05] or other diffuse lighting [Mic08, MR97, Wil04].

Camera-based approaches potentially give higher resolution sensing and can scale to detecting objects beyond fingertips, even supporting unique identification of objects using visual markers [JGAK07, Wil05]. What is perhaps most compelling about these systems, which has led to a great deal of adoption in the community, is that the sensing electronics (the camera) is readily procurable, allowing people to prototype such systems with little electronics expertise.

In these systems, many different arrangements of camera, projector and surface have been experimented with, including projection and sensing from above [Wel93], from the bottom or rear [Han05, Mic08, MR97, Wil04] or off axis [Wil05]. To avoid occlusions many systems position the camera behind the projection surface. Usually an IR light source and a camera equipped with an IR pass filter are utilized to sense contours of IR reflective objects placed on top or in front of the surface [JGAK07, MR97, Wil05]. This approach does not only track multiple fingers by several users but also allows sensing of objects. Objects can also be equipped with reflective markers or bar codes in order to identify individual objects.

Han [Han05] presents a different approach to multi-touch sensing based on FTIR. It works by mounting IR LEDs around the edge of a sheet of acrylic (or other material with similar optical properties). These LEDs shine light into the surface. The light is totally internally reflected inside the surface, and touching fingers causes some of this light to scatter downwards where it can be imaged by a camera. This is perhaps one of the most established techniques for sensing multiple touching fingers, partly because the illumination scheme greatly increases the signal-to-noise ratio making the processing of the raw sensor data much more straightforward than with diffuse illumination.

2.2 Tabletop Interaction Techniques

Following the overview of hardware platforms we ill now discuss some proposed interaction techniques. Due to the head start the **DiamondTouch** system enjoyed in terms of commercial availability, it is particular noteworthy. We start our discussions with several techniques and studies that are tightly bound to the system's capabilities. We continue with a discussion of interaction techniques that have been designed without a particular platform in mind or that utilize custom build prototypes. In the discussion we identify different interaction styles in order to inform our own explorations in the following Chapters.

2.2.1 Interactions based on the DiamondTouch

The **DiamondTouch** [DL01] system is particular noteworthy because it is capable of distinguishing input from multiple users reliably which is a feature no other important hardware platform supports. Also the system has been used in many academic studies and therefore plays an important role in forming our understanding of tabletop usage and interaction. However, the system enforces a particular interaction style because, in its unmodified state, it reports relatively little information upon contact. The limitation to an axis-aligned bounding box per user (potentially stemming from multiple fingers) does not allow for the same flexibility in input interpretation and therefore user interaction that other systems could provide. Because of these limitations many applications developed for the **DiamondTouch** platform feature an interaction style that could be described as an extension to the traditional WIMP paradigm. Interaction happens mostly through one single finger tip and is often restricted to 2D pointing, selection and movement.

Ryall, Forlines, Shen and Morris investigated the effect of group and table size on collaborative performance. The experiments were conducted on two differently sized interactive tables. **Group size** and its effects on task performance was also studied. Groups of two or four participants had to assemble a poem from individual tiles. While no significant effect was found between the **table sizes**, larger groups were faster than smaller groups. Furthermore, participants reported a strong personal preference for the larger table size and *felt* as if they completed the task faster. A variety of implications for resource management and social interactions (e.g., level of individual participation, conflict resolution) are reported. An interesting finding was that *where* work resources are displayed influenced work strategies adopted by groups which has strong implications for the design of applications for group collaboration. For larger groups additional vertical displays might be beneficial especially in data intensive tasks.

Because of the unique user identification capabilities the platform has been utilized to study techniques that depend or enforce **object ownership**.
Transitioning from individual to group work is a common practice in collaborative work. This practice often involves the passing on of work resources to other collaborators or releasing resources to the entire group. Ringel et al. [RRS+04] present and study several interaction techniques that ease the transition of ownership status between group members or from private to public accessibility. Their "relocate" technique was more efficient and also was preferred over the "resize", "reorient" and "release" techniques. In this technique ownership is associated and enforced via special areas on the tabletop display reserved for private or shared use.
In a later publication [RMRS+04] more general coordination strategies and issues are observed and discussed. Based on these observations a set of techniques is discussed to improve group collaboration interfaces through a "coordination policies" framework.

With **SIDES** [POMW06] Piper et al. present and study ways to help adolescents with Asperger's Syndrome practice effective group work skills using a multi-player interactive tabletop game. The findings indicate that cooperative tabletop computer games can be engaging and may support the acquisition of effective group work strategies among individuals with special needs - in particular when the hardware can identify who is interacting. Also several design lessons to inform the development of similar systems are discussed.

Morris, Paepcke and Winograd [MPW06] present in **TeamSearch** ways to specify boolean search queries either collaboratively or concurrently using meta information and query tokens. A study investigated whether queries constructed via a group effort or performed individually are more efficient (i.e., faster, better results). The results did not show significant efficiency benefits

for the group effort, but that collective query formation has advantages in terms of enhancing group collaboration and awareness, suggesting that group centric UIs may offer other benefits beyond the efficiency and result quality usually accredited to them.
TeamTag [MPWS06] uses a slightly modified interface to support bio-diversity researchers in classifying photographs of different animals. Tags may be applied by first touching the appropriate category and then the photograph to be tagged. Of research interest was whether participants would prefer centralized or replicated controls. Users showed a clear preference for the replicated controls albeit no efficiency benefits could be found.

The **Personal Digital Historian** [SLM+01, SLV03] is a photo sharing application specifically targeted at co-located sharing of photographs and story-telling. The system supports users in constructing narrations from a vast database of digital pictures by providing different views onto the image collection. The different layouts are organized along the four "W's" of storytelling; where, when, who and what. While many interesting information presentation and interface aspects for co-located group ware are introduced and discussed the interaction techniques are mostly limited to point and click interactions.

Forlines and Shen [FS05] present in **DTLens** interaction techniques to enable multiple users to interact simultaneously with geo-spatial data such as multi-layered maps. Users may create magnifying lenses by touching the display with two fingers. The size of the lens is identical to the bounding box reported by the hardware. Several interaction techniques are available to reposition and resize the lenses or change its zoom level. The system takes advantage of the user identifying hardware to mitigate interaction conflicts. Lens parameters may only be adjusted by the creator of the lens greatly easing transitions between individual and group work.

Wigdor et al. [WPR+07] report findings from a long term case study of one individual user performing his **everyday office work** tasks on an interactive tabletop. Relatively few differences in interaction style were found in comparison to a standard PC (not surprisingly since the tabletop was used to drive a standard MS Windows environment). However, some actions were performed using both hands. In most observed cases of bi-manualism the user applied this strategy to increase his reach and therefore optimize task performance, an interesting finding because many research articles speculate about the particular suitability of tabletop systems to support bi-manual interaction for complexer interaction styles. Some findings impacting ergonomics are offered. The user clearly preferred his tabletop mounted angled toward him but had the conflicting requirement for dual use of the device as storage space for desk clutter and sometimes meeting venue. Finally text input issues regarding the usage of a software on-screen keyboard are highlighted.

Shen et al. [SRF+06] summarize findings from various projects implemented on the **DiamondTouch** and suggests guidelines **informing the design of direct-touch tabletops**. An extensive set of informal observations complemented by results from controlled user studies are used to illustrate these guidelines. Furthermore, the authors stress the importance of interaction techniques specifically designed for tabletop interaction rather than simple adaptation of mouse and keyboard based interaction techniques.

Although many exciting application scenarios for digital tabletops have been explored it is still not entirely clear what might be the "killer application" for digital tabletops. However, in the end this might not be an overly important question – if we look back at the reasons for adaption and life-cycles of other technologies it can be said that many needed to go through several iterations and refinements before they were ready for the mass market (e.g., the mouse was invented in the 60s but only became widespread in the late 80s). It appears that multi-touch tabletops are not a platform that lends itself toward a simple extension of the desktop computing paradigm but an entirely independent class of devices best suited for special purpose applications. We would further argue that for the time being tabletop computing should still be considered as a discipline in its infancy. If we accept this precondition we should also be willing – and trying – to openly think about every aspect of the domain (and potentially question it). Therefore, we will concentrate our discussion of related work, but also our own explorations later on, on aspects that determine how we interact with the virtual realm. This inevitably shifts the focus away from applications and application specific design decisions towards more low level questions of how to move and otherwise manipulate on-screen objects.

2.2.2 Gesture-Based Interaction

Standard PC interfaces rely heavily on icons, buttons menus which are operated from a single viewpoint with one single pointer. Tabletop UIs are subject to different design constraints, especially multiple viewpoints, simultaneous interaction and the special role of orientation [KCSG03] make the usage of precisely these interface elements problematic. For example, buttons that issue commands which might change the mode or view of the entire interface might be disruptive for collaborators currently viewing different parts of a large dataset. Regular menus can also be problematic because they might not be readable or reachable from all sides of the table. Replicating all menus mitigates the problem but with the cost of wasted screen real estate, especially precious on typically low resolution tabletop displays. And centralized controls for all group members can be the source of conflict when several users interact simultaneously [MPWS06].

A particular popular design alternative for tabletop interfaces are *gesture-based* interfaces. Hand gestures in HCI can carry different meanings and have been interpreted in various ways by researchers. Generally speaking they are hand postures or movements that carry some meaning or express an idea. In order to be used in GUIs these gestures need to be sensed and interpreted in order to trigger some event or perform a command. They may be simple pointing gestures that invoke a command in combination with an on-screen tool. They may be heuristically recognized drawings or traces of more or less abstract glyphs, or they maybe some sort of symbolic gesture following a real-world metaphor. These different types of gestures can be arranged on the input continuum defined by Zhai and Milgram [ZM98] ranging from the direct to the indirect. Simple pointing to touch screen elements could be classified as direct while performing a metaphoric gesture would reside on the indirect end. In this Section we summarize some of the interfaces most related to our work that make use of some form of hand gesture. Many of the discussed techniques follow a hybrid approach where indirect gestures are used to perform some sort of mode-switch followed by a direct gesture.

An early example of direct gestures was presented by Minsky in 1984 [Min84]. The system recognized selection, move and path gestures for manipulating virtual elements on a single-user touchscreen. There were also virtual elements or tools that could be dragged over other virtual objects to perform specific actions such as copying.

Another early example of gestural input, albeit on the other end of the spectrum is the **Charade** [BBL93] system. A set of gestures is introduced to perform various commands that control a presentation system such as advancing pages or jumping to the table of content. The hand postures and movements are recognized using a data glove connected to the presentation system. In the Charade system input and output are not directly coupled as gestures can be performed anywhere while the resulting action is always performed on the main projection screen.

An example for a gesture based tabletop application [WB03] is the **RoomPlanner** system where two or more users interact simultaneous with a furniture layout application. Several direct one and two finger gestures are available to position and orient furniture elements. Furthermore several indirect, pie-menu based gestures are available to copy or create new furniture elements. A set of whole-hand gestures allows users to interact with several objects at once. An exploratory user-study revealed that the system successfully leverages multi-finger and whole-hand interactions people perform on traditional tables. Some users reported difficulties in understanding or remembering particular gestures and requested variations of available gestures or the addition of new gestures.

In an attempt to ease the design of novel gesture-based touch interfaces Wobbrock, Wilson and Yang [WWL07] developed the "**$1 recognizer**" which appears to greatly ease the process of incorporating gesture recognition in tabletop applications. Besides being simple to implement (the algorithm is presented in 100 lines of pseudo-code) it achieves high recognition accuracy with small sets of training data (97% with one template, 99% with 3+ templates). Especially interesting with regards to tabletop interfaces is that the algorithm provides rotation, scale, and position invariance. However, the $1 recognizer is a recognition *algorithm* and does not supporting developers in designing and evaluating new gestures, for example through a visual editor or guidelines for "good" gesture design.

Gestural interfaces are often designed ad-hoc and therefore resulting gestures need to be learned and memorized by users. In order to guide the design process and to reduce learnability issues Wu, Shen, Forlines, Ryall and Balakrishnan [WSR+06] developed a set of design principles (**registration, relaxation and gesture and tool reuse**). The primary focus of this work is the reduction of gesture primitives through reuse thereof as well as to provide a systematic approach to the design of custom gestures. In an example photo-handling application several gestures to copy, annotate and layout digital photos are presented. A study revealed issues with granularity of input and need for visual feedback to guide users when performing complex gestures.

Morris et al. [MHPW06] extend the gesture concept to **cooperative gestures** - multi-user gestural interactions for single display groupware (SDG) applications. A set of gestures is explored that require different levels of cooperation between multiple users. A pair of users may exchange photos by different gestures that require input from both partners for example, one partner initiates the transfer of the photo while the other partner specifies which object to transfer over a distance. Other gestures require the input of the entire group, examples given are an automatic layout of the entire screen, wiping the canvas in a drawing application or quitting the application. User feedback indicated that group interactions with a clear need for agreement within the group were greatly appreciated but users complained about "unnecessary" collaboration in more mundane interactions which could have been easily performed by a single user. Problems with accidental triggering of multi-user gestures was reported in cases where actions with a single-user meaning are overloaded with a multi-user gesture.

Transferring information over greater distances from one user to another situated at different sides of a larger tabletop is a recurring problem. Reetz et al. [RGS+06] present and compare several flicking gestures, rapid pen or finger traces performed on an virtual object to indicate motion direction and initial momentum. The virtual object then slides across the surface where another user can receive and modify the object. Several variations of the basic flicking principle are shown and studied. While regular flicking is a fast technique it is also coarse. **Superflick** adds accuracy to the gesture by introducing a second aiming step. A comparative study showed that basic flicking is significantly faster than a standard radar view, albeit less accurate for small targets. A second study revealed no performance or accuracy differences between the radar view and superflick, indicating that throwing based techniques are a viable alternative for object transfer over a distance especially when considering their simplicity and lack of additional space and mode-switch requirements.

Rekimoto presents various multi finger gestures based on the **SmartSkin** [Rek02] hardware. Besides moving objects with one finger, panning the background or a map with two fingers, users may rotate, scale and translate (RST) objects simultaneously with two or more fingers. The recognition algorithm uses a least-squares constraint solving approach to compute a RST motion best suited to the position and relative motion of the user's fingers. Other examples are a four-finger technique to manipulate control points of a Bézier curve. The presented multi-finger RST technique can now be found in virtually every UI of multi-touch capable products (e.g., Apple iPhone, Microsoft Surface) and especially Apples line of multi-touch capable products made this the de-facto standard for multi-finger zooming in the mind of many consumers.

Kruger, Carpendale, Scott and Tang [KCST05] present a technique that enables integrated rotation and translation (**RNT**) of on-screen objects with just one point of contact. Virtual objects behave similarly to objects moving against friction or through a current in the real world. Contact in the center of the object only translates and contact off-center rotates and translates the object (aligning itself with the virtual current). A user study revealed performance benefits over the traditional method to scale and rotate objects by manipulating handles on the corner of selected objects as used in many WIMP UIs. The presented technique can be especially useful on hardware platforms that only allow one touch point or only one contact per user.

Hancock, Carpendale and Cockburn [HCC07] extend the aforementioned RNT technique to support "**shallow-depth**" interactions. Shallow-depth refers to a narrow band of limited width both above and below the surface. The interaction techniques allow to rotate and translate objects within the display plane and also allow for additional rotation around all three axes of the three-dimensional space (5DOF). Starting from one-point of contact, through two- and three-points of contact techniques are discussed. The one-point technique is designed for simplicity but lacks efficiency and accuracy when compared to the multi-finger techniques. The three finger technique is the most complex to learn but offers the best performance in terms of speed and precision. Users also strongly preferred the three-finger technique for it's expressive power.

The size of human fingers and limited resolution of most sensing approaches makes precise selection on multi-touch screens difficult. Benko, Wilson and Baudisch [BWB06] describe several **precise selection** techniques for multi-touch screens. Three techniques (*Stretch*, *X-Menu* and *Slider*) are introduced to allow for pixel-precise selection. The first finger controls a cursor while the second touch-point is used to adjust the control-display ratio. A comparative study pitched the three techniques versus an offset cursor as baseline. All three techniques outperformed the baseline and were preferred by the participants. Among the three techniques *Stretch* performed best both for speed / accuracy as well as user preference. In *Stretch* a secondary finger defines a rectangular area of magnification anchored at the primary finger's position. The first finger controls the cursor while the magnification rate can be controlled using the second finger at the same time.

While gesture-based multi-touch interaction has been demonstrated in various forms and also studied in the lab relatively few field deployments and long-term studies have been conducted. A notable exception is the **CityWall** [PSJ+07]. A large vertical multi-touch display was installed in the city center of Helsinki, Finland. The display shows information (mostly photographs) about events happening in and around the city. It also allows users to upload their own content from mobile devices as well as interacting with content through a set of "standard" multi-touch gestures. During the period of eight days the usage has been monitored and studied in depth [PKS+08]. Various typical usage patterns are identified for example, crowding, massively parallel interaction, teamwork, games and negotiations over conflicts and handovers of objects between co-present but unfamiliar users. Also several roles that users took on are discussed (e.g., *teacher-apprentice*, *comedian*, *bystander*). The system showed that public multi-touch displays can be engaging and are well perceived but tensions, mostly due to established social "norms", are reported regarding personal interaction in the public space and with publicly available resources.

Continuous Tabletop Interaction

Most interaction techniques discussed above use fingertips, pen or styli to control a cursor, trace gestures or directly select on-screen elements. In theory most of these techniques could be performed with one or many standard pointing devices such as a mouse. With regards to the flexibility and richness of interaction we enjoy in the real world, this cursor model can be limiting. Recently researchers have begun to explore interaction techniques that do not assume individual *points* of contact as basic input primitive but take richer information such as shape or motion into consideration.

In **VIDEOPLACE** [KGH85] interaction using whole hands or even the entire body (or the outlines thereof) was the primary mode of interaction. Identifying individual fingers to use them as pointers or gesture recognition was the exception (e.g., only used to switch between different application modes). However, most interactions within the **VIDEOPLACE** system were of artistic or communicative (between humans) nature.

Rekimoto demonstrates another shape-based technique in the **SmartSkin** [Rek02] system. The bi-cubic interpolation scheme used to increase the resolution of the sensor data produces a "potential-field". Peaks in this data correlate with conductive objects closest to the surface but all parts of the object create some elevation (see Figure 6.1). This potential-field is used for a shape based interaction technique where virtual objects are repelled by peaks in the potential field and always try to minimize their own potential, thus users may interact with virtual objects in different ways than the common drag'n'drop paradigm. An exploratory study showed that users were open to these shape-based interactions and quickly understood that they can use their hands and even whole arms.

Wilson and Cutrell [WC05] present a related but different approach to achieve more fluid and analogue interaction techniques. In the **FlowMouse** system an optical flow algorithm is used to compute a combined affine transformation for all three RST parameters from the motion detected in the camera stream of the users hand. The proposed mathematical model for motion calculation has been applied to other hardware platforms and application scenarios. In **PlayAnywhere** [Wil05] optical flow calculation is again utilized to compute RST transforms for virtual objects but more examples are given that illustrate the richness of possible interactions. For example, users may rotate an object by placing, and rotating, several fingers or an entire hand onto the center of the virtual object or by placing two hands on two opposing edges of a page and move them about the center of rotation.

ShapeTouch [CWB+08] explores interactions that exploit shape information. Fluid manipulation of 2D objects is supported by flow-based motion estimation, and a number of new behaviors based on contact size are presented. The concept of virtual force is introduced where size of the surface contact is mapped to amount of force exerted onto virtual objects. For example, one fingertip has a smaller footprint than the entire hand and therefore a fingertip would only exert a small force while the palm would exert a large virtual force. This concept is utilized to implement a number of interaction techniques. For example, the flat of the hand may be used to scroll a document, while an index finger may be used to perform ink annotations.

The interaction techniques discussed in this section serve as inspiration for our model for tabletop interaction which we debut in Chapter 7. Especially the idea to utilize all available information sensed by the hardware such as shape or proximity rather than relying on detection and tracking of discreet contact points seems to be a promising approach in order to enable richer interactions with virtual objects while drawing upon users' everyday experiences of interacting with physical objects.

2.2.3 Tangible Interaction

On regular tables we often store, view and manipulate a variety of objects such as paper, books or cups. Given their natural support for physical objects it is not surprising that the integration of physical objects (sometimes in combination with direct-touch input) as mediators for human-computer interaction has been explored in various forms. Early work on integrating the physical with the digital has coined the terms **graspable** and **tangible user interfaces (TUI)** [FIB95, IU97]. In this body of research the tight integration of physical artifacts into the information manipulation vocabulary for controlling, organizing, manipulating and sometimes representing digital information is explored. The main motivation is to make use of the affordances of physical objects, hence exploiting motor skills and manual dexterity as well as the expressiveness of these objects. It is argued that using physical handles to digital information unlocks a richer interaction vocabulary than standard input paradigms, and also is more coherent with mental models of everyday objects formed through live-long learning.

Inspired by Fitzmaurice's seminal work [FIB95] a raft of tangible user interfaces in different forms and application domains have been presented. Some TUI examples are literal instantiations of metaphors [UI97, UI99b] where the physical and the digital are tightly coupled. Other variations allow for more generic mixed physical and graphical interaction [RUO01]. Often, uses of the tangible paradigm are motivated by the goal to support co-located collaboration, for example **TViews** [MRD06], and **URP** [UI99b].

The variety in the presented approaches has in turn motivated several frameworks that describe and categorize TUIs by their functionality [UI00], and explore possible types of coupling and representation [HRL99]. These mappings and structuring of the design space have provided designers with useful tools to make decisions when creating tangible interfaces. For example, Benford et al. [BSK+05] describe the relationship between movements that are naturally carried out by users, movements that can be sensed and movements that are useful in particular applications, especially with regards to how sensing and movements do or do not overlap. Fishkin [Fis04] provides a framework for the structural analysis of tangible interaction along the dimensions of metaphor and embodiment. Hornecker and Buur [HB06] extend previous work by considering not only the aspects of the interface but also the emotional and social aspects of the interaction experience itself. These existing frameworks provide excellent concepts and tools to analyze and map out the design space as well as to characterize systems and to discover uncharted space. In the remainder of this section we discuss several tangible user interfaces using a categorization most similar to the frameworks proposed by Holmquist [HRL99] and Fishkin [Fis04]. However, it is important to remember that the used categories *tools, containers* and *navigation handles* are somewhat fuzzy and some TUI instantiations fit on more than one category.

Tools

Tools are physical objects that can be used to manipulate, edit or navigate digital information. Objects that mediate interaction from the physical domain into the virtual by altering the state of the digital.

Fitzmaurice, Ishii and Buxton [FIB95] first introduce the concept of **graspable interfaces**. Physical entities that can be used as input devices but have certain distinctions from regular input devices such as mice and keyboards. The input and output is directly spatially coupled – that is "**bricks**" are positioned directly on virtual objects, displayed on an interactive surface, and can be used to change parameters thereof. Several mock-ups illustrate how bricks may be used to select on-screen objects. Multiple bricks may used to change appearance parameters of individual objects, such as scale or color, or to move several objects at once. Some of these concepts are incorporated into **GraspDraw** a drawing application running on the **ActiveDesk**, a back-projected interactive table. Magnetic field tracking (Ascension Bird) is used to sense two active bricks.

Similar concepts have been explored in various application domains. The Urban Planner Workbench (**URP**) allows users to control several parameters in an urban planning simulation. Several physical objects are tracked by a vision-based system using an optical tagging scheme. The system used two classes of objects. First, tangible representations of buildings can be freely positioned on a top-projected tabletop. Simulation data such as shadows cast by these objects are projected onto their respective location, corrected for position and orientation of the object as well as for system wide parameters (e.g., time of day). Second, a set of abstract tools may be used to manipulate these system-wide settings. A clock is used to set the time of day, a inverse weather vane controls wind direction and strength in an airflow simulation. The demo application is followed by a discussion of different roles an object can assume in tangible tabletop interaction ranging from *pure object* (a real-world object) to *object as reconfigurable tool*. An interesting observation was that the presented prototype and others based on the combination of tangibles and tabletop displays are, by their nature, strongly engaging and it is argued that the unique combination of physical and digital is a cognitively powerful and intriguing one. This is a common and recurring argument throughout the literature. Other intrinsic benefits claimed for this approach are intuitiveness [IU97], motor memory [KHT06], learnability [RMB+98] and the possibilities of conveying the rich meanings in social settings [HB06].

In Ullmer and Ishii's **MetaDesk** [UI97] the idea of symbolic tools is taken a step further where physical objects embody the information they manipulate in their shape. For example, a plastic model of MIT's great dome - once positioned on a campus map - serves as control over the map's translation and orientation. In a similar vein **IlluminatingClay** [PRI02] and **Sandscape** [IRP+04] allow people to interact with real clay (or sand) whose shape is tracked and used to form virtual 3D imagery. Here the input "device" also serves as projection screen almost eliminating the distinction between the two concepts.

DataTiles [RUO01] is another example for tangible tools, albeit here the effect of the tool is constrained locally. Only virtual objects that are directly underneath or next to one of the transparent tiles can be manipulated and configured by the function embodied in the tile. Similarly to graphical filters, these tiles allow for different visualizations and manipulations of data displayed underneath. For example, users may fast-forward a video using a tile with an embedded dial. Several tiles can be combined to form more complex manipulations.

TViews [MRD06] is a generic platform to create applications using tangible devices or "pucks" as input to digital media applications on interactive tables. Examples are typically aimed at everyday social activities and domestic environments. In *PhotoSorting* the physical pucks are used to select photos and move them around the surface. *MapBrowser* allows to arrange and navigate photo collections enhanced with geo-information based on their capture location. **TViews** is also notable as it is one of the first examples of an deployment of an interactive tabletop outside the lab [MRD07]. In this case one unit was left at the home of one participant who used the device to share his own photo collection with visiting friends and family.

In **TViews** tangible tools are the only means of input to the system. Recently several hardware advancements have blurred the line between tangible and direct-touch interaction. In **PlayAnywhere** [Wil05] a vision-based system is discussed, capable of tracking multiple hands and sheets of paper and special objects identified by a visual barcode scheme. These barcodes allow the identification and tracking of objects (position and orientation) which allows for the usage of arbitrary objects as tools in applications. For example, audio CDs may be enhanced with such markers and used to play the respective audio files when detected by the system.

The **Reactable** by Jordà et al. is a combined platform for direct-touch and tangible interaction on an interactive tabletop. Plastic objects tagged with another barcode scheme are tracked using a vision-based system with a camera mounted underneath the back-projected screen (cf. 6.16). These pucks are used as input to a dynamic, modular synthesizer. Each pucks presence, location, orientation and spatial relationship to other pucks controls different parameters used to generate electronic music. By re-arranging the pucks several users may casually interact with the system and compose music on the fly.

Containers

Tangible containers are defined by Holmquist [HRL99] as "generic objects which can be associated with any type of digital information". Usually containers are objects designed to transport information from one place to another while hiding underlying complexities such as network connectivity.

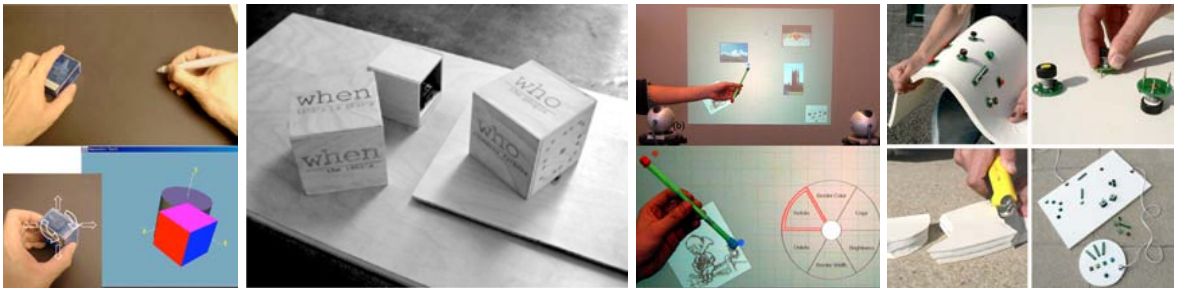
An early example for this class of tangibles as described by Crampton Smith [CS95] is Durell Bishop's mock-up of the **Marble answering machine**. In this concept each marble represents one voice message. Messages can be played back by placing one of the marbles into a special slot.

mediaBlocks [UI99a] is a tangible interface for physically capturing and transporting online digital media. Wooden blocks embedded with a digital ID can interface with various devices such as video cameras, displays and printers allowing digital media to be copied or transported from one device to another. Documents may be copied from a display (augmented with a reader for these blocks) then, physically, carried over to a large projection screen to discuss the contents and finally be dropped off at a printer to retain hard copies.

Besides the tool functionality in **DataTiles** [RUO01] tiles may also function as containers. Since the tiles can be recognized and identified by the system, information can be dragged onto tiles where it is stored until dropped off at another location. A variation of this concept are *portal tiles*; tiles that represent real world objects, devices or even people. For example, information may be sent to a printer, a camera stream may be accessed or an e-mail sent all by placing the respective tiles and dragging information to or from the tiles.

Navigation Handles

The last class of tangible devices are objects that serve as handles for navigation of digital content, in many cases exploiting the 3D nature of physical objects, hence increasing interaction expressivity and reducing the complexity when performing abstract commands or complex interactions with digital media. The reader should bear in mind that due to the scope of this thesis this discussion has to be limited to tangible input in combination with interactive surfaces. There is



(a) (b) (c) (d)

Figure 2.1: (a) Rekimoto's Toolstone. Top: bi-manual input; non-dominant hand operates ToolStone while dominant hand operates pen. Bottom: Using the ToolStone for 3D manipulation. (b) Navigational Blocks: Different Blocks are used to navigate through a large geographical data base based on "who", "what", "when" categories. (c) Top: VisionWand setup; two webcams track Wand in 3D in order to interact with large displays. Bottom: Performing complex commands from a distance. (d) VoodooIO: a flexible network substrate can be used to transform everyday surfaces into control areas. Various modular controls can be used to assemble and (re-)configure interface controls. The substrate can be cut and interconnected.

an extensive body of literature describing tangible input in spatial augmented and virtual reality and other fields that can not be discussed here.

Rekimoto and Sciammarella present the **ToolStone** [RS00] a self-contained tangible high DOF input device capable of sensing orientation in 3D (Figure 2.1 a). This device is used to show-case several bi-manual interaction techniques where the dominant hand operates a pen on a touch enabled surface and the ToolStone is used to manipulate a tool palette or the color chooser in a painting application. Other interaction techniques are demonstrated that directly map the 3D rotation of the device to parameters of a virtual 3D object such as orientation and zoom. Finally, a method to control a virtual camera in 3D has been implemented. It is argued that the passive haptic feedback provided by the device makes it easier to operate secondary functions in the interface (mode-switches, tool-parameters) without visual attention and thus decreasing cognitive load and reducing the necessity to interrupt the current workflow. However, no experimental evidence is provided but left for future work.

Camarata et al. [CDJG02] follow up on this basic idea and present various **Navigational Blocks**. Wooden blocks with embedded sensors for orientation and position measurement. These blocks are used to navigate tourist information at interactive info points. Each cube is marked with a token ("who", "what" and "when") and when placed on a special active platform this triggers a query to the attached display (Figure 2.1 b). The respective information can be navigated by moving the block around the active platform. also embedded in the blocks are elector-magnets that can be used to communicate relations between different concepts. When two or blocks are placed on the active platform they will either repel or attract each other based on availability of related topics in the database.

Cao and Balakrishnan present the **Vision Wand** [CB03] a passive wand tracked in 3D by two ordinary webcams. 3D position and orientation are used to recognize a complex gesture and posture vocabulary for interaction with large displays (see Figure 2.1 c). A series of direct manipulation gestures shows how the wand can be used to move digital photos, scale and rotate them by performing equivalent gestures. In addition, the wand may be used to invoke several abstract commands. For example, a gesture is introduced that triggers the display of a pie-menu. Rotating the wand afterward selects menu entries. In another complex command on-screen objects may be selected by pointing with one end of the wand. Pulling back from the display brings up a query lens, objects within the radius display additional information in a properties tooltip. The radius of the query lens can be adjusted by moving the wand to and from the display. An informal study revealed that participants had no trouble learning and memorizing the complete gesture set. Also the mapping of distance from the display to various parameters was reportedly intuitive. An interesting finding was that different users reported different postures and gestures as most comfortable which might be an issue when designing such systems. The authors also highlight that more complex gesture sets might be harder to memorize and that the continuous recognition mode might cause spurious input (aka "clutching error") when users perform ordinary movements not meant as command.

Voodoo IO [VG07] is an application development framework and toolkit for the rapid development of tangible applications. Rather than conventional input devices with a rigid shape and configuration the authors present a "malleable control structure for softwired interfaces". A networked substrate allows for the physical addition and reconfiguration of control elements to a special soft control area, potentially enabling any surface to be turned into a control area by simply plugging controls such as buttons or sliders into the surface (Figure 2.1 d). Several example applications are discussed for example, extending standard input devices in order to gain dedicated controls for specialty functionality in a gaming scenario (cf. [VGREG06]) or controlling a geo-information system on a large horizontal whiteboard enhanced with movable custom controls.

An often repeated argument in favor of tangible input devices is the assumed reduction in cognitive load and possibility to exploit motor-skills and motor-memory for eyes-free interaction. However, in over a decade of TUI research it has proved difficult to experimentally show benefits of the TUI paradigm in regards to efficiency or reduction of cognitive load. In **physical handles at the interactive surface** Terrenghi et al. [TKR+08] specifically investigate whether physical handles for information navigation have measurable benefits over a pure direct-touch approach. Two versions of a photo-browsing and organizing application have been explored in a comparative study. No significant differences in efficiency could be found but a couple of different interaction styles were observed further suggesting that tangibles might have intrinsic qualities that go beyond efficiency. The discussion echoes Rekimoto's assumption that TUIs are particularly engaging and foster curiosity on the users side.

2.3 Summary

Two major interaction styles have surfaced over recent years. Gesture-based and tangible interaction. Since in a typical tabletop setting no keyboards or mice are available performing complex commands can be challenging it seems promising to use gestures for command invocation. Many interesting interaction techniques and applications based on this principle have been demonstrated. Gesture-based systems allow for an immediate interaction, the possibility to directly touch information and spatial coupling of input and output make this interaction paradigm a promising candidate to the special design constraints imposed by the tabletop characteristics.

Tangible interaction promises to design richer interactions making use of manual dexterity and motor skills. The 3D nature of many physical objects affords specific ways of usage and as such helps to mitigate learnability and long-term memorization issues. Many researchers have speculated that motor memory would bear efficancy benefits over other interaction styles. Finally many positive aspects for embodied interaction and social interaction (e.g., access allocation, shareability) have been accredited to tangible interaction.

TUI and gesture-based interaction have often been studied in isolation even though they share many characteristics. To better understand the differences and commonalities of these two approaches we discuss explorations into both interaction paradigms in the following Chapters. We then analyze their suitability, strengths and shortcomings for tabletop interaction. Based on this analysis we suggest a new model for tabletop interaction that allows for richer human computer interaction more akin to the flexible and continuous ways in which we manipulate objects in the real world.

In this vein we explore a new model for tabletop interaction in the following chapters of this dissertation. Combining some of the positive aspects of *physicality* highlighted in the study of our initial prototypes with a interaction style that is more open-ended and *flexible* allowing users to make sense of the virtual world through (physical) exploration and experimentation.

Part II

Exploring Tabletop Interaction Styles

"Take a method and try it. If it fails, admit it frankly, and try another. But by all means, try something."

– Franklin D. Roosevelt

Drawing upon the analysis of related work in the field of digital tabletop systems presented in Chapter 2 this Part introduces our own initial explorations into tabletop interaction styles. Chapter 2 revealed two interaction paradigms that can be found in one or the other form across various proposed tabletop systems: gesture-based interaction and tangible[1] interaction. Both interaction paradigms promise appealing characteristics worthwhile exploring in the particular context of tabletop computing such as potentially more intuitive manipulations and a better integration between the digital and the physical space.

Often systems applying one of these interaction paradigms have been evaluated in formal user studies but within the context of their application domain and with respect to the aspects of particular interest to the underlying motivations. In other words evaluations have often focused on whether the design choices were sufficient or appropriate to efficiently complete particular tasks or interact with a single system. In this Chapter we perform a meta analysis of two systems we have built and evaluated – our original evaluations were designed to answer similar questions, for example "do our design choices work in this particular context?" and are detailed in the respective publications.

In this Part we want to take a step back and use the chance to revisit the results and observations we made during long hours of studying our prototypes. In order to be able to judge the potential of the interaction styles as generic paradigms for tabletop computing. Anticipating that interactions with digital tabletops will be different from standard computers, more ad-hoc, dynamic and also more embedded into the social interactions with other users we use *flexibility* and *physicality* as criteria for our analysis. *Flexibility* refers to the openness of the interaction; meaning how well does the interface lend itself for experimentation on the users side? How easy is it to discover functionality through simple experimentation and application of real world knowledge? How well can systems based on this interaction style be designed for appropriation? *Physicality* refers to the level of realism in the behavior of interface elements and how well the interactions exploit our fine-grained motor skills and manual dexterity.

Our insights from these explorations serve then as input for a new model for tabletop interaction discussed in detail in Part III of this dissertation.

[1] in the context of tabletop systems the TUI approach is often combined with direct-touch interaction; we refer to interfaces like this as *hybrid* interfaces

Chapter 3

Gesture based Interaction on Tabletops

Our discussion in Chapter 2 highlighted a variety of tabletop technologies and applications that have been proposed in recent years. Clearly this new class of devices has caught the imagination of researchers and practitioners alike leading to a raft of experimentations with and applications for digital tabletops. Although as of now it is not clear what the model of interaction for tabletop computing is, early research in the field has shown that tabletop interfaces are subject to different design constraints than traditional GUIs. For example, one of the appealing characteristics of digital tabletops is their natural support for co-located collaboration – users can sit around the table and maintain eye-contact while they interact with the computer. When considering that each user has a unique viewpoint it becomes immediately clear that many WIMP concepts such as menus and buttons at fixed locations become a hurdle for simultaneous interaction with the system. Furthermore, orientation of on-screen objects plays an important role for text readability and for group collaboration, interaction and coordination [KCSG03, KCST05] (*orientation* problem).

Another appealing quality of tabletop interaction is the possibility to directly touch information in a way often commented on as being natural. While it has been shown that direct-touch interaction can be effective in pointing tasks [SS91] (for bigger targets) direct-touch interaction can be problematic when interacting with typical (smaller) elements often found in traditional interfaces such as menus, buttons or sliders. This is mostly caused by two problems; the users finger occludes the region of interest in the critical moment before touching the display and the finger's size is many times larger than individual pixels which makes it difficult to accurately select small targets such as buttons (*occlusion* and *fat-finger* problem [WFB+07]).

The absence of keyboards and mice from tabletop setups together with the orientation, occlusion and fat-finger problems have lead to experimentations with alternative interaction paradigms for digital tabletops. Gesture-based systems allow for an immediate interaction, the possibility to directly touch information and spatial coupling of input and output make this interaction paradigm a promising candidate to the special design constraints imposed by the tabletop characteristics.

Of course “gesture-based interfaces” is a loosely defined term and can be applied to many different approaches. In the context of this thesis we focus on finger or pen-trace gesture interfaces as discussed in Section 2.2.2. Pen-trace gesture interfaces usually require some form of gesture recognizer that interprets geometric shapes drawn by the user with a finger or a stylus. These interfaces are often accredited with three main advantages:

Discoverability because gestures can be designed so that they resemble an activity or metaphor from the real world they are believed to be particular easy to learn.

Ease of use for similar reasons it is often claimed that gestures are easier to operate and use than interfaces based on abstract menus, buttons or textual commands.

Visibility of action given appropriate visual feedback gesture-based interfaces can provide users with awareness of their own actions due to tight spatial coupling of input and output and also provide group awareness in collaborative settings. In the latter case users’ interaction with the interface helps to reveal to one another what the task at hand is, and also helps collaborators to understand which parts of the information are presently being inspected (“look at this”).

In the following sections we discuss our exploration into gesture-based interfaces for interactive surface computing. In order to gain a better understanding of this interaction paradigm we built and studied a prototypical application – called **Brainstorm** – for collaborative problem solving in an environment enhanced with several interactive surfaces. The system uses gestures to invoke almost every single command in the system and we carefully designed the interface to address the orientation and occlusion problem. We designed several gestures ranging from literal real world metaphors to more abstract gesture to interact with a graphical user interface. the UI itself draws many clues and inspirations from the real world to increase discoverability and ease of use. We discuss our design choices and present results from a lab based user study. Finally we reflect on our observations from many hours of formal evaluation and informal system use.

3.1 Brainstorm: A Case Study

Brainstorm is a system built using several interactive surfaces including a digital tabletop and several wall-mounted displays. The system is intended for supporting co-located collaborative problem solving. Collaborative problem solving requires knowledge and information to be exchanged among team members; different skills have to be coordinated and the information communicated by others needs interpretation, so that new ideas can be created and new solutions can be found. This process - with its core requirements of communication, coordination, and interpretation - is called collaborative creative problem solving [Ama96].

The design of the **Brainstorm** socio-technical system is meant to explore the possibility of merging the physical and social qualities of a traditional face-to-face collaborative creative environment together with some of the benefits of digital technology, such as persistent data storage, distributed information access, and the possibility to review previous processes or to undo certain actions. Thus, this section discusses the design challenge of embedding digital technology in a collaborative creative process without causing communication breakdowns, while still taking advantage of some of the qualities of Electronic Brainstorming Systems (EBS).

The issue of more fluid interaction with large, high-resolution displays that support creative group processes has been treated in [GSW01, KNF$^+$01]. A summary of recent advances in the field of interaction techniques for large displays can be found in Czerwinski et al. [CRM$^+$06]. Most of this research has focused either on the properties and design implications for vertical large displays [RCB$^+$05] or on the influence of horizontal displays on co-located collaborative work [SGM03].

Probably most related to our work, especially with regards to our attempts at blurring the line between the physical and the digital, is the **Designers' Outpost** by Klemmer et al. [KNF$^+$01]. The **Designers' Outpost** is a support tool for web designers that combines the benefits of a large, vertical workspace, the flexibility of physical media (post-its) and the benefits of electronic media to annotate and structure information. A SmartBoard [SMA03] is augmented with a computer vision system to capture physical media that can be attached to the display and to capture interaction with tangible interface elements such as a physical rubber and a move tool. In many ways this system served as inspiration for the **Brainstorm** system discussed in this chapter, only that we were interested in building a purely virtual interface, fully harnessing the benefits of the digital. Of course one of the design goals for **Brainstorm** was to mimic the *flexibility* afforded by lightweight physical artifacts such as post-it notes. However, in Section 3.5 we shall learn that we only succeeded partially.

> **Contribution Statement:** Most of the work discussed in this Section has been published in [HTB$^+$07]. I am the first author on that publication and I have initiated and lead the project. As with any other scientific publication all of my co-authors have contributed significantly. Sebastian Boring has built and implemented many aspects of the interactive wall display used in this project [BHB07]. David Kim has contributed to the implementation and user study. All other authors have had significant input on the design of the system and the study as well as on writing the paper.

3.2 A Side Note on Brainstorming

Brainstorming is a technique for divergent thinking. It can be individual, although the term more usually refers to a group process for generating as many ideas or options as possible in response to an open question. Thus, it is frequently used for collaborative creative problem solving and it builds on a few main principles: i.e., quantity over quality of ideas, elaboration on others' ideas, and absence of criticism [Osb53]. The technique relies on the communication among group members to stimulate idea generation, and on coordination to maximize the individuals' involvement and interpretation of ideas in order to create new intellectual associations, i.e., to increase the production of ideas.

Although Osborn [Osb53] claimed synergy effects of brainstorming, which positively affect the productivity of ideas, other studies (e.g., [DM87], [DR04]) have shown that these benefits are apparently outweighed by several negative social implications of the technique, such as apprehension of social judgments in face-to-face conditions. Those studies claim that nominal brainstorming groups (aggregating ideas from separate individuals) outperform face-to-face groups. In this context, EBSs that support distributed collaborators have been successful in increasing productivity of ideas, apparently because they allow for anonymity, which reduces evaluation apprehension.

On the other hand, face-to-face collaborative creative problem solving is still common in practice, and its value is probably not to be associated with the number of generated ideas only. The individuals' subjective perception of the outcome of the process plays indeed an important role as well, and depends on the degree to which personal interests are represented and valued in the group's output. Furthermore, the face-to-face brainstorming situation has qualities which, in the long run, might even outweigh pure productivity measurements, namely the positive social aspects of team building, group awareness, and a shared sense of achievement.

In such contexts of face-to-face brainstorming, EBSs seem to perform poorly in comparison to nominal (i.e., distributed) brainstorming settings because of their disruptive effect. Using single-user systems, such as laptops, in a co-located collaborative setting leads, in most cases, to a communication breakdown since the user's concentration has to shift away from the group toward the device in order to use it. Furthermore, the size of a personal computer screen or the keyboard, as well as the turn taking implied by devices for single usage, seem to hinder the communication process.

For this reason, in face-to-face contexts digital technology is often absent or shut down because it results disruptive of the group communication and of the "creative flow" [Csi96]. Therefore, instead of relying on digital technologies, co-located creative meetings commonly rely on the physical benefits (e.g., gathering around a shared space) and social benefits (e.g., having equal access to information) afforded by physical surfaces, such as tables and walls, to exchange and visualize different types of information artifacts (e.g., post-its, paper documents, pictures, etc.) in an immediate way. The results of such processes are often turned into a digital format in the end, by taking pictures of whiteboards and posters, or typing notes in digital documents in order to distribute and archive those results. Several transitions from physical to digital media occur which require additional work, and are mostly unable to capture and represent how the creative process has unfolded in time.

Considering this trade-off between the social and physical benefits of face-to-face collaboration, and the benefits of storage in digital format provided by EBSs, the design of the **Brainstorm** electronic system tries to combine those benefits in order to:

- maintain the social benefits of face-to-face collaborative creativity: These depend on the individuals' subjective perception of the group process.
- exploit the benefits of EBSs which are normally recognized in distributed collaboration, such as the capability to archive and easily review the collaborative process in different locations and points in time.

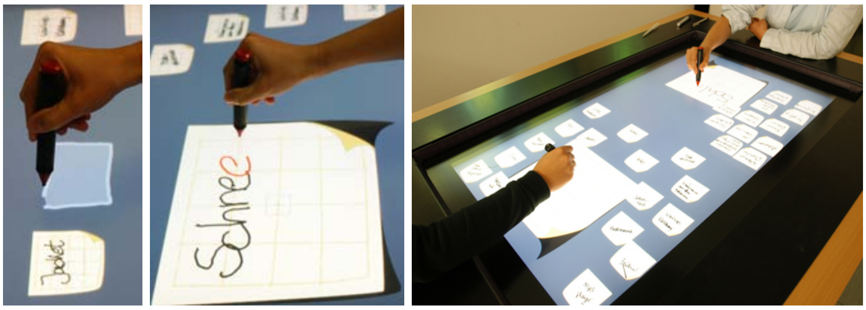
- explore how such a combination can affect creative collaborative patterns in terms of generation and organization of ideas, communication, and subjective experience.

3.3 Designing the Brainstorm System

Building on these considerations we have developed a system that spans across several interactive displays embedded into tables and walls. The displays used in this system are commercially available Smartboards™in case of the table and the center display of the interactive wall (see Figure 3.1). The wall display is extended by two additional, back-projected displays. In addition to the contact reporting by the Smartboard the entire wall is turned into an interactive surface by a custom built finger tracking system [BHB07]. This allows an almost seamless information transfer from one display to another. The utilized Smartboards sensing is limited to two simultaneous points of contact. Therefore, our system was designed with only two users in mind.

Figure 3.1: Overview of the physical setup in Brainstorm. Smartboards embedded into the table and wall function as primary input surfaces. In the periphery additional interactive displays can be used to store information not currently used. A custom user interface has been implemented mimicking paper-based brainstorming but enhancing the technique through additional functionality invoked through gesture-based interaction.

The design of the user interface builds metaphorically on paper post-its, on the ways in which they are socially used, as well as on their manipulation vocabulary in the physical world in order to suggest ways in which ideas can be generated and manipulated as information units in the EBS appliance. Studies of paper in work practice, in fact, show that paper continues to be widely used [SH03], some of the reasons including its spatial flexibility (it can be quickly arranged in the physical space), sociability (it facilitates face-to-face communication by being passed on), and tailorability (it is easily annotated). Post-its, in particular, are commonly used in the the idea card method for brainwriting, which is based on Geschka's [Ges78] and Van Gundy's [VG88] "Interactive Brainwriting Pool Technique". In this method, group members write their ideas on a piece of paper that is then placed in the center of the table for another member to read prior to writing their next comment. In this way, post-its afford the externalization and record of ideas in written rather than just verbal form in the generative phase. Furthermore, they support a certain territoriality and the creation of semantic regions. When participants are given a stack of post-its and start sticking them around their working area, they define their personal region, which remains visible to others, thus creating a mutual awareness among participants. Using post-its on vertical surfaces supports the convergent thinking phase too, when group members stick and move post-its on flip charts or whiteboards in order to recognize relationships and create clusters.



(a) (b) (c)

Figure 3.2: (a) Creating a post-it by drawing a rectangular closed path. (b) Writing on an enlarged post-it. Handwriting is recognized and stored for later retrieval. (c) Two users interacting with the system at the same time.

Drawing upon these considerations, the **Brainstorm** interface was designed around the paper-based brainwriting technique. Users can start creating ideas by drawing a square on the table surface.The system provides immediate visual feedback by rendering the recognized path and a prediction of the area enclosed by the user-drawn path (see Figure 3.2 (a)). Upon recognition this gesture triggers the appearance of a large yellow square, resembling a post-it, thus defining the area to write in (Figure 3.2 (b)). By tapping a designated area marked as a small square in the center of the post-it, the latter shrinks to a smaller size and becomes movable.

The user can then create new post-its/ideas by drawing new squares in a blank region of the table and writing within the yellow region. This choice was made in order to create visual constraints for writing, so as to identify ideas as units, and to create visual cues for distinguishing territories and patterns. When the post-it is shrunk, its content is still readable. Such graphical post-its can be edited, moved, deleted, and copied by any participant after they have been created. Furthermore both users can create and edited post-its simultaneously (Figure 3.2 (c)). To delete a post-it, this can be dragged to the edge of the screen till it disappears. To copy a post-it, it can be

(a) (b) (c)

Figure 3.3: Flicking of virtual post-its across the table. (a) Post-its have varying velocity depending on speed and impulse strength of dragging gesture. (b) Sudden increase in velocity causes post-its to accelerate according to the user generated impulse. (c) This interaction technique can be used to transfer information across a distance greater than arms reach.

virtually flipped by tapping on its plied bottom right corner; then, it can be duplicated by tapping on the “copy” icon displayed on its back side.

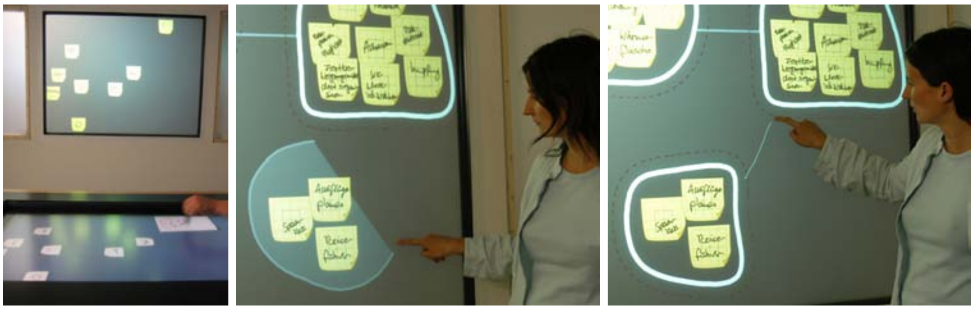
Additionally, a mechanism to encourage users to build on each other’s ideas was implemented: post-its have a “sense” of velocity and inertia. When dragged around the table at slow speeds they follow the point of contact directly. But users can send the post-its sliding across the table by performing a quick flick of the wrist. Causing the post-it to accelerate and slide across the table (see Figure 3.3. Once it approaches the opposing edge of the table it smoothly reorients itself toward the other user: This was intended to support the explicit sharing of ideas so as to encourage the creation of association chains.

To this end, the design of **Brainstorm** implements a limited manipulation vocabulary and relies on simple marking gestures for direct manipulation such as drawing a square, writing text, and stroking for moving, whose direct feedback is augmented by the coincident spatial mapping of input and output (i.e., there is no such device as a pointer or a remote controller). This creates a transparent causal relationship between gestures and output, and supports visibility of gestures. Furthermore, by simply dragging the post-its, temporal spatial arrangements can be created.

The immediate and visible change of the shared visual landscape is supported by the system in additional ways. As the participants create post-its in their working areas, thus already creating a distinct territorial set-up, the post-its appear simultaneously on the vertical display, which is located next to the table. On the vertical display, the post-its are reoriented upright, i.e., readable for both readers, but they maintain a spatial mapping to the territorial set-up on the table display (Figure 3.4 (a)). In this sense, the perception of territoriality and group awareness are supported: A participant will recognize his/her own “territory” (i.e., contribution) on the wall, and at the same time will gain an overview of the ideas created by the group.

When users move from the interactive table to the wall display, they can spatially organize the ideas which were automatically displayed by rearranging them on the wall. In addition, they can create clusters by drawing a circle around several post-its (Figure 3.4 (b)).

Clusters are merged by dragging them close together. Drawing a cross on the border of a cluster causes it to dissolve into single post-its again. Clusters can be connected to each other or to single post-its by drawing a line from the border of one cluster to the border of another one or to the center of a post-it (Figure 3.4 (c)). Finally, whole clusters can be moved across the display, thus moving all the post-its they contain. This set of clustering techniques clearly extends the functionality of a physical whiteboard or flip chart while it maintains the direct manipulation characteristics thereof. This aspect, in turn, can facilitate the creation of a structured knowledge representation (e.g., a mind map) easily editable by every participant.



(a) (b) (c)

Figure 3.4: Interaction across displays: (a) Post-its created on the table appear spatially mapped on the wall display. (b) Creating new clusters by drawing circular enclosure around several post-its. (c) Connecting clusters by drawing connecting lines. Clusters can also be annotated with additional post-it notes.

3.4 Evaluation

In order to assess these design choices with respect to their underlying motivations, the **Brainstorm** EBS was compared to the paper-based brainstorming process. The detailed results on the comparative study are published in [HTB+07]. Here the goal was to understand the implications of blending such an EBS in the physical space, in comparison to a traditional paper-based brainstorming technique. However, in the context of this thesis we will focus on discussing results and observations relevant to the analysis and assessment of the chosen gesture-based approach as interaction style for tabletop computing. Therefore the discussion in Section 3.5 focuses more on the qualitative observations made during the study. Aspects of particular interest were:

- the interface with the **Brainstorm** system at a pragmatic level and whether the chosen interaction elements allowed for a fluid and continuous interaction when generating ideas, discussing ideas with others and spatially organizing ideas. We also highlight commonalities and differences in respect to the interactions in the real world (i.e., the paper based condition)

- the discoverability of the interface elements or how much instructions the participants needed in order to master the interface initially.

- the memorability of the interface elements or how well participants could remember gestures after extended use of the system. We discuss differences across different types of gestures – more abstract gestures (e.g., copying, deleting) – and more concrete or physically inspired gestures (e.g., flicking, creating post-its).

3.4.1 Technical Setup and Procedure

The system was deployed in the FLUIDUM instrumented environment, containing an interactive meeting table as well as displays embedded into an interactive wall. This wall consists of an interactive surface with a width of 5 meters and a height of 2.5 meters containing three back-projected displays. The two side displays as well as the rest of the wall are tracked by four cameras mounted in the four corners. The center display additionally provides high precision input through a DViT panel (cf. [BHB07] for more details). Two single points of contact can simultaneously be tracked both on the table and on the wall, thus enabling parallel interaction of two users.

We conducted a within-group comparative study between the **Brainstorm** EBS and the original paper-based brainstorming technique with 30 participants in 15 teams, dealing with two tasks each. The participants represented a variety of professional backgrounds (e.g., computer science students, architects, designers, civil engineers, musicians, as well as journalists), ages (between 20 and 50 years old), and nationalities (5 countries).

The participants had to complete one different task/problem per technique, the techniques were presented in counterbalanced order. In the first task, the teams were asked to take care of an Inuit coming to a foreign country neither speaking the country's language, nor having any useful equipment for the new environment. In the second task, the teams had to discuss their own needs when they would leave their home country for emigration into harsh, icy arctic territories. The subjects were asked to collect all material and immaterial items they would consider necessary for survival under these conditions. Taking into account the broad variance in participants' professional education, these tasks were chosen because they could be addressed without any domain specific knowledge. Furthermore, they were rather simple tasks to which everybody could relate, and thus contribute a significant number of ideas.

3.5 Observations and Implications

Adopting a qualitative approach, we ran questionnaires before and after the task to evaluate the subjective expectations, perception and assessment of the **Brainstorm** system (cf. [HTB+07]). Furthermore, the sessions were video taped for later analysis in order to recognize association chains and communication patterns.

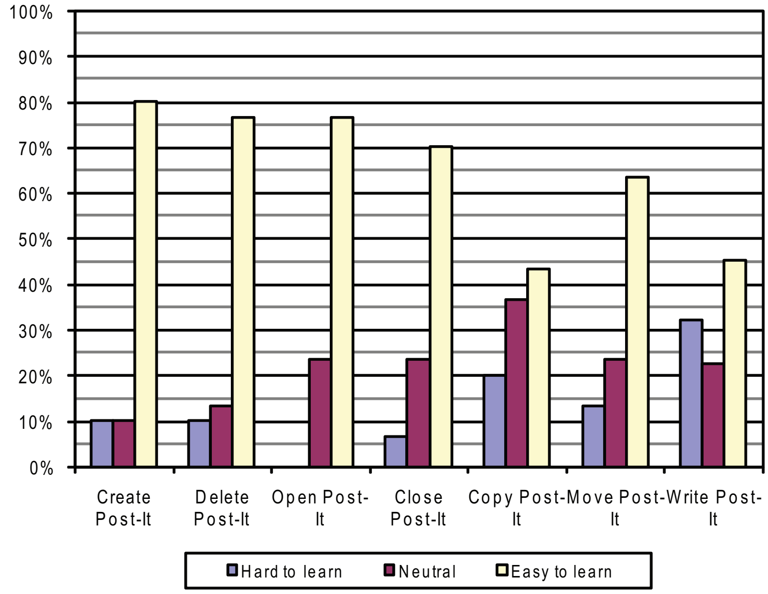


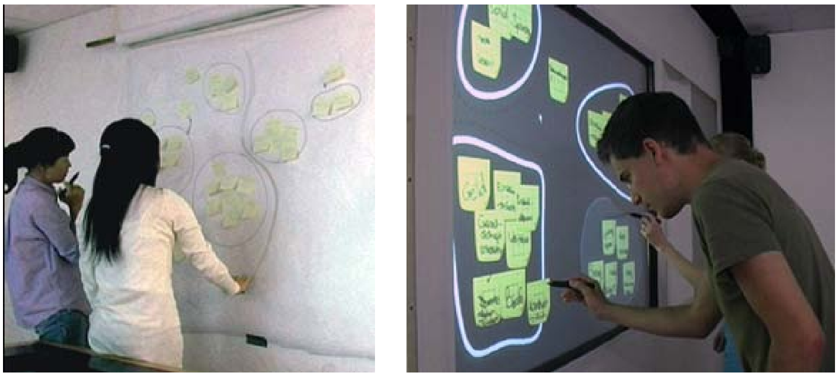
Figure 3.5: Subjective judgement on how hard/easy participants found particular gestures to learn in the Brainstorm system.

In the post-test questionnaire we asked the participants about the learnability of the implemented gestures. Figure 3.5 summarizes the results of users' ratings of the different gestures. Here we received consistently good results regarding all interactions. Only the hand-writing text-entry on the table received rather low ratings. We think that this might be due to the clumsy pen used, and the slightly unusual hand posture which was required in order not to confuse the DViT tracking system. The "copy post-it" interaction was also rated ambiguously, however this interaction is not really a gesture (it resembels more a traditional menu based interaction) so it is hard to compare against the other interactions. Participants stated that they found the interface of the system easy to learn. We credit this on the close resemblance of each interface element to the real world. For example, the interface contains visual elements that resemble the real world equivalents (e.g. post-it notes). These elements are also manipulated in similar ways as they would be in the real world (writing on paper with a pen). This allows users to build on knowledge they gathered from a life-long learning experience with the real world and the objects in it. Touch sensitive interfaces and fluid gestures make using the technology more continuous and analog. This allows users to apply strategies they already use in the real world to both implicitly and explicitly convey information about the objects in the environment (e.g., territoriality).

3.5.1 Physicality

Even if the close resemblance of virtual items to real world artifacts is beneficial to the ease of learning the interface, it is worth exploring the specific and different affordances of digital media. These can augment physical actions, providing effects which are only possible in the digital realm (e.g. the automatic re-orientation and appearance on the wall of virtual post-its).

As long as objects have a clearly distinct and explainable behavior, users seem to be willing to accept and use a technique even if it is unrealistic in the strict sense. For example, the participants reacted positively to the possibility to skid post-its across the table, even if this is not as easily achieved with real paper (due to its weight and friction). Although the system does not provide an accurate simulation of the physical world it clearly benefits from borrowing visual and behavioral metaphors from the real-world. However, this limited degree of *physicality* also possesses some limitations. These become apparent when paying close attention to the differences in interaction strategies applied by participants in the digital vs. the physical realm when organizing ideas into clusters in the structural phase of the brainstorming process.



(a) (b)

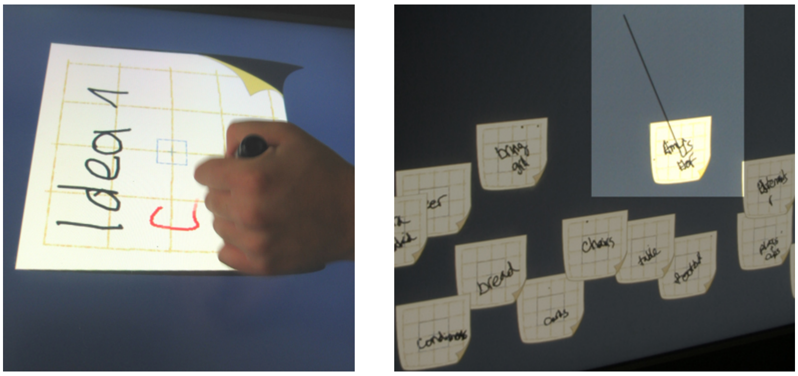
Figure 3.6: Applying different strategies to semantically sort post-its. (a) Background annotations; various stroke patterns and region (sub)-division in the paper condition. (b) Spatially arranging post-its to convey meaning in digital condition.

An obvious advantage of the digital is that many objects can be moved, copied or deleted at once and we could observe how participants made frequent use of this feature. In fact in the digital condition spatially arranging post-its was the predominant strategy and was preferred even to clustering. The other features such as connecting clusters and especially annotating clusters were only rarely used. In harsh contrast participants showed a variety of strategies to work around the more laborious task of physically moving clusters of post-its in the traditional conditions. For example, they labeled clusters, created sub-clusters or divided existing clusters into semantically disjunct areas by using differently stroked lines or background patterns (Figure 3.6). Of course these functionalities could be implemented in the digital as well but a fundamental differences exists: In general the users showed great aptitude in appropriating the available materials to work around the *absence of features* (e.g., copying, mass moving) by exploiting the physical properties of objects. For example, scribbling onto various parts of the background, annotating connecting lines or sticking several post-its on-top of each other.

3.5.2 Flexibility

One fundamental problem is that the behavior of on-screen elements appears to be physical or realistic only at first sight – and this helps in reducing the learning threshold – but is preprogrammed or scripted after all. The underlying problem here is that the user interface is subject to a limited set of pre-defined reactions to user input. Therefore, the designers and programmers had to anticipate each users needs at any given moment in any given situation. When observing the frequency and creativity with which participants made use of the affordances of simple tools such as pen and paper in the real world condition it becomes quite obvious that not all of these could have been anticipated in advance let alone the difficulties in sensing, recognizing and differentiating subtle changes in user interaction. In hindsight we would argue that the pseudo-physicality in the **Brainstorm** system was successful as measurement to lower the initial learning threshold but can be a limiting aspect in terms of *flexibility*.

The following aspects illustrate as to why and how this lack of flexibility can be a problem. While we have discussed benefits for learnability of our real-world inspired gesture-based interface this design choice raised false, sometimes problematic, users' expectations. This led to frustrations when the system did not respond as they expected based on their real-world knowledge. In the simplest cases this was a matter of gesture recognition; in the real world we have a variety of strategies to push objects around. We can touch them from the top and drag them or we can use one or more fingers to push objects from the side. **Brainstorm** only implemented the former and we frequently observed users trying to perform the latter – often repeating their attempts under growing frustration and comments that this "should work like this if it was the real thing". Clearly here a mismatch between implementation and mental model can be observed.



(a) (b)

Figure 3.7: Problems with explicit mode switching in Brainstorm: (a) Awkward posture for handwriting. Only one hand is used to write on paper. (b) Artifacts caused by the user trying to move the post-it while in writing mode.

In other cases the need to recognize certain user intentions and the limitations in the sensing technology caused similar problems. For example, input in **Brainstorm** happens through a single point of contact (per user) so that for many functionalities we had to design abstract or iconic gestures to switch modes. The most prominent example is how users have to switch between writing on post-its and moving them. A double tap within a designated area enlarges post-is and makes them writable but fixes them to their location, a second double tap shrinks them and makes them mobile again. We frequently observed how this mode switch caused problems (even after extended use). Often users would try to move objects when in writing mode, causing unwanted inking across the written text or users tried to quickly scribble something onto post-its without enlarging them first (see Figure 3.7). In the latter case users often tried to hold onto the post-it with the second hand to hold it in place for writing (before they remembered that they had to double-tap post-its to switch modes).

3.5.3 Summary

In summary we would argue that exploiting physical affordances in the virtual interface was successful to a certain degree, especially as a learning aid. However, some of our design choices turned out to be problematic. These can be categorized in two ways. First, over-simplifying Newtonian physics caused user frustration especially when they could not apply real world strategies to achieve seemingly simple things such as moving objects but had to perform a specific gesture in a particular, pre-defined way. Second, the resemblance of the interface to physical objects seemed to encourage users to interact with the system in more analogue, richer ways than just through a single point of contact. We observed bi-manual interactions and attempts to use multiple fingers at once to manipulate one object or even whole hands to move several objects at once.

From our exploration into gesture-based interaction we can distill the following key aspects to inform the next steps in our research endeavor:

- Real world resemblance and metaphoric gestures can lower the learning threshold. This interaction style could be appropriate for applications with limited feature sets and casual application domains so that problems with long term memorization are not as important.
- The need for gesture recognition (and therefore anticipation of user intentions) can be a limiting factor in terms of interaction flexibility. In the worst case this limitation can frustrate users and render a design in-successful.
- Using a pen or stylus in combination with pen-trace gestures does not mimic the rich ways of interaction we enjoy when manipulating real world objects sufficiently.
- Limiting input to a single point of contact can be problematic in some situations - especially if graphical elements resemble real world objects but can only be manipulated through a pointer based interaction model.

These observations are the main motivation for our next exploration into tangible objects in combination with interactive surfaces to further unlock our manual dexterity.

Chapter 4

Tangible and Hybrid Interaction on Tabletops

Tangible user interfaces (TUIs), inspired by the seminal work of Fitzmaurice et al. [FIB95] expand the interaction vocabulary by exploiting fine grained motor skills that humans possess when manipulating tangible objects. The main benefits claimed in this area of research are intuitiveness [IU97], motor memory [KHT06], learnability [RMB+98] and the possibilities of conveying the rich meanings in social settings [HB06]. Some TUI examples are literal instantiations of metaphors [UI97, UI99b] where the physical and the digital are tightly coupled. Other variations allow for more generic mixed physical and graphical interaction [RUO01]. Often uses of the tangible paradigm are motivated by the goal to support co-located collaboration, for example TViews [MRD06], and Urp.

Recent hardware advances have made it feasible to sense and identify tangible objects on tabletop displays. Wilson [Wil05] demonstrates a vision-based system capable of tracking physical objects through visual barcodes, hands and sheets of paper using IR-illumination and an off-axis camera equipped with an IR cutoff filter. A similar technique is used in the reacTable [JGAK07] to track objects that serve as input to a virtual musical instrument. This hybrid approach to interface design appears as particularly promising because of two reasons: First, direct-touch interaction alone is an intuitive and easy to learn interaction style. However, the bandwidth of input and the interaction vocabulary are limited in comparison to the flexibility we enjoy when interacting with the real world. Second, in the light of our analysis of a purely gesture-based approach (cf. 3) and recalling the user frustration caused by the limited degree of *physicality* in the purely virtual interface of the **Brainstorm** system using physical objects as handles for interaction appears as an interesting avenue for exploration.

In this Chapter we describe our exploration of hybrid interaction (i.e., tangible interaction combined with direct-touch interaction) using **Photohelix** [HBB07], a system designed for sharing personal photo collections on a digital table, as vehicle. We report findings from several user studies. At the end of this Chapter we furthermore discuss the suitability of this approach as a general interaction model for tabletop computing.

4.1 Motivation

Interactive surfaces and digital tabletops in particular, offer a compelling platform for shared display collaboration, allowing multiple users to interact simultaneously with a shared information landscape. These platforms provide the same (social) functions as traditional tables allowing multiple people to sit around the table and share artifacts on the surface of the table only with the difference that the artifacts can be digital. An often heard explanation is that digital tabletops afford mutual eye contact and body language as well as other properties important for verbal and non-verbal communication such as the possibility to interact with and exchange data artifacts through direct-touch interaction.

One of the main advantages of interactive surfaces is the flexibility of the interface; because it is purely virtual the interface can be dynamically reconfigured to serve different purposes be it for work or play. However, touch sensitive surfaces do not offer the same tactile feedback which traditional (non touchscreen) interfaces provide through physical buttons, knobs and switches. This feedback is important for motor learning and the automation of repetitive tasks such as touch typing on a QWERTY-keyboard (text entry is a task still notoriously difficult on direct-touch interfaces cf. [HHCC07]). Using physical controls in combination with interactive surfaces promises to combine the best of two worlds – dynamically reconfigurable graphical output coupled with the possibility to directly interact with digital content and at the same time physical handles for interaction. These promise to unlock the operation of an interface – or key elements thereof – without visual attention in a multitasking situation such as sharing of digital photos.

An understanding of the operation of user interface (UI) elements without looking at them (or only briefly) from an embodied cognition perspective [Cla00] would though suggest that the use of 3D elements at the interface offers a form of tactile feedback unavailable in a direct touch enabled UI. This tactile feedback should theoretically redistribute cognitive effort into other sensory modalities. So whereas with the direct-touch UI the cognitive effort is largely expended through visual attention, with the tangible interface some of the effort of control (i.e. feedback loops governing appropriate interface manipulation and understanding that you are still within the bounds of reasonable control movement) becomes tied to tactile interaction. We call this *eyes-free* manipulation. To give a simple analogy we could consider driving a car. This is a visually intensive activity, visually monitoring the road and traffic is mandatory. If for example, the controls for driving required additional visual monitoring such as in a car that was completely drive-by-wire and the control mechanisms were based entirely on a direct-touch UI system, then the car would become too dangerous to drive. Consequently cars have maintained physical analogue controls which provide tactile feedback of their manipulations (e.g. steering wheels and pedals) so that the visual resources are kept free for relevant activity.

A second argument for the usage of physical handles is the expressiveness and flexibility provided by real 3D objects as input devices. Designing tangible interfaces promises to exploit fine-grained motor skills and exploiting 3D shape and mechanical constraints to implicitly convey the correct usage of an artifact. For example, a well designed door handle communicates without further instruction how to operate it. In addition to their main purpose physical objects can be repurposed to serve other uses. For example, imagine the plethora of applications a simple screw driver can be put to (besides driving screws). Along those different uses come various ways how we grasp and manipulate physical objects depending on the current task. When fixing a screw we hold the screwdriver differently as when punching holes into the lid of a shoebox or when simply carrying the tool around. In general our motor skills allow for a variety of object manipulations ranging from fine-grained to coarse interactions. Direct-touch interfaces usually do not discriminate between different ways of touch, in contrast tangible objects naturally afford this richer interaction vocabulary.

In the following section we explore hybrid (e.g., tangible plus direct-touch) interfaces with special focus on the following aspects:

Physical affordances: We explore how physical handles on interactive surfaces can exploit the manipulation fidelity and tactile feedback afforded by tangible interfaces in order to create a richer interaction vocabulary.

Flexibility of interaction we're also interested in learning how users make sense of physical artifacts and how they exploit the physical properties to re-purpose the interface to serve their current needs.

Support for bi-manual interaction: In the real-world we often use both hands to manipulate various objects. Previous studies have accredited several cognitive and manual benefits to bi-manual input [LZB98]. Although interactive surfaces often allow for bi-manual input they do not seem to encourage it *per se* [TKSI07]. We are interested whether physical objects offer themselves for this particular interaction style.

Eyes-free manipulation: Again due to the tactile feedback provided by tangible objects we expect that physical handles can be used to operate elements of the interface in an *eyes-free* fashion freeing up resources for other aspects in multi-task contexts.

Contribution Statement: Two papers on **Photohelix**, the system serving as basis for discussion in this Chapter, have been published [HBB07, HK09]. I am the first author on both publication and I have initiated and lead the project(s). Domikus Baur has contributed significantly on the implementation of the original system. David Kirk has helped in designing the user study and evaluating the data in the second publication [HK09]. All other authors have had significant input on the design of the system and the study as well as on writing the papers.

4.2 Designing the PhotoHelix

We developed **Photohelix** (see Figure 4.1), an application tailored for co-located browsing and sharing of pictures on a digital tabletop (cf. [HBB07]). The system uses the notion of time and events to organize collections. Events are represented as image piles on a helix-shaped calendar (Figure 4.1 (b)). Events and pictures are accessed, manipulated and inspected using a hybrid,

(a) (b) (c)

Figure 4.1: (a) Photohelix overview: A physical handle is used to position and control a spiral-shaped calendar. Pictures are loosely grouped into events. (b) Users can (re-)arrange event structure as well as individual photos using a set of lightweight interaction techniques to facilitate dialogue about the photos. (c) Two users engaged in photo-talk using the Photohelix.

bi-manual interaction technique. One hand operates a physical handle to position and control the calendar view (rotation adjusts the current time setting). The other hand is used to inspect and modify events as well as individual pictures for browsing and sharing purposes.

The system was developed and deployed on a custom interactive table, which contains a 42-inch LCD display with a native resolution of 1360×768 pixels and an overlaid touch-sensitive DViT [SMA03] panel for interactivity.

To fashion the physical control object, we disassembled an IKEA kitchen timer and equipped it with the electronics of a wireless mouse to measure rotation. Turning the upper part of the control object results in a standard mouse event that translates to the rotation of the helix. The position of the control object on the table is tracked by the DViT panel (see Figure 4.1 (a + b)).

Photohelix was written in Java with a graphical presentation layer based on the University of Maryland's Piccolo framework [BGM04]. We wrote an additional event-handling system that merges and interprets rotary encoder and touch events. These events are fed into a gesture recognizer, which enables gesture-based interaction with, and manipulation of, the photo collection and individual pictures. Metadata for individual photos, such as the capture date, is taken from the EXIF data.

4.2.1 Visualization

Tightly coupled to the physical control object is its virtual counterpart, a graphical visualization of the photo collection. It has the shape of a spiral and represents a timeline, on which the photos are organized, according to their capture date. Initially, photos are grouped into piles if they belong to a temporally continuous sequence (see Figure 4.2 on the left). This gives users an overview of their collection and supports orientation within the collection by narrowing down the search space.

The position and rotation of the spiral are controlled by the physical control object, hence it serves as a natural token to facilitate control allocation and turn taking in face-to-face com-

Figure 4.2: A screenshot of Photohelix. The distinct functional areas (here: details above the helix, storage to the right) evolve dynamically and can be rearranged individually.

munication and as a physical embodiment of the entire collection. The timeline is dynamically generated and spans from the oldest image in the collection – placed in the center of the spiral, to the most recent image – placed at the outer end of the spiral. The inner spiral windings are shorter than the outer ones. This implies, that more space is available to place image piles in the outer, or newer, regions of the spiral. This nicely matches the observation, that people tend to take more photos with increased frequency over time. Furthermore, newer piles are depicted bigger and hence are easier to decipher. This also correlates with the observation that newer collection items are more frequently accessed than older ones [KSRW06].

Another component to **Photohelix**'s spiral-shaped timeline is a semi-transparent lens that is overlaid on a certain section of the spiral. Pictures and piles of pictures that fall under the lens are shown in more detail thus providing "details on demand" (see Figure 4.2 above the helix). **Photohelix** works in two organizational forms: spatial arrangement and semantic grouping. Pictures are either shown individually (but arranged chronologically) or as so-called events. Events denote a stronger, more semantical coherence of the images therein and have to be created by the user (see Figure 4.3). Events are similar to folders in standard file managers. Each picture or event, when it falls under the lens, is called out and enlarged. It remains connected to the respective pile on the helix by an "umbilical cord". These images are again arranged chronologically along an imaginary line that runs parallel to the spiral's timeline. This leaves temporal relations intact and, in most cases, is equivalent to a semantic grouping, since temporal sorting tends to create spatial arrangements that are perceived as coherent [RW03].

4.2.2 Interaction

When the control device is set down onto the table, the spiral appears. For a few seconds, it remains semi-transparent and both the lens and the spiral rotate with the physical handle. During this time, a user can determine the initial position of the lens. Right-handers will, for example, move the lens to the upper right side of the spiral (see Figures 4.1, 4.2) so that they can conveniently turn the handle with the left hand, while using their right hand and a pen for interaction with the enlarged photos.

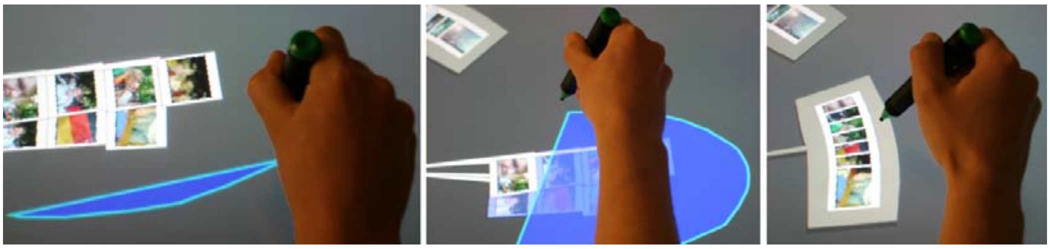


Figure 4.3: Grouping photos into a new event.

This mechanism also solves the general orientation problem by allowing each user around the table to adjust their **Photohelix** to best suit their needs (if several helices are available). It is also possible to reorient the whole interface at any time by just lifting it up, if several people share one helix or if the seating arrangement changes. To always ensure a comfortable working position, the helix can also be repositioned at any time by moving the physical handle to another spot on the table.

After the user has adjusted the initial orientation, the spiral is rendered solid and the lens remains fixed on an imaginary line running along the radius of the spiral. The spiral now turns with the handle, and the user can bring different areas of the spiral underneath the lens. The lens will travel inward and outward with the spiral windings, with every full turn applied by the user. To scroll faster, the handle can be twisted and then let run freely, to scroll back or forth several windings. The physical inertia of the handle in connection with a non-linear mapping of the time scale thus supports fast physical scrolling to cover larger time frames.

Individual images and events can be moved freely on the table surface, for example, when overriding the default chronological arrangement or organizing larger arrangements into sub-groups. To create events of closely related images, the user can simply circle the individual images with the pen. These are then automatically grouped into a new event, rendered as a slightly curved box containing semi-overlapping images (see Figure 4.3). New events also appear as new piles on the spiral and are connected to their pile by the umbilical cord. Cutting this cord dissolves the event again. To inspect the contents of events, the user can flip through the stack with the pen and see each individual photo in full (see Figure 4.4). This interaction technique resembles the handling of flip-books.

When a photo is dragged out of such a group, a full-size copy of the image is created and

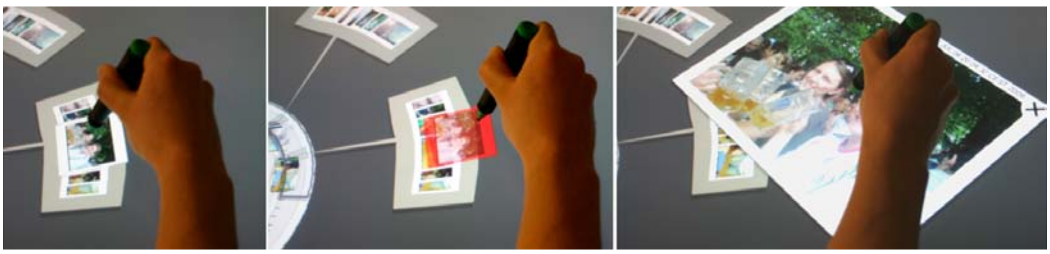


Figure 4.4: Flipping through an event to inspect images. Dragging images out of the event to create an enlarged copy.

positioned on the table (see Figure 4.4, third image), which can then be moved with the pen or a finger. With the dedicated widget at one of its corners, it can be scaled and rotated (see Figure 4.5). This mechanism specifically supports the creation of temporary structures (e.g., several shots of the same person in one pile), without modifying the long term organization of the collection.

4.3 Browsing, Filing and Sharing

Throughout the body of literature, a set of typical activities performed with media collections can be found. Future applications should try to support these activities, which are: *1) Filing* - The task of sorting media into folders or albums. *2) Selecting* - A repetitive activity in which users go through their collections and decide which items to keep and which to get rid of. *3) Sharing* - Often the ultimate usage of media at the end of its lifecycle. This can be performed remotely via e-mail or websites but also (and preferably [CRM04]) co-located for communication and storytelling, such as updating friends and family about recent events. *4) Browsing* - Users look at pictures from different time periods, possibly to revive old and forgotten memories.In this section we will discuss how the presented visualization and interaction techniques support the activities (*browsing, filing, selecting, sharing*).

Figure 4.5: Rotating and resizing pictures in Photohelix.

Browsing a photo collection can be done on different scales: Large-scale browsing indicates the act of going through a collection to identify a certain set of pictures. Small-scale browsing refers to inspecting events and images in more detail, for example, comparing several similar images to further use them for sharing or printing. **Photohelix** provides a convenient overview of the entire collection, structured by time. The automatic grouping of photos into piles of thumbnails serves two functions. First, the collection is presented in a space efficient way to avoid information overflow. Second, specific events or situations can be recognized on the basis of their representatives and the pile's position in time. Turning the helix brings different time intervals under closer inspection. Events and pictures displayed in the detail view (see Figure 4.2) convey more information to the user since all images are (at least partly) visible. This presentation is still space-efficient, and also resembles the way in which printed photos can be spread out. Flipping through events allows a fast inspection of large sequences. And the photos found while browsing, can easily be dragged out of the sequence in order to inspect the full-size version of individual pictures in further detail. The *filing* process is made more efficient by the automatic arrangement by time, and the ability to freely manipulate photos and events. This eliminates additional steps such as navigating folders. In many cases, the chronological sorting already meets the users' organizational intent. In addition, photos can be spatially arranged on the entire surface of the desk, which allows individual semantic mappings. For example one can create piles (e.g., left is for bad photos, right for good ones, top for funny, bottom for serious).

Photohelix particularly supports the *sharing* of photos, in this scenario, showing the photos to the people around the table. For this purpose, they can be freely moved and rotated toward others. In fact, the individual arrangement on the table can be used to convey parts or the structure of the story to the observers. For example, users can create a heap of pictures close to their edge of the table. While telling a story they can subsequently move and orient currently discussed images toward the audience. Additionally, collaborators can pick photos up and further inspect them at any time. It is also easy to hand over the entire collection since it is represented by and linked to the physical handle.

The current implementation of **Photohelix** does not support the *selecting* activity. In order to support this, it would be mandatory to delete "bad" pictures. We experimented with several interaction techniques to delete pictures. However, all of them where prone to in-accurate or faulty input. Furthermore, we encountered difficulties in attaining "raw" images, since many users already performed the selection process during or directly after downloading images from their camera. Furthermore selecting or curating a collection of photographs is more likely to happen alone and on a Standard PC rather than on a tabletop computer in the presence of others. Therefore, we solely focused on the first three activities.

4.4 Evaluation

To verify our design decisions, we evaluated **Photohelix** in a qualitative user study. We were specifically interested in whether our interaction techniques actively support the highly dynamic and informal activities associated with photo handling and the results are summarized here (details can be found in [HBB07]. Later we discuss the suitability of this hybrid interaction approach in general in section 4.5.

Participants Twenty participants (13 male, 7 female) between the ages of 18 and 34 were recruited from amongst our students and the local community. All of them had normal or corrected-to-normal vision and were right-handed. All participants were power users who worked on a PC for four or more hours daily. In contrast, most participants had little to no exposure to interactive surfaces (including PDAs and Tablet PCs). Only one participant reported an occasional usage of interactive whiteboards. All participants own a digital camera (33% use it 3 to 5 times a year, 48% take pictures once or twice a month 19% use it weekly or daily) and image collection sizes ranged from approximately 100 to 10,000 pictures (with an average of 3340).

Study Setup and Tasks We envisioned a scenario that includes elements of storytelling and picture-sharing, which required that participants be familiar with the images. Participants were therefore asked to bring a subset of their own collection. Typical image sets included 80 to 100 pictures from a time frame of approximately two years. They were also distributed over 6 to 8 different occasions (e.g., vacation, barbecue). While the size of these sample collections were not realistic, their distribution over occasions and time seemed to resemble real configurations (cf. [KSRW06]).

The four tasks were designed so that users would gain exposure to all aspects of **Photohelix**'s functionality, and so that we could map tasks to the different activities identified in our requirements analysis. After completing an explorative warm-up task scaffolded with instructions on using **Photohelix**, participants were asked in Task 1 to *file* the images of their collections into events, thus permanently archiving the pictures. In Task 2 they should *browse* the entire collection and choose one particular event. They were subsequently asked to select one representative photo, enlarge it and explain why they had chosen it. Task 3 was aimed at *sharing* and participants had to give an update about a recent vacation. During the course of this process they were asked to enlarge several images and show them to the study conductor, who played the role of a friend or acquaintance. Task 4 was to choose four possible candidates from each of the four seasons to be used as the desktop wallpaper.

4.4.1 Results

In our experiments we gathered quantitative data (i.e., Likert-scale responses) as well as qualitative data from a semi-structured interview with open-ended fill-in responses. We also video taped every session and analyzed these recordings afterwards.

We wanted to find out whether users liked our system, and which aspects were especially appreciated or needed improvement. To answer these questions we evaluated the responses to several Likert-Scale questions (Scale: disagree (1) – neutral (3) – agree (5)) as well as the free-form comments. In general people liked **Photohelix** (4.1/5) and thought it was “easy” and to use (3.7/5). They also liked the look and feel of the interface (4.2/5). Additionally, they thought that the visualization provides a good overview of the photo collection (3.9/5). When asked to rate specific functionalities of the system (see Figure 4.6), users liked the chronological sorting of pictures (4.85/5) and the visualization of time using a spiral-shaped calender (3.8/5). Furthermore, the possibility to freely position the spiral (4/5) and images (4.35/5) on the table received good ratings, as did the usage of the physical handle to adjust the time (3.85/5). Users appreciated the interaction techniques to create (4.1/5), flip through (3.7/5) and dissolve (4/5) events as well as the interaction techniques to rotate and scale images (4.3/5).

The qualitative comments we received further emphasize the above ratings. Several comments suggest that our design goals have had a positive impact. One participant said *“I like that all the pictures are already ordered by time. I like that I could see all the pictures quickly, just by turning the dial or flipping through the photos, and that I don’t have to click into a folder in order to retrieve pictures.”* Several other comments along these lines suggest that the *overview at all times* and *details on demand* paradigms are indeed beneficial for the browsing and filing

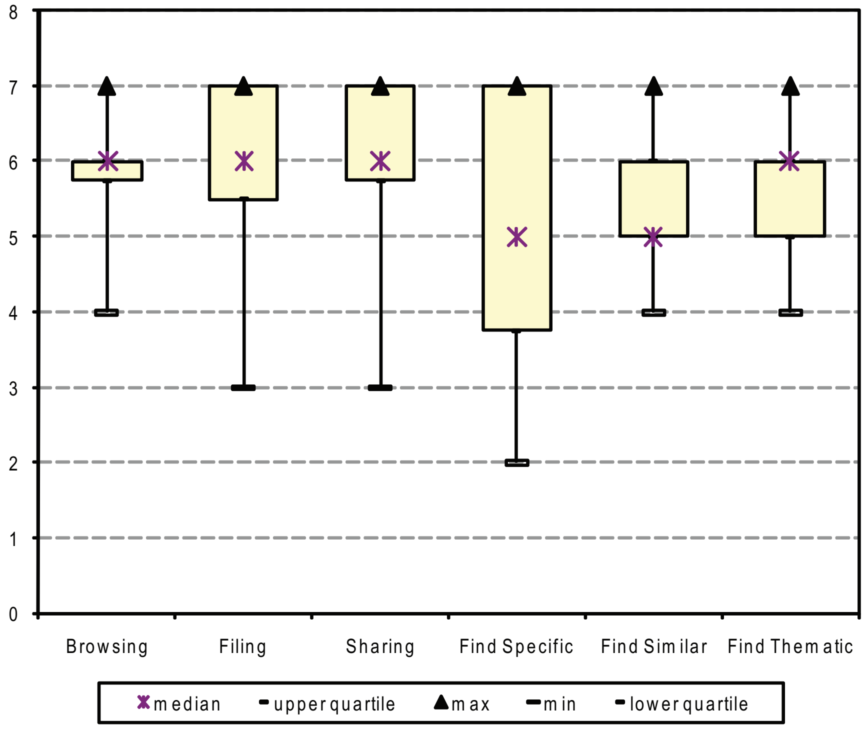


Figure 4.6: Appreciation of different functionalities in Photohelix. (Scale: dislike (1) – neutral (3) – appreciate (5))

activities. We also received many comments on the *flexibility* of the interface: *"... it was very intuitive, cumbersome copying of images becomes obsolete due to the possibility to create copies and temporal collections by simple dragging"*; and on its qualities for sharing: *"... browsing and viewing photos together is nicer with this kind of interface. It's more fun, too."*

4.5 Observations and Implications

The results from our evaluation suggest that the current design provides several benefits for browsing, organizing and sharing (as in storytelling) digital photo collections. The physical handle serves as a graspable representation of the entire collection and the helix shaped calendar functions as a possible visualization that is coherent with most users' mental model of their collection. Thus the combination of the two provides effective means to access the collection and to retrieve individual pictures for further inspection. For a discussion of issues in the particular context of photo handling please refer to [HBB07].

In the remainder of this section we discuss aspects of the interaction styles along the criteria of *flexibility* and *physicality* introduced earlier in this Chapter. In addition to our findings and observations from the user study discussed in Section 4.4 we have collected much more data from users interacting with the **Photohelix** over the span of two additional experiments. One user study (as of now unpublished) was performed to learn more about particular differences and commonalities between direct-touch and tangible interaction. A third user study investigated a particular conversational aspect, called sidetracking, that we observed frequently in participants engaging in photo-talk [HK09].

4.5.1 Physicality

When launching into our exploration we had a set of expectations (listed in 4.1) motivating the use of a physical handle for interaction. Some of them could be confirmed some require further study.

First, we wanted to unlock manual dexterity and exploit physical affordances to create a richer interaction vocabulary. Although the virtual manipulations in **Photohelix** are limited (i.e., positioning the calendar, adjusting the lens position) we could observe how users exploited their fine-grained motor skills (and how they quickly learned to operate the device) and the mechanical properties of the device in order to accomplish their current task at hand. Figure 4.7 illustrates a typical learning process. Many times participants would initially treat and touch the dial with great care and pay a great deal of visual attention when attempting to figure out the mechanics of the device (Figure 4.7 a). Often both hands are used, possibly to gain more control over the object. In this stage turning of the upper half of the device was usually very slow almost as if participants were worried to damage the device or otherwise cause malfunctioning of the system (e.g., confusing the tracking). After a couple of minutes confidence rose significantly. Usually participants reverted to on-handed operation of the device and also relaxed their posture; sitting more upright and resting the forearm on the table's edge (Figure 4.7 b). However, in this intermediate stage most participants did always look at the device when turning the dial. After further familiarization with the system and the physical device participants ended up using the device without spending much visual attention for example, solely focusing on events brought under inspection in order to find a particular set of pictures.

(a) (b) (c)

Figure 4.7: Evolution of manipulation style over time. (a) Careful, two-handed manipulation of physical object, full visual attention. (b) one-handed manipulation, full visual attention. (c) one-handed manipulation, shared visual attention.

Furthermore, we could observe how users exploited the physical properties of the dial. After they had gained confidence in operating the device we could observe how they used different grasping and manipulation strategies in different situations. For example, grasping the dial with all four fingers to precisely adjust the position of the lens (e.g., when two picture sets were close to each other). Or using just the index finger to accelerate the dial and let it spin afterward to fast-forward, sacrificing precision for additional speed. This is an example where users flexibly re-purpose the device in ways not anticipated during the design of the system and not possible in a pure direct-touch version. However, this flexibility is limited to what can be sensed by the system; instances where this can be problematic are discussed in the *flexibility* section of this discussion.

Our third concern was whether the physical handle in **Photohelix** would encourage bi-manual interaction. Given the possibility to operate the device without paying much attention we did expect users to simultaneously interact with photos using their fingers or the pen and operate the device using the non-dominant hand. Figure 4.8 illustrates various degrees of bi-manual interaction. We did observe both symmetric and asymmetric bi-manual interaction. That is interacting with both hands in an coordinated effort to accomplish one task (e.g., scaling a photo with one hand, while moving the entire helix away to avoid overlap of visual elements as in Figure 4.8 c) or using the two hands simultaneous but with separate concerns (e.g., closing open photos while beginning to scroll and search for new images). However, the majority of interactions with the **Photohelix** did not happen in a bi-manual fashion – or at least hands were not used in parallel very often. Predominantly participants would use the dominant hand to manipulate individual images while resting the non-dominant hand on the table's edge (Figure 4.8 a) or use the non-dominant hand to operate the physical device to access particular events (Figure 4.8 (b)). Finally, some users only used their dominant hand to operate both the direct-touch elements and the physical handle.

Although bi-manual interaction did occur – within our studies we did not see it as much as anticipated. The reasons ultimately remain unclear but one possible explanation is that the

(a) (b) (c)

Figure 4.8: Various degrees of bi-manualism when interacting with the Photohelix. (a) one-handed manipulation of individual photos. (b) one-handed manipulation of physical interaction handle. (c) simultaneous manipulation of physical handle and individual photos.

task allocation to the dominant and non-dominant hand was too diverse. Usually one manipulates and views individual photos *or* browses the entire collection – both happened seldomly in parallel. Another possible explanation, especially for the participants that always or predominantly used their dominant hand, is that there might be a strong influence due to the acquaintance with the WIMP paradigm and its single cursor model (we did not actively encourage bi-manual interaction during our instructions). A theory that can be found elsewhere in the literature as well [TKR+08, TKSI07]. Finally, it is conceivable that we have been observing the natural human tendency to minimize efforts. Why use both hands if the one is enough? Although we can not provide any experimental evidence to when and why participants would use two hands it seems plausible to assume that they would only use both hands when there is an immediate benefit such as quicker task completion, more control or reduced work load. This puts our initial assumption that physical objects would naturally encourage bi-manualism somewhat into perspective.

Our last expectation concerned the operation of hybrid interfaces without visual attention or *eyes-free*. The use of physical 3D elements at the interface offers a form of tactile feedback unavailable in a direct-touch enabled GUI. This tactile feedback should theoretically redistribute cognitive effort into other sensory modalities. So while with the GUI the cognitive effort is largely expended through visual attention, with the tangible interface some of the effort of control (i.e. feedback loops governing appropriate extent of interface manipulation and understanding that you are still within the bounds of reasonable control movement) becomes tied to tactile interactions. Consequently, this should free up the resources of visual attention to engage in social processes, for which eye contact is arguably important for maintaining the flow of human-to-human interaction.

To further assess our assumptions we performed a comparative user study with the **Photohelix** system. As alternative to the hybrid version of the interface we designed a pure direct-touch version of the interface incorporating precisely the same functionality. We hypothesized that the tangible interface would free *visual* and *cognitive* resources and reduce visual monitoring of the dial during social (i.e., two-user story-telling) use. 20 participants were asked to bring their own photos to talk about and were divided into two conditions (tangible and direct-touch). We recorded log data (e.g., type and duration of interaction with the system) and recorded each session using two cameras, one mounted overhead to monitor manual interaction with the interface and one over the experimenters' shoulder giving full-face view of participant and some surface information. Post-experiment, we analyzed the log data and counted instances of eye-contact and instances of visual device monitoring using an open-coding method.

To attain more qualitative data about the social interaction between the participant and the experimenter we applied a technique referred to as "thin-slicing". This technique is adapted from a research methodology used in social psychology [AR92], a technique that harnesses the ability of human observers to make accurate snap judgments about certain social situations from the presentation of short-duration video-clips, referred to as 'thin slices of behavior'. The technique has originally been used to determine key indicators of social behavior, for example to measure communicative efficacy. This methodology has been productively adapted and validated as a research technique for analyzing social interactions during novel technology use by Lindley [Lin06] and it is in this vein that we used the technique. This approach offers a more objective analysis, free from participant-experimenter expectation bias (the participant might be inclined otherwise to respond favorably to the experimenter in self-rated analysis) and which can be analyzed for reliability with blind comparisons of multiple raters.

(a) (b)

Figure 4.9: Comparison of eye use patterns across conditions. (a) Participant talking about photo and interacting with dial in tangible condition. (b) The same interactions in the direct-touch condition.

In contrast to our expectations, post study analysis of all three data collections did not reveal any significant differences between the two conditions. Furthermore, the log data, count for incidence of surface and device monitoring as well as scores given by independent raters were virtually identical across conditions. In consequence our data did not allow to draw any conclusions how tangibility might affect patterns of eye use during interaction with an otherwise direct-touch interface. Looking closely at Figure 4.9 might suggest a simple explanation for these results; since our setup (relatively large display, small interface elements for navigation) and the task (photo-talk) are extremely visual it is reasonable to assume that the displayed information was dominant in this task. In consequence, should an effect exist that can be accredited to tangibility in the context of *eyes-free* usage, it was overshadowed by other aspects of our particular application scenario.

It might be tempting to try to isolate the effects of tangibility further in order to get a more accurate measurement of its benefits. For example, one could construct a similar experiment where physical interface and graphics are not directly (physically) co-located (an approach taken in [WWJ+09]). We chose not to do this because we were interested in how well tangibles work within the context of surface computing and therefore the co-located graphics are an essential aspect.

It remains open for debate whether measured effects that might be overshadowed by an integral aspect of the type of systems we're looking at (co-located graphics) can be transfered or generalized into real world settings. Re-visiting our earlier example of driving a car might be helpful in this context. Here clearly the physicality of the controls has an important impact on the safety of the vehicle's operation. It is important though that learning how to drive a car requires a lot of training and many sequences are repeated countless times. Repeating certain activities is a key aspect in motor learning and an irreplaceable one. Of course the tactile feedback provided by the controls plays an important role in the moment of interaction but when re-collecting our own initial attempts at driving a car most of us would agree that this aspect alone does not suffice.

Of course this raises questions whether many of the qualities often accredited to tangible interface artifacts can be translated to a hybrid tangible / interactive surfaces context. For further reading we would also like to draw the readers attention to two recent studies which compared pure direct-touch interfaces with hybrid interfaces in multi-media navigation scenarios. While the former study [TKR+08] could not find experimental differences across the two conditions, the latter study reports statistically significant efficiency benefits for the tangible condition [WWJ+09]. While this question warrants further research and is certainly an interesting one we would argue that the usage of tangibles in this context should be considered carefully in future applications. Also claims about assumed benefits of tangibility should be treated with care.

4.5.2 Flexibility

Because we are using tangible input devices (physical objects and touch surfaces) to manipulate virtual things, people bring with them many experiences from the physical world. Due to the almost unconstrained freedom of interaction we have in the real world, it is easy to frustrate or confuse users through a disparity of performed action and system response. To clarify, one should consider the following example often observed during our user studies. Participants often raised the complaint that the arrangement of photos on the calendar is fixed and linear (sometimes causing manual effort to reach chronologically distant events) they also expressed the wish for some sort of zooming mechanism many times simultaneously lifting the handle off the table to indicate how this could be implemented. When using tangibles as input devices it is practically impossible to anticipate all possible expected (ab)-uses of the device.

Consequently, it is easy to generate an unexpected or confusing system response. It is necessary to consider these physical experiences and to try and match the effect of a person's actions with their expectations. Although more studies would be necessary to further investigate this issue, the authors hunch is that consistency and externalization of rules is a key issue here. For example, participants responded well to limitations through mechanical constraints, possibly because they understood certain manipulations are impossible with this particular device. Other interactions that are mechanically possible (such as lifting the device or tilting it), but were not sensed, could be observed repeatedly. Often complemented by user comments on what event the particular manipulation should trigger in the virtual interface.

4.5.3 Summary

In summary we have showcased many interesting aspects of physical handles for interaction in the context of surface computing. In particular we have demonstrated how tangible elements can help to enable richer interaction strategies through their physical affordances, how 3D shape and mechanical constraints can make hybrid interfaces more approachable, easier to learn and how they invite for exploration. We have also explored other characteristics and benefits often accredited to tangibles such as their suitability for bi-manual interaction and *eyes-free* operation. However, here we could not find experimental evidence for an actual benefit when compared to direct-touch interfaces. Finally we highlighted how tangible interface elements share similar problems in terms of *flexibility* with gesture-based interfaces.

From our exploration into hybrid interaction on tabletops we can distill the following key aspects to inform the next steps in our research endeavor:

- Physical affordances make system functionality more accessible for exploration. Physical handles also seem to encourage learning through experimentation.
- The 3D nature inherently possessed by tangibles, externalize some of the functionality built into the system. This can mitigate learning and memorization issues.
- Tactile feedback and mechanical constraints can afford a richer interaction vocabulary.
- Sensing and differentiating of fine-grained manipulations can be a limiting factor.
- Predefined mappings from sensed input to system behavior can also pose a problem in respect to flexibility.

Chapter 5

Discussion

In Chapters 3, 4 we discussed two interaction paradigms typical for many tabletop applications; gesture-based [Min84, ZM98] and tangible interaction [IU97, UI00]. Tabletop settings often lack keyboards and mice, therefore performing complex commands can be challenging [SRF+06, MCP+06]. It seems promising to use gestures for command invocation [WB03]. Many interesting interaction techniques and applications based on this principle have been demonstrated [MHPW06, WB03, WSR+06, FS05, WMW09]. The gesture-based interaction paradigm is a promising candidate to address the special design constraints imposed by the tabletop characteristics, such as orientation (text and otherwise), coordination of (multi-user) interaction [KCSG03] and lack of standard input devices. Gesture-based input allows for an immediate interaction, the possibility to directly touch information and spatial coupling of input and output. Also the possibility to design applications without abstract menu or command interfaces seems promising in terms of learnability and general ease of use. However, many gesture-based applications have been designed ad-hoc and little knowledge about how to design such interfaces is available. Furthermore, studies have revealed that too complex gesture sets can be challenging for the user and hard to remember (especially for interactions that are not performed on a regular basis) [HTB+07].

Our own studies of the gesture-based interaction style using a prototypical application for collaborative problem solving confirmed some of the assumptions often claimed in favor of gesture-based interaction. In particular the low learning threshold for novice users. Another positive aspect of this interaction style is the possibility to design more organic or pseudo-physical interfaces and thereby create a tighter coupling between the digital and physical domain. However, our experiences with the BrainStorm system (cf 3.1, [HTB+07]) and other gesture-based prototypes unveiled several issues with this approach. Two aspects seem to be especially problematic; due to the lack of feed-forward (visualization of possible commands before gesture registration) many participants had difficulties in memorizing certain gestures - a problem that is likely to become even more severe in systems with a higher complexity and more commands.

Based on our observations the lack of *flexibility* was the most severe issue. Here the need to recognize a pre-defined set of gestures and the need to map these gestures to scripted system response seems to be fundamentally problematic. One might argue that our gestures were not designed well enough and some of the problems we encountered in terms of user frustration could indeed be mitigated through additional feedback or better interface design. Although recent work by Wobbrock et al. [WMW09] has yielded interesting results from user defined gestures on tabletop computers, it remains debatable whether statically defined gestures and scripted gesture-command mappings are a good fit as an interaction style, especially when compared to the rich, flexible and often dynamically adaptive ways in which we engage with real-world objects and how we use these to solve problems or make use of them.

Tangible interaction [IU97, UI00] promises to design richer interactions making use of manual dexterity and motor skills. The 3D nature of many physical objects affords specific ways of usage and as such helps to mitigate learnability and long-term memorization issues [Cla00]. Many researchers have speculated that motor memory would bear efficiency benefits over other interaction styles (e.g., [RS00]) (although experimental evidence thereof is lacking as of now). Finally many positive aspects for embodied interaction and social interaction (e.g., access allocation, shareability) have been accredited to tangible interaction [HB06, BSK+05]. One drawback of tangible interaction is the need for special purpose devices as well as sensing hard- and software. In order to optimally exploit physical affordances one would have to custom build many different devices for every conceivable application. Furthermore, the flexibility and richness of interaction undoubtedly a quality of physical objects can be severely limited by the sensing scheme utilized. Interaction with real-world objects is often not restricted to simply functional modifications but consists of a myriad of nuances. For example, it might not be enough to be able to sense if an object has been touched in some situations but how it has been touched in order to correctly deduce the users intention. Even if sensing of interactions was not a problem, the recorded data still needs to be interpreted and mapped to some kind of response on the systems side. These mappings are usually designed and implemented a priori and might not appropriately capture the users intention in every situation.

Our experiences from **Photohelix** [HBB07, HK09] and other hybrid prototypes [TKR+08, Hil07, HWTB07] revealed interesting but not entirely unambiguous results. Our designs were received well by user study participants and the tangible aspects were commented on as being “natural” to use, easy to learn and also “more fun” than purely digital interfaces. We could also observe how the usage of tangible objects enables richer manipulations and also re-purposing or re-designing of interaction elements – of course only to a limited degree, restricted by the limitations of the virtual interface. But it is noteworthy that we could not, across several user studies, unearth any experimental evidence for or against some of the claimed benefits of tangible input devices on digital tabletops (listed in 4.1) over pure direct-touch interfaces. However, in 4.5 we discussed and showcased several interesting aspects of *physicality* in hybrid interfaces such as the **Photohelix** that warrant further investigation of this interaction paradigm.

The tangible interaction paradigm suffers from similar limitations as the gesture-based interaction style discussed in Chapter 3 in the context of flexibility. Although we have seen how the physical handle enabled richer manipulations and various strategies as how exactly an object can be manipulated in the end these manipulations have to be sensed and mapped to events in the graphical user interface. In other words the recognition and interpretation bottleneck persists. Mechanical constraints and possibly actuation can mitigate this problem somehow but can not do away with the more fundamental problem of scripted and predefined system behavior. This constraint suffers from the asymmetry in knowledge about the systems innards. While the developer knows (and defines) the rules of system behavior the user has to rely on system feedback to make sense of these rules. In the real world we all have developed an understanding about the laws of physics, through schooling and through first hand experience – therefore it is much easier to predict the behavior of individual objects and that of more complex mechanisms.

Part III

A New Model for Tabletop Interaction

In the first two parts of this thesis we explored interaction styles for tabletop computing based on a literature review (cf. Chapter 2) followed by our own explorations and the analysis thereof (Part II). As discussed in 5 we have seen that both gesture-based and tangible (or hybrid) interaction are viable options for tabletop computing applications. We have also uncovered some limitations for both paradigms. Primarily aspects of *physicality* in our prototypes have shown to be valuable. Real world resemblance of on-screen elements such as the pseudo-physical gestures and behavior in the case of BrainStorm (cf. 3) and physical affordances of tangible artifacts in the case of Photohelix (cf. 4). However, both approaches were limited by the lack of flexibility in the interface which is caused by pre-programmed or scripted behavior of virtual interface elements.

In the following Part we discuss a new model for tabletop interaction that aims to combine the important aspects of physicality and flexibility in a coherent way. A model that increases the interaction fidelity of direct-touch interfaces beyond the "finger-as-cursor model" while maintaining a maximum of flexibility and ridding the interface of scripted and static behavior. This model is based on two fundamental concepts:

- Allowing users to make use of their full manual dexterity when interacting with virtual worlds by enabling rich interactions with multiple fingers and whole hands, both hands at the same time and also through physical objects.
- Making virtual objects feel more real so that users can apply real world knowledge to manipulate virtual objects and apply various manipulation strategies as they see fit. Realistic behavior of virtual objects may also help to make interfaces easier to explore and discover functionality. In our interaction model the rules that apply to virtual objects are similar to those of the real world.

One important precondition for this new model is hardware capable of sensing these rich ways of direct-touch interaction. Our own early explorations have been carried out with tabletop hardware with sensing capabilities limited to two contact points. Furthermore, the used Smartboards [SMA03] only provide a single x,y coordinate for each touching object thus poorly approximating the shape and size of contacts – that is not differentiating between a flat hand and a fingertip for example. These limitations have shown in various ways during our explorations. For example, effectively limiting Photohelix to a single-user application. Another effect of these limitations prevented users from assuming a more natural posture for handwriting in **Brainstorm** (i.e., holding down the sheet of paper with the flat of one hand and writing with the other hand).

Emerging tabletop hardware promises to address these limitations by sensing multiple points of contact (such as fingertips) simultaneously and whole hands or even the outlines of objects – both of hands and of other physical objects – in contact with the display surface. Enabling multiple users to interact with the system simultaneously and enabling individual users to interact with the system in more natural and flexible ways, applying different manipulation strategies similar to interactions with the real world. To provide an overview of available hardware platforms and approaches to rich interactive surface sensing we discuss solutions proposed and demonstrated by the research community and products now being offered by various companies in Chapter 6.

After the related work we discuss our own interactive surface prototype capable of sensing multiple simultaneous contacts by fingers, hands and objects in Section 6.2, followed by a comparison and assessment of various sensing approaches in Section 6.3.

In Chapter 7 we then discuss our interaction model combining rich input from interactive surface hardware with a 3D environment powered by a gaming physics simulation. We describe various iterations of the model and discuss trade-offs between implementation alternatives. We showcase some of the compelling interactions enabled by this new model. Finally we discuss results and observations from a lab-based user study. This study revealed interesting insights into users' reactions to our new model for tabletop interaction.

During our evaluations of this model we discovered one major limitation. Because sensing in most surface hardware is optimized toward detecting *on-surface* contact and to discriminate it from fingers and other objects *above* the surface, the sensed data is usually 2D. In contrast, our proposed model is based on a gaming physics engine which is inherently 3D. Here the mismatch of input and output dimensionality is a fundamental limitation in manipulating objects using the third dimension. When interacting in the real world we routinely manipulate objects in 3D – even on tables – such as leafing through the pages of a book, stacking objects on top of each other or holding sheets of paper in various ways to reveal or hide content from others. The mismatch of sensed input and displayed graphics makes some of the simplest real-world actions such as stacking objects or placing them in containers difficult if not impossible using these interfaces.

In Chapter 8 we discuss several emerging display and sensing technologies that allow to, simultaneously, project a stable image onto the surface and image through the display to capture the users' hands at a greater distance than possible in traditional setups. Allowing us to develop techniques that include the space above the tabletop into the interaction with the digital realm.

This part of the thesis is then concluded by a chapter on a technique based on these emerging display technologies developed to enable interactions *on* the surface and *above* the surface (Chapter 9). Our goal has been to define a technique that feels as natural and as direct as possible, giving users the sense they are actually lifting the objects off the surface. We chart the evolution of this technique, implemented on two rear projection-vision prototypes. Both use special projection screen materials to allow sensing at significant depths beyond the display surface. Existing and new computer vision techniques are employed to detect hand gestures and poses to allow object manipulation in 3D.

Chapter 6

Technical Foundation

> "People who are really serious about software should make their own hardware."
> **Alan D. Kay**

6.1 Multi-touch Input Technologies

Digital tabletops share many commonalities with regular (often single user) touchpads. Early research into multi-touch capable touchpads has been undertaken by Lee, Buxton and Smith in 1985 [LBS85]. Their digitizer is capable of detecting multiple fingers simultaneously with an array of capacitive proximity sensors (the fingers serve as the second plate of the capacitor). In addition to the location of fingertips the hardware is capable of approximating pressure values through the increase of capacitance as the fingertip flattens against the sensors surface. Many tabletop systems that have been proposed later to support co-located multi user collaboration are based on the same capacitive sensing principle. For example, **DiamondTouch** [DL01] and **SmartSkin** [Rek02] are based on this idea and are discussed in detail later in this chapter alongside other systems that stand for this first category of tabletop hardware: *embedded multi-touch sensing* (Section 6.1.1).

Another way to sense contact with a digitally enhanced surface is through computer vision. Usually a camera is pointed directly at the users hand or a camera is positioned behind a projection screen. Vision algorithms are then used to detect fingers and other objects touching the screen. One of the earliest vision systems for full hand interaction is Krueger's VIDEOPLACE [KGH85] a system that combines a participants live video image with computer generated graphics. A video image of the user's hand is segmented and used as input for various aesthetic and practically motivated applications. VIDEOPLACE allows for full hand interaction and does not introduce abstract concepts such as pointers. Wellner's DigitalDesk system [Wel93] uses thresholding techniques to differentiate a pointing finger from a cluttered background. Contact with the desks surface is detected using a microphone built into the surface. The system demonstrated rich interaction techniques with real paper and virtual elements projected onto the desk. Later many systems have explored various ways of detecting when users touch a display surface, in many cases using some kind of infrared illumination scheme and image thresholding technique. Examples are Matsushita and Rekimoto's HoloWall [MR97] or Wilson's PlayAnywhere [Wil05] system. These and other *vision-based* approaches are discussed in detail in Section 6.1.2.

Categorizing the different approaches is not only interesting for classification purposes but also and more important has significant impact on interaction styles that are enabled or restricted by the properties of the hardware and its chosen sensing approach. The properties of these different classes of devices typically support one interaction style well while they sometimes are limiting for others as we have seen in Chapters 3 and 4. Furthermore, novel sensing developments will always directly impact the user experience as they change how users can interact with the system. Therefore new interaction techniques are sometimes enabled by novel sensing approaches as in the case for our new model for tabletop interaction (Chapter 7 – or they trigger the development and refinement of novel hardware, as is the case with our interaction technique discussed in Chapter 9.

To gain an even better understanding of the properties of this kind of interactive devices we discuss our own experiences from building such a vision-based sensor in Section 6.2. This chapter is then concluded by a discussion of the pros and cons of the respective approaches categorized and discussed here in Section 6.3.

6.1.1 Embedded Multi-Touch Sensing

In the following sections we discuss several systems that use embedded sensors to detect touch in different ways. The main advantage of embedding the sensing infrastructure into the display surface is the enabled form-factor. Devices using embedded sensing can be placed onto, or build into, a normal table allowing users to sit comfortably around the table while resting their feet underneath it.

Capacitive Sensing

Touchpads that can sense multiple fingers were first presented by Lee, Buxton and Smith in 1985 [LBS85]. The **digitizer tablet** was composed of a matrix of small metal plates, each serving as capacitive proximity sensor where the finger and metal sensor serve as the two plates of a capacitor. A row and column scheme was used to read the capacitance between finger and sensor. The system was capable of accurately determining the position of several simultaneously touching finger tips. In addition the system could approximate pressure applied by each finger as the capacitance changes with the finger tip flattening against the surface. The authors also provide early motivation why multi-touch capabilities are indeed wishful. Examples given are a polyphonic piano simulation or graphical applications where each contact point controls one parameter of a multi-parametric command simultaneously. The ability to sense pressure could be used to introduce a hover state in cursor control applications usually missing from most touchpads. The system was designed as pure input device and all graphical output was displayed on an additional monitor. Hence, the digitizer lacked one crucial property essential for tabletop systems; the direct spacial coupling of input and output.

Utilizing a similar sensing scheme but adding direct coupling of input and information display is the **DiamondTouch** system by Dietz and Leigh [DL01]. Developed as a prototype by Mitsubishi Electronics Research Laboratories (MERL) it is now available as commercial product. At this stage units are still hand produced and costly. More information is available online [1]. The **DiamondTouch** system is an early example of single display groupware (SDG) [SBD99] – a term now commonly applied to most tabletop applications supporting multiple users. The digitizer is composed of a lattice of antennas embedded in a projection screen. The antennas couple capacitively with the finger of users upon touch, again enabling the accurate detection of multiple simultaneous touches. In addition to other systems **DiamondTouch** hardware is capable of identifying individual users. In order to do so users are also coupled to a receiver through a capacitor plate embedded in the chairs around the table. By driving each of the antennas with a unique signal the system is able to discriminate which finger belongs to which user offering unique interaction possibilities. The system uses time-division multiplexing to cycle through the rows and columns of antennas. Therefore the provided signal only yields the margins of a contact region. A user touching with two or more fingers produces an ambiguous signal leaving room for several interpretations of the actual contact position. In these cases the system only reports an axis-aligned bounding box around the area touched by the user. This limitation also has implications on the interaction techniques that have been proposed for **DiamondTouch** systems. Another drawback is the fact that the sensing surface is completely opaque and requires top-down projection to produce an image. This increases the size of the system and causes occlusions from hands and forearms when users interact digital information shown on the table's surface. However, it is noteworthy that this system has been used in many research projects and tabletop studies. In consequence the **DiamondTouch** system has a significant impact on both interaction techniques and the communities understanding of tabletop computing in general.

Also based on the capacitive coupling principle is Rekimoto's **SmartSkin** [Rek02]. **SmartSkin** uses a grid of 72, 100 mm square sensors rather then the 160 + 96 (256), 5 mm-wide strips in **DiamondTouch**. The sensors measure the electrode's self-capacitance, or capacitance from electrode to ground (other touchpads such as the **DiamondTouch** system often measure mutual-capacitance between electrodes). This self-capacitance changes when a grounded conductive object approaches the electrode and concentrates electric field lines. As a conductive object approaches one of the sensors the signal drops and the system can defer proximity to the object. Applying bicubic interpolation over its sensor values the system can produce a higher resolution proximity map. In a later stage computer vision techniques are used to process this proximity map and detect multiple contacts and complex contact regions as well as an approximation of the distance between the surface and conductive objects such as hands. This approach enables richer interactions not limited to the fingertips as it provides more detailed sensing data including outlines of whole hands or forearms in contrast to a simple bounding box. However, the system can not determine which user is in contact with the surface. The paper also briefly discusses several interesting interaction techniques. For example, exploiting the ability to determine proximity to

[1] `http://www.circletwelve.com/home.html` - verified 13/1/2009

implement a hover state in mouse based interactions. Another example makes use of the available rich shape information, interpreting the sensor data as potential fields where approaching objects (hands, forearms) repel virtual objects (See Figure 6.1).

(a) (b) (c)

Figure 6.1: Smartskin [Rek02] shape based interaction techniques. (a) Objects approaching the surface create a potential field virtual objects are repelled. (b) Users can interact with multiple fingers and both hands. (c) Entire arms can be used to gather several objects at once.

Wayne Westerman describes a similar sensing scheme in his dissertation [Wes99] albeit in a different form factor. This work has been the foundation for various products that have been commercially available from **FingerWorks** such as the **iGesturePad** [Fin08]. FingerWorks now has ceased operation. The hardware was aimed at replacing conventional keyboards and therefore had a similar size (20 cm x 40 cm). The sensor Matrix had a higher resolution than the **SmartSkin** system (using 1600 electrodes plates). Making use of the high resolution proximity map generated by the hardware Westerman also details a variety of advanced computer vision algorithms that reliably detect and discriminate multiple simultaneous finger contacts. The thesis also introduces techniques to differentiate palms from fingers and identification of individual fingers in order to improve text input experience and performance. Finally, the system allowed to perform various gestures to control high degree of freedom graphical applications without removing the hands from the touchpad.

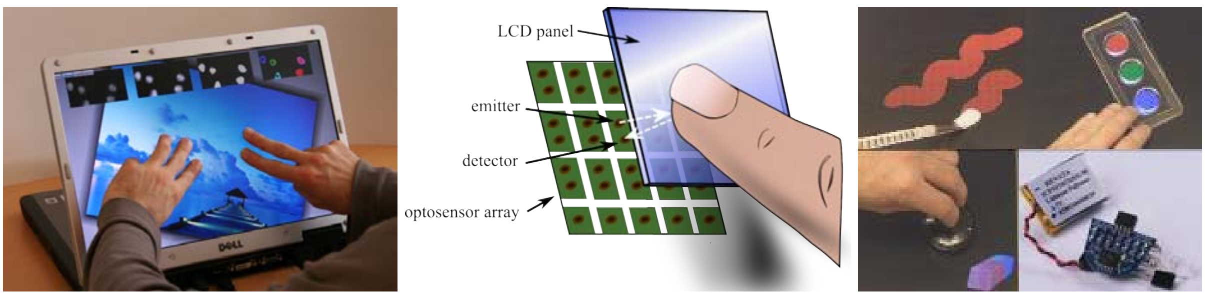
Resistive Sensing

Most traditional single-touch screens are based on the resistive sensing principle. This involves two resistive thin, metallic layers embedded in the screen itself. These layers are usually separated by a thin air gap. When an object touches the screen the two layers connect and a voltage change can be measured using two electrodes ($X\&Y$) and interpreted as touch event. Being based on pressure, resistive touch screens can be used with non-conductive objects such as styli or gloved hands. However, touchscreens based on resistive sensing need to be calibrated (sometimes calibration needs to be repeated routinely) and more important usually do not support multi-touch. In general if two fingers are in contact with the screen the barycenter between those points will be reported as cursor position.

Several multi-touch devices have been produced commercially, although their exact mechanism of operation is not always known both the **JazzMutant Lemur** [Jaz08] and the **Tactiva TactaPad** [Tac05] seem to apply a variation of resistive sensing that enables multi-touch detection and in case of the **TactaPad** also pressure sensitivity.

Optical Sensing

One major advantage of embedded sensing is the compact form factor this approach enables. By embedding some kind of electronics into the actual surface no space underneath is required as it is the case with most camera based systems (see Section 6.1.2). However, most systems that are based on sensing electrical signals use an opaque surface material to cover the sensing electronics from view (and to shield users from electric currents). In consequence these systems have to rely on top down projection in order to produce visuals. This reduces the compactness of the setup significantly and also yields a variety of other problems such as occlusion caused by the users hands and forearms (or even heads when users try to lean over the display in order to inspect information up close). Recently a new approach to interactive surface sensing has emerged. In order to fully integrate sensing and graphical output researchers have begun to embed optical sensing capabilities into regular thin form factor displays (e.g., a laptop's LCD display) and other compact setups.



(a) (b) (c)

Figure 6.2: (a) Thinsight [HIB+07] embedded optical multi-touch sensing. (b) schematic of sensing principle using retro-reflective optosensors. (c) tangible interaction demonstrated on second generation prototype [IBH+08]

Thinsight [HIB+07] is a new approach, fully integrated into a thin form factor, to sensing multiple fingers and physical objects in front of or on top of a regular LCD display (see Figure 6.2 a). In order to image through the display and identify fingers and objects custom built hardware has been embedded behind the LCD panel. At the core of the system is a grid of retro-reflective optosensors (see Figure 6.2 b). These devices serve two functions: IR-light is shown through the display by small IR-emitters; the optosensors also contain photodiodes sensitive to IR light optically isolated from the emitters. Any IR reflective object in front of the display will reflect the light back through the display where it can be picked-up by the detector. The result of all the emitter / detector pairs in the grid is essentially a low resolution image of the scene in front of the display in the infrared spectrum. Several interpolation and smoothing techniques are discussed in the paper in order to attain an image suitable to reliably detect hands and other IR-reflective objects. Furthermore, **Thinsight** is capable of two-way communication with devices in front of the display via suitably modulated IR light. For example, data could be transmitted using the established IrDa protocol from the display to handheld devices. In the other direction a remote pointing device equipped with an IR LED can be used to send data (e.g., specific commands) to the display. This is even possible at the same time as sensing multi-touch input. The initial prototype was restricted in its size, so that only a small fraction of the display was active. In a later publication [IBH+08] this initial proof-of-concept was extended to a larger form factor where the entire visible area of a 21" LCD panel was able to sense touch input. Beyond scaling the sensing technique to a larger screen-size the authors present several interaction techniques making use of the system's unique capabilities. Applications demonstrated included a painting application working with fingers and reflective brushes and palettes. Besides passive objects where presence detection is sufficient, also detection of object class based on shape heuristics is demonstrated (e.g., the color palette). Finally, object instance identification based on passive, reflective or active, IR-emitting markers (see Figure 6.2 c) has been shown.

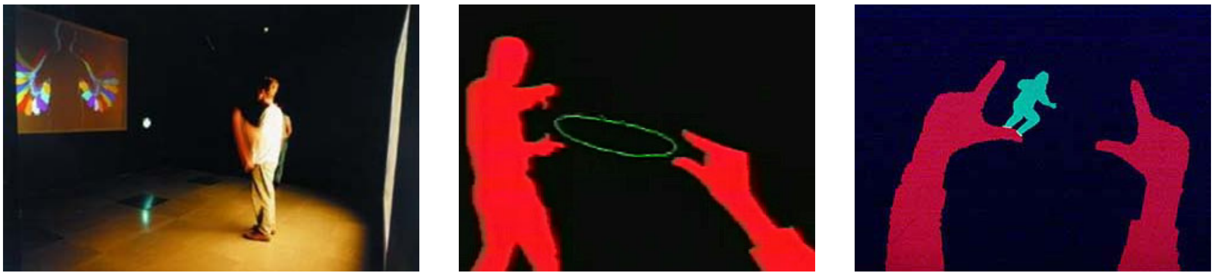
Sidesight [BIH08] uses a similar sensing scheme enabling touch input around devices wit a small screen such as mobile phones. The idea is to avoid occluding information shown on the display while providing users with richer interaction techniques that make use of the space around the device. The system uses infrared proximity sensors embedded along the sides of the device to detect fingers and to track their position. A proof of concept image manipulation and menu selection application showcases the interactions enabled by the technology.

6.1.2 Vision-Based Multi-Touch Sensing

A second strategy to implement multi-touch capable hardware is to use cameras and computer vision techniques to detect fingertips and other objects. Vision-based systems can loosely be grouped into two categories; "direct" systems, where cameras are aimed directly at the users hands, and "indirect" systems where cameras (usually behind a projection screen) observe changes in an image taken from the surface itself. Vision-based approaches have been tremendously popular and proved to be powerful in many projects proposed in the literature. One reason is that the sensor – a digital camera – is readily procurable and yields results quickly. A second compelling advantage is the flexibility and power of computer vision techniques that can be harnessed in vision-based sensing.

Hand and Gesture Recognition

One of the earliest examples of computer vision techniques in natural hand gesture human computer interaction are several projects by Myron Krueger and colleagues. In **VIDEOPLACE** [KGH85] users stand in front of a blank, wall-sized screen (see Figure 6.3 a). Camera images are segmented by the system so to attain a silhouette of the users hand or even entire body (see Figure 6.3 b + c). The outlines of captured objects are then projected back onto the

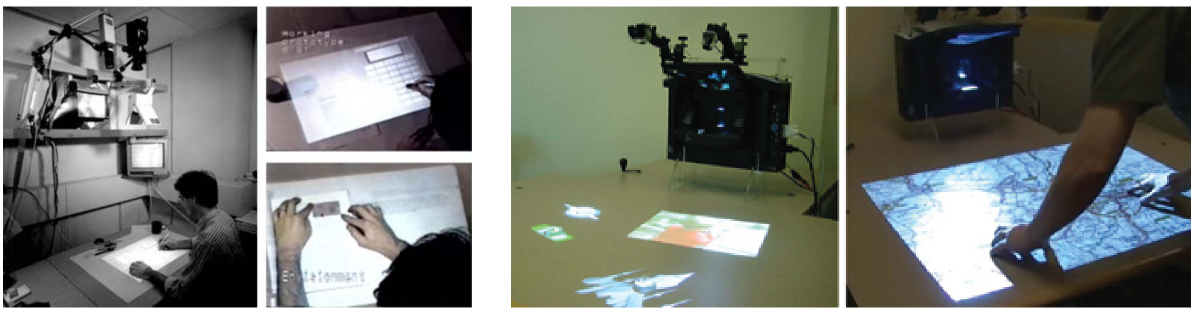


(a) (b) (c)

Figure 6.3: (a) The VIDEOPLACE [KGH85] installation. (b) users could interact with virtual objects with their entire body. (c) Outlines of hands were segmented from the background and used in the interaction.

screen and augmented with computer graphics or used to interact with virtual objects. This technique and similar ones have been demonstrated in purely artistic contexts and applications such as remote communication between two VIDEOPLACES.

Wellner's **Digital Desk** [Wel93] is an early example of a vision based tabletop system. The system uses frame differencing of thresholded images in order to segment fingers from a cluttered background (a regular work desk see Figure 6.4 a). A microphone is then used to determine when users tap the surface of the desk to indicate a "click". Several interesting applications have been shown that merged the physical items on the desk with virtual objects projected onto the table from above, such as a virtual calculator displayed next to a printed spreadsheet. Figures from the spreadsheet may be used as input to the digital calculator by simply pointing with a pen. The **Digital Desk** still serves as a source of inspiration for many recent tabletop projects.



(a) (b)

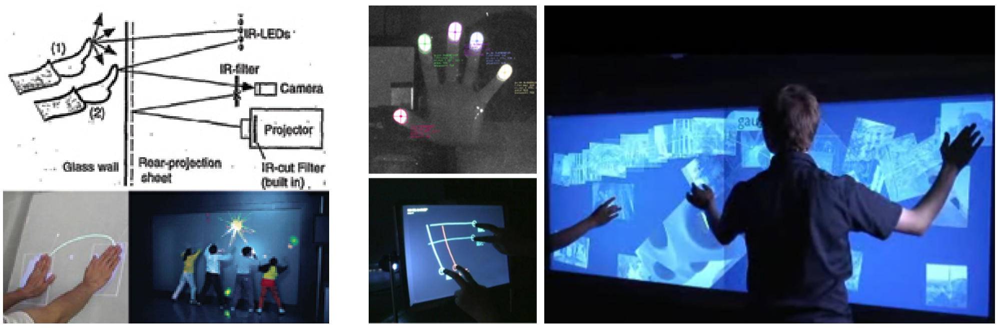
Figure 6.4: (a) Wellner's DigitalDesk setup on the left. Example applications include a projected calculator alongside a printed spreadsheet and copying printed information through simple gestures. (b) PlayAnywhere's compact and mobile setup. Bi-Manual map manipulations enabled by optical flow motion estimation.

Wilson demonstrates several interesting advancements to the direct vision-based approach in the **PlayAnywhere** [Wil05] system, a compact portable front-projected and vision-based tabletop prototype. A commercially available short throw projector is augmented with an IR illuminant and an off-axis mounted camera equipped with a matching IR-pass filter. Several computer-vision techniques are demonstrated to detect touch/no-touch events, detect and track visual barcodes (position and orientation), track pages of regular paper as well as a novel algorithm to interact with on-screen objects through the calculation of motion-flow rather than tracking of individual fingertips (Figure 6.4 b). Touch detection is accomplished through IR shadow tracking. Changes in the shape of shadows casted by fingers are heuristically evaluated. As a finger approaches the surface the distance between finger and shadow decreases, ultimately the finger occludes its own shadow. This approach only works for one finger per hand. In order to enable more fluid interactions with virtual information using multiple fingers or even whole hands a novel algorithm to calculate optical flow from the captured image is detailed. Instead of tracking discrete contact points, simple statistics about changes in the image are calculated, and areas of motion at the location of virtual objects are integrated into a continuous position and orientation transform.

The **visual touchpad** [ML04] system follows a similar idea but features more robust finger tracking and an extensive set of gestures on and above the touchpad. The disparity of two, off-axis cameras is used to track fingers and calculate the height above a low-cost "touchpad" (basically a black cardboard rectangle). After background subtraction to isolate hand contours, the system uses heuristics to label fingers and calculate their 2D orientation allowing for multi-touch input as well as multi-finger gesture recognition. Several interaction techniques are demonstrated that enable fluid and bi-manual interaction in photo manipulating, finger painting and text input scenarios.

Indirect Multi-Touch Recognition

In the last few years a second approach to vision based sensing has enjoyed great popularity in the research community and in commercially available products: "indirect" vision systems. The key aspects here are an IR illumination scheme and a camera pointed at the projection screen from



(a) (b) (c)

Figure 6.5: (a) HoloWall illumination and sensing scheme on top. Bi-manual and multi-user interaction on bottom . (b) FTIR screen as seen by the camera. Two-handed multi-touch interaction enabled by the approach. (c) The approach scales to large wall-sized displays.

behind or underneath. Touch events are not detected via directly segmenting hands and fingertips against the background and using second measurements (microphones, stereo disparity). Instead, these systems use a diffusive projection screen and the reflectiveness of skin in the IR spectrum to detect when the surface is being touched.

Matsushita and Rekimoto's **HoloWall** [MR97] is an early and typical example for this approach. The HoloWall is a wall size interactive display. It consists of a regular sheet of glass coated with a semi-opaque, diffuse projection screen. Behind the wall an IR light source and a video camera equipped with an IR cut-off filter are installed (only letting light pass above a wavelength of 840*nm*). A diffuse projection screen in front of the wall shields objects from the camera's view. Unless an IR reflective object - human skin is roughly 20% IR reflective - is brought into close vicinity of the glass. Depending on the selected threshold, objects close enough (0 to 30 cm) reflect some of the IR light back through the screen and therefore become visible to the camera (Figure 6.5 a). Not only fingers but any IR reflective object can be sensed, for example objects tagged with a 2D barcode can be located and identified. The system can detect many contact points simultaneously and because of the camera's location occlusion problems are mitigated.

Also relying on sensing in the IR domain, Han demonstrated a simply albeit robust scheme based on *frustrated total internal reflection* [Han05] (**FTIR**). When light encounters an interface between media with different indices of refraction the light becomes refracted and remains within the medium with the higher index of refraction. This phenomenon is the basis for all optical light-guides. However, another material in contact with the interface between the two materials can change the index of refraction and frustrate the total internal reflection, causing light to escape the waveguide at this point.

(a) (b) (c)

Figure 6.6: (a) Multiple tradeshow visitors interacting with a custom made, multi-touch enabled tourist guide application. (b) The table in the show room at IFA 2007 in Berlin. (c) A virtual tourist guide for the city of Berlin has been developed and deployed on the multi-touch table.

The approach uses an IR edge-lit sheet of acrylic as waveguide. Fingers touching the panel cause **IR** light to escape and reflect it back through the screen where a camera equipped with an IR -pass filter detects this scattered light (Figure 6.5 c + d). Simple computer vision techniques are used to detect fingertips and multiple contact points can be tracked at interactive rates across video frames. This system caught the attention of many researchers and DIY enthusiasts alike. Due to its relative simplicity and ease of construction a multitude of devices in various form factors and for different application domains have been built based on this approach. For example, we have built our own multi-touch table based on **FTIR** together with a Berlin based company (Foresee[2]). The table was exhibited in a public showroom at the IFA tradeshow in September 2007 (see Figure 6.6). One limitation of the **FTIR** approach is the requirement for touching objects to be soft and slightly wet, such as skin. Therefore the system cannot detect objects made from hard material (e.g., a coffee mug) or detect visual markers such as barcodes. Furthermore the approach requires some form of compliant surface that allows to project virtual imagery onto the waveguide without getting in the way of the **FTIR** effect.

6.1.3 Other Tabletop Hardware

SMART's **DViT** [Mar95, SMA03] uses four IR cameras to detect the location of up to two fingertips or pens in contact with a large vertical screen. This limitation can make collaboration on the same screen difficult - from the users perspective this limitation appears arbitrary. Furthermore, contacts close to each other or right on one of the diagonals are often not tracked correctly. The device is marketed as vertical, kiosk-style display and SMART have not released software explicitly making use of the multi-touch capabilities. Nonetheless, SMART's DViT technology has been used in a number of research applications including Interface Currents [HCS06] and also our own work detailed in Chapters 3 and 4. Furthermore the DViT Technology plays an important role in the **i-LAND** project [SlH+99] and particularly the InteracTable©and the DynaWall©(see also Figure 6.7 a).

(a) (b)

Figure 6.7: (a) SmartBoards and other direct-touch enabled surfaces in the i-Land project.(b) GelForce sensing principle and visualizations of the force-field produced by tangible objects and multiple hands interacting with the sensor.

In contrast to most vision-based touch systems, which can only determine the position, shape, or area of contact, the **GelForce** system of Vlack et al. [VMK+05, KVM+05] measures the trac-

[2] http://www.foresee.biz

tion force applied to the touchpad. The system tracks a dense array of colored markers embedded in a block of clear silicone to determine the traction field. Since the system detects deformation due to both pressure on the surface as well as lateral forces, it can be used for both isotonic and isometric input. This ability makes the device appropriate for both position and rate control (Figure 6.7 b).

6.2 Liquid Displacement Sensing

In this section we describe a new technique for rapidly prototyping multi-touch and object sensing surfaces, which carries some unique properties when compared to existing approaches. It works by liquid displacement inside a malleable projection surface. A latex pouch is filled with a mixture of water and black ink. The pouch serves both as projection screen and transducer for user input. The black liquid hides the white latex surface from the camera when nothing touches the surface. Touching objects displace the liquid and press the latex onto an acrylic or glass plate placed underneath the surface. This reveals the shape of the touching object in bright white to a camera mounted behind the surface. This provides both touch and pressure information and a distinct organic quality when touched.

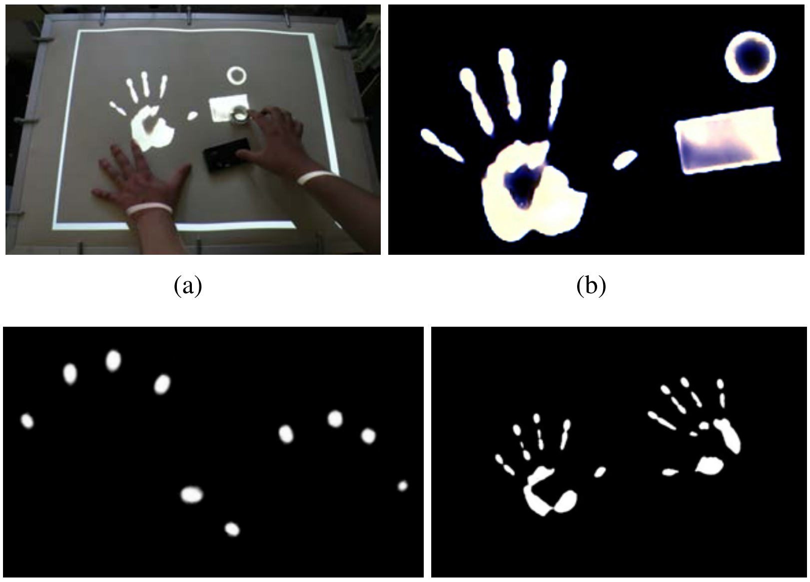
The system is easy to build and produces an extremely clean signal revealing multiple fingers, whole hands and objects that can be processed using computer vision techniques. This approach provides an even easier mechanism to build interactive surfaces than techniques such as frustrated total internal reflection (**FTIR**) [Han05], requiring no Infrared (IR) lighting, no mounting of LEDs, and no soldering to build. In this chapter we provide an overview of the technique, some of its unique capabilities, and uncover some of the trade-offs between viscosity of liquid, surface malleability, air pressure and the volume of liquid used. Our aim is to allow practitioners to build and experiment with such systems more readily.

> **Contribution Statement:** Our experiences from building and iterating several tabltop prototypes based on the liquid displacement principle are discussed in [HKI08]. I am the first author of this paper and I have initiated and driven the project. David Kim helped me in building all of the iterations and contributed many original ideas to the project. Shahram Izadi had significant input on the design of the system as well as on writing the paper.

6.2.1 LDS Principle and Implementation

Figure 6.9 gives an overview of our setup. A camera and visible light source(s) are placed underneath a sheet of acrylic or glass. In our prototype we used a 800x600mm, 5mm thick sheet of acrylic, other transparent materials such as common glass would work as well. Black liquid is poured onto this surface, and white latex rubber (approx. 0.4 mm thick), silicone sealant and an alloy frame are used to form a pouch to contain the liquid and stop it from leaking. In our experiments we used common black printer ink as the liquid.

The light is predominately absorbed by the black liquid so that almost no signal is picked up by the camera when nothing is touching the surface. Whenever objects press onto the surface, liquid is displaced and the white latex is moved toward the acrylic, causing light to be reflected at points where the user is touching. The shape of the touching object becomes visible from the underneath and can be imaged by the camera. The contrast between object contours and the background is high as light is reflected to a much higher degree by the white latex.



(a) (b)

(c)

Figure 6.8: The capabilities of a liquid displacement table. (a) the user pressing a hand, cellphone and tape roll onto the surface to interact. Here the objects cause the black liquid to be displaced as the white latex is pressed onto a sheet of acrylic. This produces a bright white imprint of the objects, which is captured by a camera placed underneath the surface. (b) the raw unprocessed images as captured by the camera. These are also projected onto the surface from above. Note that the approach requires near zero force to interact, and that objects can be sensed at speed in full motion. (c) The raw image is extremely clean and requires little post-processing; sensor data from the user pressing fingertips and whole-hands onto the surface.

As shown in Figure 6.8, from this signal minimal computer vision algorithms are needed to compute touch. Only lens correction, connected component analysis are needed to compute fingertip locations. The high contrast between touching fingers and the surrounding black liquid makes binarization, smoothing and rectification steps unnecessary. Further, the ability to capture object outlines through the surface allows contour-based algorithms such as the Sobel filter to be utilized.

6.2.2 Surface Material Properties

Building a sensor with the approach outlined above is straightforward and can be carried out with limited cost and expertise (see Figure 6.10). In this section we uncover some of the trade-offs that different material configurations can cause for the sensing fidelity. In particular, surface elasticity, air pressure, liquid viscosity and volume of the liquid. We have experimented with many of these aspects in order to find a 'sweet spot' that gives us the right balance of sensing precision, display capabilities and feel. Figure 6.11 gives an overview of the properties discussed in the following paragraphs.

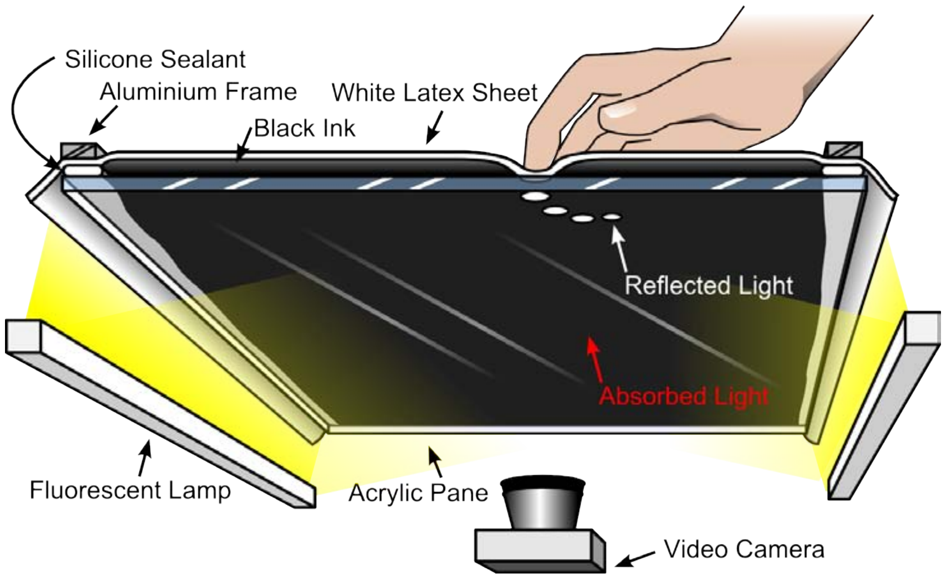
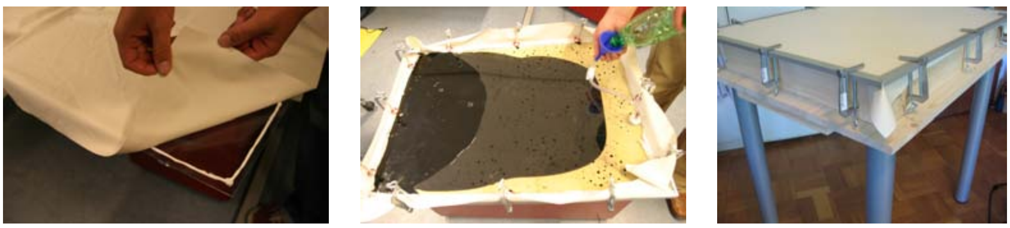


Figure 6.9: Sensing overview. Black liquid absorbs light and hides the white latex surface from the camera when nothing touches the surface. Touching objects (e.g., fingers) displace the liquid, press the latex onto the acrylic or glass, causing more light to reflect, revealing the shape of the touching object in bright white to the camera underneath.

Surface Malleability Surface malleability depends largely on the material used for the pouch and its thickness. In order to detect fingers and other objects the material needs to be relatively thin (we used 0.4 mm thick latex) and elastic. If the material is too thick or rigid, object outlines can become imprecise. Furthermore, object outlines in close proximity can become fused (Figure 6.12 (a)). Using a material that is too thin for the pouch will not block light shining onto the surface from top and therefore reduce the image contrast and sharpness of the sensor image. Further, thin or elastic material will deform according to the liquid motion and cause rippling effects distortions in the projection (Figure 6.12 (b+c)).



(a) (b) (c)

Figure 6.10: Building steps. (a) Placing the latex sheet on top of the acrylic. (b) Pouring the black liquid into the latex pouch. (c) Final table setup with aluminum frame and silicon sealed panel.

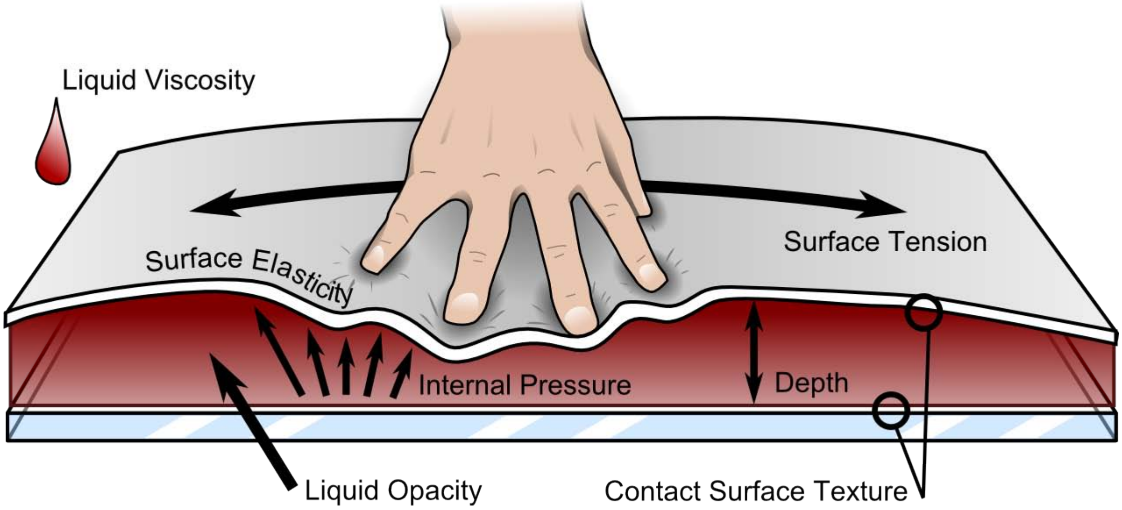


Figure 6.11: Various properties to consider when building tables based on our approach such as surface malleability and tension but also liquid type, volume and color.

Surface Tension In addition to the material qualities, surface malleability is affected by the amount of tension used in stretching the material. More tension can reduce the rippling and fusing effects. However, too much tension again reduces precision in sensing contours. Too much tension prevents the liquid from filling concave holes in or between objects, for example the archway of a palm pressed onto the table. Too little tension however, causes a delay in relaxation of the surface (deformation hysteresis) and also leads to motion blur in the camera image (Figure 6.13, a). We achieved the best results with a mild tension that allowed depressing the surface with nearly zero force but prevented the material to ripple or cause motion blur.

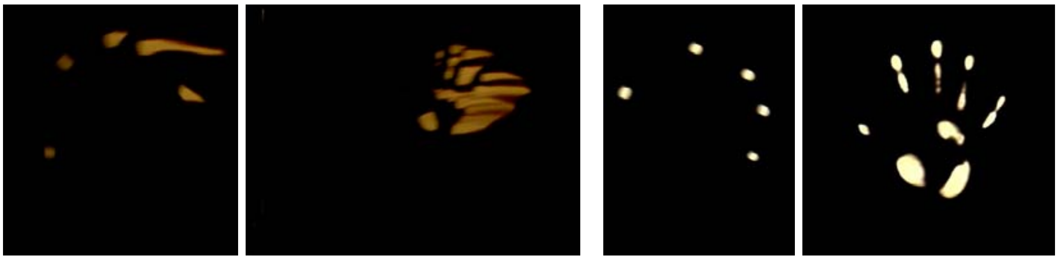
Viscosity, Color and Tint The rippling, deformation hysteresis, motion blur and fusing effects are also directly dependent on liquid viscosity. We use water dyed printer ink but other fluids are applicable as well. Also using substrates, such as gels is an option. In general terms,

(a) (b) (c)

Figure 6.12: Bridging and rippling effect. (a) Finger tips close to each other are sometimes fused together. (b) Ripples caused by a user's touch. (c) Distortion caused by rippling effect.

viscosity is the resistance of liquids to flow. High viscosity fluids (e.g., crude oil) flow slower than low viscosity fluids (e.g., water). In consequence high viscosity fluids can be used to reduce the rippling and fusing effects. However, due to their resistance to flow they can increase motion blur (Figure 6.13, a). We are still experimenting with viscosity but have found printer ink to be a sufficient first step.

Equally important as viscosity are liquid color and opacity. Using a black, completely opaque liquid has the advantage of crisp object contours and high contrast between areas being touched and the background. However, other colors and levels of opacity are permissible, which can give more detailed depth-based contours (e.g. see Figure 6.14 (a)).



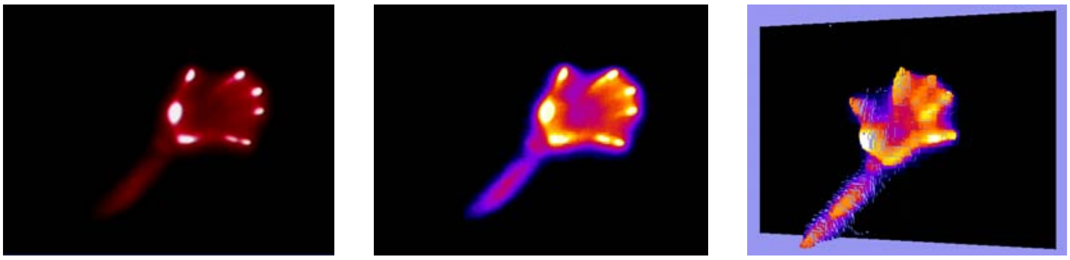
(a) (b)

Figure 6.13: Motion blur caused by adhesion. (a) Finger tips and hands in motion appear blurry. (b) Added tension in the surface and internal pouch pressure can mitigate this problem.

Liquid Volume Precision in pressure sensing and amount of derived pressure information largely depends on the volume of the liquid. The HapticLens [Sin97] project demonstrated how the liquid displacement principle can be leveraged to extract relief information of objects depressing a malleable surface (albeit not table sized). Increasing the liquid depth and implicitly the liquid volume greatly extends the ability to measure pressure as shown in Figure 6.14.

Using a thick layer of partially transparent liquid allows simple approximations of pressure using pixel intensity. However, while the resulting imagery seems promising, the increased liquid volume can lead to high deformation of the projection surface increasing the rippling effect. As described earlier, the water bed effect can be countered with higher surface tension and an elevated brim but problems with surface deformation when touched persist, especially when several objects touch the surface at once.

Internal Pouch Pressure A final parameter to consider is the internal pressure of the pouch. Pressurizing the pouch makes it slightly harder to depress the surface but increases the tendency of the liquid to fill cavities and concave elements of touching object, hence reducing fusing and motion blur effects (Figure 6.13, b). Contours of objects appear more defined as the internal pressure forces the surface material to nestle against the depressing object. Another option to control the rippling and motion blur effects is to pressurize the pouch with air in addition to a thin layer of liquid, so that an air gap persists between the liquid and projection screen. In combination with modest surface tension this prevents the surface from producing visible ripples almost entirely. Motion blur is also suppressed effectively as the internal pressure counters adhesion between the acrylic and pouch material, thus speeding up contact relaxation.



(a) (b) (c)

Figure 6.14: Effect of liquid volume, opacity and color. (a) Raw image from pressure sensitive version of our setup. (b) Mapping pixel intensity to a false color pressure map. (c) Height map generated from pixel intensity values.

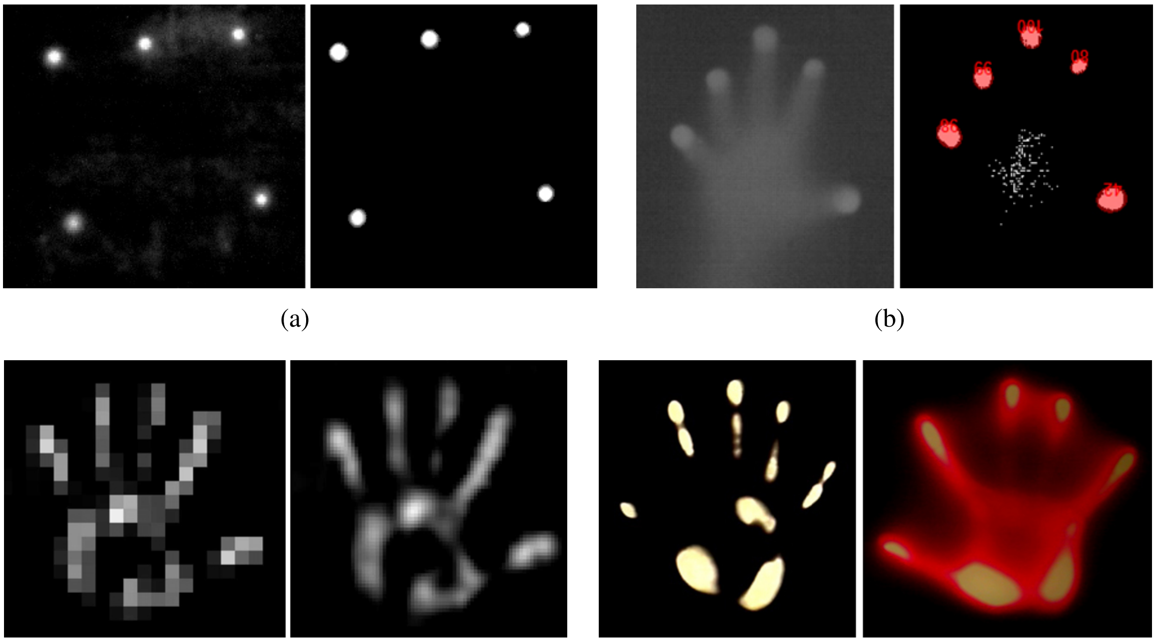
6.2.3 Summary

We have presented a new approach for rapidly prototyping multi-touch and object sensing surfaces. It works by liquid displacement inside a malleable projection surface. Our approach allows recognition of shapes and outlines of many different objects touching the surface with high precision. The approach is near zero force, and works without additional IR illumination. This frees practitioners from the need of cumbersome soldering or mounting of IR sources. Like **FTIR** we generate a high signal-to-noise ratio, producing distinctly sharp images from the camera, and easing the processing phase. Our approach has the following qualities:

- Soft and malleable surface that provides an organic feel when touched.
- Recognition of shapes and outlines of objects touching the surface with high precision.
- Works without additional IR illumination. This frees developers from the need of cumbersome soldering or mounting of IR sources.
- Provides a high signal-to-noise ratio in the image, which can be used for touch detection, contour and shape-based algorithms, and also pressure sensing. Our hope is that given the simplicity of the approach it will prove useful to researchers and DIY enthusiasts alike, allowing rapid experimentation with direct input tabletops.

There are of course trade-offs with the approach. Being a camera-based system means that significant amount of free space is required behind the panel. Furthermore, the non-segmented pouch does require the panel to be positioned horizontally. Using a liquid filled cell structure might enable other non-horizontal and non-planar designs. Finally, the current design does not allow for rear-projected setups creating extra space requirements for a projector mounted above the surface. Occlusion of projected content by users' hands becomes a problem due to this limitation as well.

When comparing the raw data (Figure 6.15) other sensing solutions discussed in this chapter we can see that our approach compliments the existing range of approaches. The signal is fairly unique, enabling advanced techniques such as pressure sensing or even some 3D shape reconstruction to be captured from imprints of hands and other objects. However, while all sensing approaches have their specifics one aspect spans across all approaches pictured in Figure 6.15: They are capable of sensing multiple fingertips, whole-hands and the outline thereof. These capabilities allow for flexible and rich interactions with the digital.



(a) (b) (c) (d)

Figure 6.15: (a) FTIR: produces a relatively clean image but requires processing. Only soft, organic objects such as fingers can be sensed. No hover or depth information available. (b) Diffuse Illumination: Raw image requires more processing but a narrow range of hover can be covered. Because light is shone through the surface physical objects reflect IR light and can be tracked. (c) Capacitive: proximity map sampled from sensor plates and image generated via bicubic interpolation. Provides accurate proximity measurements to sense objects on-surface and slightly above. Physical objects can not be tracked (at least not without augmentation). (d) Liquid Displacement Sensing: Raw data from two sensor configurations. Signal with very high signal-to-noise ratio. Arbitrary objects can be sensed including their outlines and, in the second configuration, pressure information can be estimated.

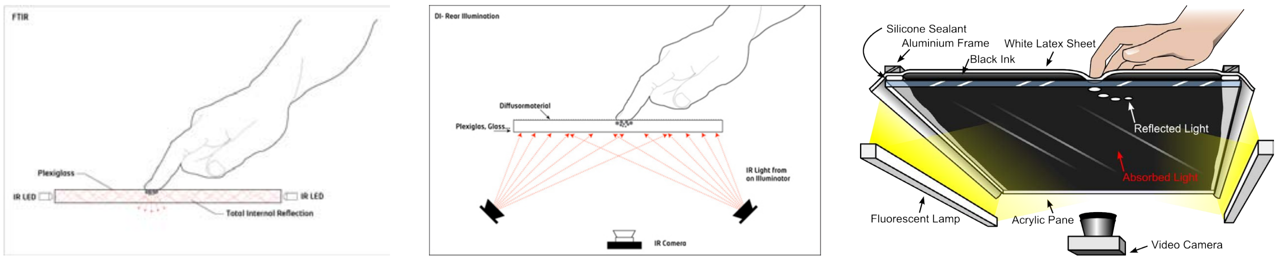
6.3 Chapter Summary

6.3.1 Pros and Cons of Sensing Approaches

As discussed earlier in this chapter, approaches to multi-touch sensing can loosely be categorized into two categories. Both categories have advantages and disadvantages inherent to their functional principle. Embedded sensing, and in particular capacitive sensing, is a reliable and durable technology due to decades of engineering and refinements (i.e., its history in single-touch devices). Capacitance sensors can be embedded in a variety of materials both opaque (e.g., **SmartSkin**) and transparent (e.g., Apple iPhone) which makes this technology suitable for screen- and projection-based setups. Furthermore, systems based on this setup can detect objects not only in contact with the screen but also from a distance (useful for applications that require a hover state). Touch/no-touch discrimination can be achieved wit a high reliability through choice of an appropriate threshold. Finally, applying a projected capacitance scheme as in **DiamondTouch** enables unique identification and discrimination of multiple users. The biggest drawback of this approach is the limited flexibility. Physical objects can not be sensed (unless equipped with conductive antennas) nor can gloved hands or fingers be sensed. Depending on the particular implementation the type of signal and information that is available for post-processing can be limiting as well. Most capacitance sensors only report the location of a contact point, sometimes enriched by pressure information, but can not provide object shape information. In summary, these approaches lend themselves best to scenarios were precision and reliability are more important than flexibility. For example, when simple (multi-)touch detection is enough.

For vision-based systems the opposite can be said. Their biggest advantage is flexibility due to a myriad of computer vision techniques that can be used to process and make sense of the raw signal. In many implementations not only points of contact can be sensed but also shape, type and even instance of objects touching the surface (using barcodes). As many systems discussed in Section 6.1.2 demonstrate, vision-based tabletops can be the basis for many applications that go beyond what we are used to from traditional user interfaces. Including the merging of physical and digital information, fluid multi-touch and whole-hand interactions, gaming and tangible interactions to name just a few. However, this flexibility comes with a variety of drawbacks that should be considered. For example, vision-based systems are in general not as robust as other touch sensing approaches because they are subject to changes in lighting conditions and other calibration issues which is an important hurdle to overcome for systems targeted at the mass market or even systems that require continuous up-time in uncontrollable environments such as public kiosks or museum installations. Camera speed, resolution and image quality all play into the equation and can be problematic. Many commodity cameras are too slow to sense rapid motion and other high-fidelity interactions (typical framerates are 30-60Hz while mice report updates at several hundred Hz). Very fine-grained interactions can be limited due to the relatively small resolution (which, to make matters worse, usually is inverse proportional to the framerate) or bad image quality (e.g., caused by poor sensor sensitivity). Also, computer vision techniques in general, have high computational costs and take away precious CPU and more recently GPU cycles from other tasks such as rendering.

In conclusion vision-based systems are better suited in applications where sensing flexibility is a key concern. Particularly when experimenting with new, non-standardized interaction techniques. If the capability to sense multiple cursors at once is all that is required or if reliable user identification is unavoidable capacitance-based systems might be the better choice.



(a) (b) (c)

Figure 6.16: (a) FTIR principle: In an edge-lit sheet of acrylic, IR light is totally internally reflected unless frustrated by a finger in contact with the medium. Scattered light is picked up by the camera underneath. (b) Diffuse Illumination principle: Projection screen equipped with a diffuser is lit by IR light sources from behind. Only objects in close vicinity to the screen reflect IR light back toward the camera. (c) Liquid Displacement principle: Black liquid absorbs light and hides the white latex surface from the camera when nothing touches the surface. Touching objects (e.g., fingers) displace the liquid, press the latex onto the acrylic or glass, causing more light to reflect, revealing the shape of the touching object in bright white to the camera underneath.

Many different interactive tabletop technologies have surfaced over the years and vision-based systems have proved particular power and flexibility, allowing for sensing capabilities that surpass those of regular touchpads (e.g., sensing of multiple fingers and other objects). Most systems discussed in this section place cameras and illumination behind the projection surface (mitigating occlusion problems). The projection screen itself usually is coated with some sort of diffuser in order to reveal projected graphics to the human eye. Optical sensing usually happens in the IR light spectrum to avoid interference from ambient lighting and the projected image. In general two illumination schemes have proved themselves particularly well suited; diffuse illumination and **FTIR** (see Figure 6.16 a,b) many examples exist in research and commercially available products begin to appear for both approaches [Mic08, SMA09, Per06]. Furthermore, we have presented an alternative approach to vision-based multi-touch and object sensing surfaces (cf. Section 6.2 and Figure 6.16 c).

6.3.2 How Hardware Matters

In the previous sections we gave a detailed overview of available hardware approaches – both from the literature (Section 6.1) and our own approach (Section 6.2). This detailed discussion of hardware solutions might seem excessive for a thesis with the goal of proposing a new model for tabletop interaction. However, the author is convinced that tabletop computing is a sub-field of human computer interaction where an in-depth understanding of available hardware choices and understanding of new developments is crucial. Never before (maybe except recent developments in mobile phone user interfaces) were hardware and software developments so interwoven as they are in interactive surface computing. For example, consider the recent transgression from single- to multi-core CPUs. While this new hardware paradigm has significant impact on the performance and also on programming paradigms – the direct impact on how users interact with a multi-core system versus a single-core equipped system has not changed (at least not as a direct consequence). In contrast, tabletop hardware advancements (or limitations) influence the way humans can interact with the digital directly and inevitably.

This begins with the size of the table itself as Ryall et al. [RFSM04] observed in a study comparing group performance in identical tasks performed on differently sized versions of the **DiamondTouch** table. But it also matters how an image is produced; while front projected systems allow for table designs that have enough space underneath the projection screen for multiple users to sit around the table and rest their legs underneath the table, they suffer from occlusion problems due to objects in the optical path (e.g., hands, heads). Back projected systems mitigate the occlusion problem nicely but require more space under the projection screen which makes them more suitable for settings in which people stand around the table. This again has an impact on how people interact with a system. One can imagine that standing around a table would encourage more lightweight interactions while seated arrangements could allow for longer more in-depth work such as reading or drawing. Finally the sensing capabilities of the chosen approach may be most important aspect. The interaction possible on a system that only recognizes one or two touch points or pens does not lend itself toward the same interaction techniques as a system that is capable of sensing multiple fingers, whole hands and other physical objects simultaneously. Equally important might be whether a system can recognize multiple contacts, like many vision-based systems, or multiple users such as the **DiamondTouch** table.

The influence is not unidirectional; while new developments appear in the hardware arena at a fast pace we as a community also continuously refine what interacting with interactive surfaces means. Therefore, we constantly refine the requirements for the design of tabletop hardware and impose new challenges for the designers of such hardware. For example, only a few years ago many doubted that being capable of sensing multiple points of contact was indeed necessary or beneficial (most techniques to do so are known for decades). The onset of products (e.g., Microsoft Surface, iPhone, **DiamondTouch**) that integrate multi-touch capabilities and graphical interfaces specifically designed for such hardware helped to create a new perspective on this issue.

The following table summarizes some of the sensing aspects to consider when choosing a hardware category or platform:

			MT	User ID	Shape Information	Proximity Information	Object Presence	Object ID
Embedded	Capacitive	Diamond Touch	✓	✓	Bounding Box	✗	✗	✗
		SmartSkin	✓	✗	✓	✓	✓2	✗
	Optical	ThinSight	✓	✗	✓	✓	✓	✓
Vision-Based	Direct	Visual Touchpad	✓	✗	✗	✓	✗	✗
		Play Anywhere	one finger per hand	✗	Motion	✓	✓	✓
	Indirect	FTIR	✓	✗	✓	✗	✗	✗
		Microsoft Surface	✓	✗	✓1	✓1	✓	✓
		DViT	2 max	✗	✗	✗	✓	✗

1: requires special software. 2: requires special tags.

Figure 6.17: Overview over most important sensing aspects of systems discussed in previous section.

Table 6.17 gives an overview over the systems discussed in the previous sections and how they perform in respect to the factors listed above. Equally important for the resulting user experience are the output capabilities of tabletop systems:

Occlusions: the display may be occluded or have shadows cast upon (e.g., front or top projected systems suffer from this problem)

Lighting: is the display still legible under various lighting conditions (projected and vision-based systems will usually work better under controlled lighting conditions)

Viewpoint independance: does the display support a large field of view and is it orientation independent (viewable from all sides)?

Resolution: is the display capable of displaying information at a reasonable resolution (e.g., impacting text legibility) - LCD based solutions will usually fare better.

Clearly these novel hardware developments enable new interaction styles that go beyond the cursor based interaction we are used to from regular desktops. Furthermore, our own explorations of digital tabletop interaction were severely constraint by the limitations of the input hardware, especially when considering our criteria of *flexibility* and *physicality*. After discussing the sensing side of the various approaches presented in this chapter, enabling the sensing fidelity we deem necessary, in the following chapter we introduce our new model for tabletop interaction that makes use of the rich kind of input data provided by vision-based interactive surfaces.

Chapter 7

Bringing Physics to the Surface

Up to this point we have discussed existing approaches to tabletop interaction. From the literature we have distilled two particularly popular interaction styles; gesture-based and tangible interaction. Studying our own prototypes based on these paradigms in Chapters 3, 4 revealed several issues and short-comings of these approaches. Foremost, the lack of flexibility which sometimes frustrated users. We have also observed that limitations inflicted by the hardware was a limiting factor in many situations. We could clearly observe that, especially novice, users wanted to interact with the system in richer ways than what could be sensed.

In Section 6.2 we proposed our own tabletop prototype capable of sensing rich user input. Furthermore, both the research community [MR97, Rek02, Wil05, Wil04, Han05] and industry [Mic08, DL01, SMA09] have demonstrated and sometimes productized setups capable of similar sensing fidelity. In this Chapter we want to explore how this rich sensing data can be applied to novel interaction techniques and ultimately propose a new model for tabletop interaction.

In our earlier explorations (cf. 3.5) we could observe how the more physical inspired gestures such as flicking of post-its were received more favorable (and memorized better) than the iconic, abstract gestures. We have also experienced how the scripted nature of these "pseudo-physical" interactions can be problematic. Users that were under the assumption that virtual objects behave according to the laws of physics or at least similar to real world objects often tried to perform other physical manipulations – these had not been anticipated by the developer in advance and as result were not possible. Therefore the main goal of this chapter is to develop an interaction model that allows for interactions with the virtual realm through manipulations similar to those performed on real world objects without the limitations imposed by scripted or pre-defined behavior of on screen objects.

Recently, sophisticated physics simulation packages have become accessible and found widespread use in many 3D computer games. These physics simulations are capable of modeling complex mechanical structures and model their behavior realistically in 3D graphical applications. In our interaction model we make use of the capabilities of these physics simulations and explore the intersection of rich surface input data and virtual worlds augmented by realistic, open-ended and non-scripted behavior of virtual objects.

Modeling rich 2D sensor data within a 3D physics simulation is non-trivial and we highlight and discuss several of the challenges we encountered when developing our interaction model. Based on our previous experiences we wanted our model to support the following aspects:

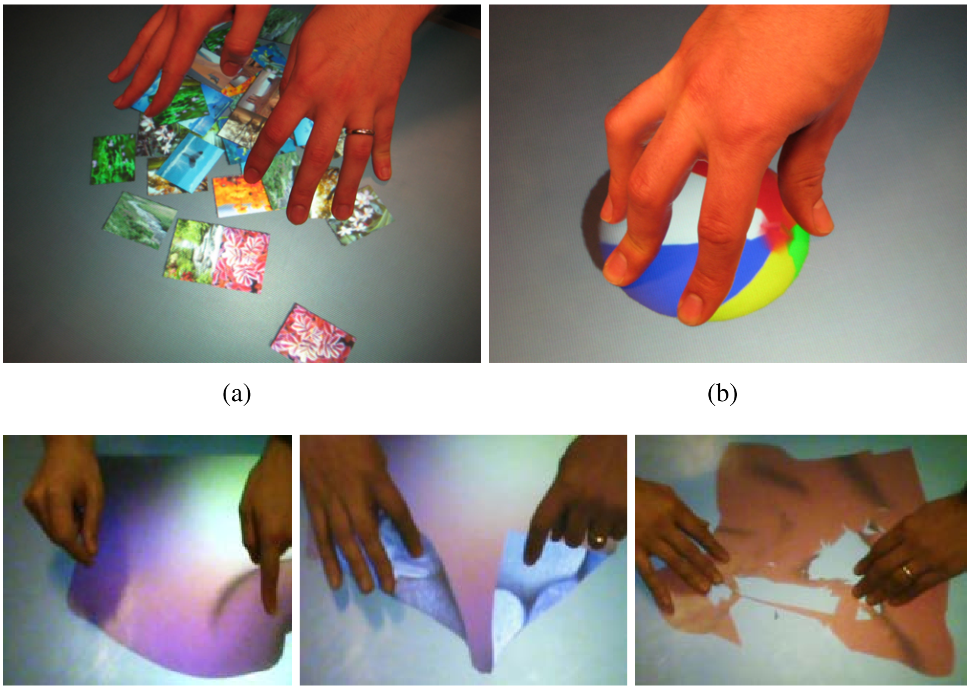
- Enable rich physical gestures through manipulations similar to the real world.
- Using sophisticated physics simulations to add real world dynamics to virtual objects and enable users to interact through the exertion of forces such as friction and collisions.
- Support of multiple simultaneous contact points (not just fingertips) and and also real (tangible) objects.
- A technique that works within the bounds of the physics simulation and makes use of the sophisticated constraint solver build into many available software packages.
- A generic model that works with different virtual objects irrespective of their shape or material (e.g., boxes, spheres, cloth)

This enables a variety of fine-grained and casual interactions, supporting finger-based, whole-hand, and tangible input. We demonstrate how our technique can be used to add real-world dynamics to interactive surfaces such as a vision-based tabletop, creating a fluid and natural experience. Our approach hides from application developers many of the complexities inherent in using physics engines, allowing the creation of applications without preprogrammed interaction behavior or gesture recognition.

> **Contribution Statement:** The evolution of the interaction model described in this chapter has been published as peer reviewed full paper [WIH+08]. Most of this work has been done at two Microsoft Research labs in Cambridge, UK and Redmond, USA. Many of the discussed models have been designed and implemented while I was an research intern in the Cambridge Lab. Together with my manager Shahram Izadi I designed and implemented a version of the direct forces (cf. 7.2.1), joints (cf. 7.2.2) and discreet-proxy (cf. 7.3.1) model. The *particle proxy* model was designed and implemented by Andy Wilson in Redmond (initially independent). Armando Garcia-Mendoza re-implemented large parts of this work to produce a final version incorporating all variations of the technique. All co-authors contributed to the design, execution and evaluation of the study as well as to the submitted paper.

7.1 Motivation

Emerging interactive surface technologies allow users to interact with the digital world by directly touching and manipulating on-screen content. People who use such systems often comment that this ability to touch digital content adds a physical or tangible quality to the interaction, making the virtual feel more real. Many interactive surface -based applications attempt to further highlight this “pseudo-physicality” by carefully designing interface objects that exhibit a sense of real-world behavior. One example is the common rotate and translate behavior found in many table-top applications, where the interaction is analogous to moving a sheet of paper on a flat surface with one or more fingers [KCST05, LPS+06]. Although such an interaction may feel realistic, it is very much preprogrammed. For example, the way in which a digital photo rotates or translates is defined by specific interaction logic. Such techniques thus possess an inherent scripted nature which may break down once the user interacts with the system in ways unanticipated by the developer.



(a) (b)

(c)

Figure 7.1: Some examples of physics-enabled interactions supported by our technique. (a) Gathering and moving multiple objects using both hands and multiple fingers simultaneously. (b) Interacting with a ball through collision and friction forces. (c) Folding a cloth-like 3D mesh and tearing it apart.

In this Chapter we take a different approach for supporting pseudo-physical interactions on surface technologies – systems that are often capable of sensing multiple touch points and possibly even more sophisticated shape information. Specifically, we utilize physics engines used in computer games that simulate Newtonian physics, thus enabling interaction with digital objects by modeling quantities such as force, mass, velocity, and friction. Such engines allow the user to control and manipulate virtual objects through sets of parameters more analogous to those of real-world interactions.

While physics engines are comprehensive, they are also complex to master. Many coefficients and parameters are exposed to application developers, and controlling the simulation via user input is non-trivial, particularly when considering more than a single contact point. We present a simple yet powerful technique for modeling rich data, sensed from surface technologies, as input within the physics simulation. Our approach models one or more contact points, such as those sensed from a multi-touch surface [DL01, Han05], and also scales to represent more sophisticated shapes such as outlines of the whole hand or tangible objects on the surface [Mic08, Wil05, Rek02]. This allows the user to interact with objects by friction forces and collisions, but avoids exposing users and application programmers to the complexities of the physics engine.

We demonstrate the applicability of the technique using a commercially available games physics engine and a prototype vision-based interactive surface. We highlight some of the interactions such an approach affords, for example gathering multiple digital objects or fine control of a virtual ball using friction and collision forces, as shown in Figure 7.1. Our system creates natural and fluid physics-based interactions "for free" – i.e., without the need either to explicitly program this behavior into the system or to recognize gestures. We also demonstrate the ability of advanced physics simulators to enable user interaction with more complex materials such as soft bodies and cloth, as shown in Figure 7.1.

We have experimented with various alternatives for simulating surface input within the physics world, the trade-offs of which are discussed in this chapter. Our aim is to allow practitioners to understand the nuances of these alternatives so that they may further explore the intersection between interactive surfaces and physics. We also discuss both early experiences using our technique and some of its limitations.

7.1.1 Related Work

Much work has recently been published on interactive surfaces, and particularly on direct input tabletops [DL01, Han05, Wil05]. Some of this work examines physics like interactions. For example, Kruger et al. [KCST05] explored simple notions of friction and motion to allow an object to be translated and rotated using a single point of contact. Other systems [Gei98, RGS+06] have implemented gestures such as flicking, throwing, and pushing to add velocity and inertia to on-screen objects. Beyond tabletops, a number of interactive systems have explored the *physicality* of objects as a principle around which interaction is organized. Electronic files stack like real documents [MSW92], move out of the way as if they have a solid form [RCL+98], peel back like paper [Dra04], and move as if alive [CU93]. Rather than employing a full physics model, such systems use pseudo-physics in minimal ways to support subtle interaction possibilities. The present work looks to address a richer set of physics-enabled behavior, and focuses on problems related to input.

Various attempts to provide richer and more realistic 3D interactions have also been explored in the context of interactive tabletops [GW07]. Ståhl et al. [SWS+02] describe a tabletop where objects float to the surface when accessed and sink back to the ground when no longer used. Hancock et al. present a set of methods to compensate for off-axis viewing [HC07] on multi-user tabletops, as well as techniques for *shallow-depth* interaction [HCC07]. These systems allow basic manipulation of 3D objects but do not model interactions based on a physics simulation.

Realistic dynamics simulation has a long history in graphics and animation communities. Baraff [Bar89] gives an overview of the main concepts for calculation of rigid-body dynamics. More recently, real-time collision detection and response with stable friction calculations became feasible [Bar94]. Erleben et al. [ESHD05] provide an overview of current techniques and advancements that led to the development of sophisticated physics engines for simulation and gaming, such as **PhysX**, **Havok**, **Newton**, and **ODE**. However, these techniques have yet to reach the user interface and interaction research communities broadly.

One notable exception is **BumpTop** [AB06], which uses a physics engine to add real-world dynamics to the Tablet PC desktop. It supports notions such as collisions, mass, and inertia, and higher level constructs such as piling. The work nicely demonstrates some of the capabilities of a modern physics engine. However, the approach is based on a single point of input and menu based selection. Compared to the rich means available for manipulating physical objects, this single-point input model can be limiting.

Multiple inputs are considered in the application of an early physics engine by Baraff with the **Responsive Workbench** [FTB+00]. This system simulates a 3D bimanual assembly task using two 6 DOF input devices and a stereoscopic display. An object may be manipulated with one hand by placing eight springs connected to the corners of a virtual cube rigidly attached to the user's hand. Bimanual interactions are supported by the superposition of forces from both hands. Thus, when one hand is released, the object snaps to the position and orientation of the other. Interactive surface s, particularly vision-based systems [Mic08, Wil05, Han05, Wil07], allow the capture of rich sensor data that not only includes multi-touch data and detailed shape information, such as images of the users' entire hands or other tangible objects near the surface. We present a technique for modeling this rich and diverse sensor input, representing these effectively as friction and contact forces in the physics simulation. These capabilities augment and extend the single contact model utilized in **BumpTop** and the bimanual approach in **Responsive Workbench**, supporting single-touch, multi-touch, contour-based, and tangible input using a single technique.

7.1.2 Interactive Surface Input

A contact on an interactive surface (e.g., a fingertip touching the surface) is most easily represented as a discrete 2D point. In the case of vision-based interactive surfaces, neighboring sensor pixels are usually grouped into continuous regions or connected components [CCL04], with the idea that each component corresponds to a contact. The center of the component is then easily calculated. This approach thus reduces each contact to a point-based representation, regardless of shape.

This point representation of contacts allows application developers to think in terms of familiar point rectangle hit testing algorithms typical of traditional cursor-based systems, but it imposes significant limits on interaction. First, point based hit testing may fail to catch the user touching a virtual object if the contact is not compact, as when the user places a large part of their hand on surface. In this case, the center point may lie outside the target object. Secondly, tracking point-based contacts to deduce motion can lead to difficult problems related to correspondence. For example, consider two fingers that move so near to each other that they now appear as a single contact. The choice of which of the original contacts to eliminate can result in different motion interpretations. Finally, reducing contacts to points prevents users from drawing on the full spectrum of manipulation styles found in everyday life. Consider the multiple grasping strategies illustrated in Figure 7.2, for example. Each gives a different feeling of control in the real world. Ultimately, it seems that point-based systems encourage the exclusive use of index fingers on interactive surfaces.

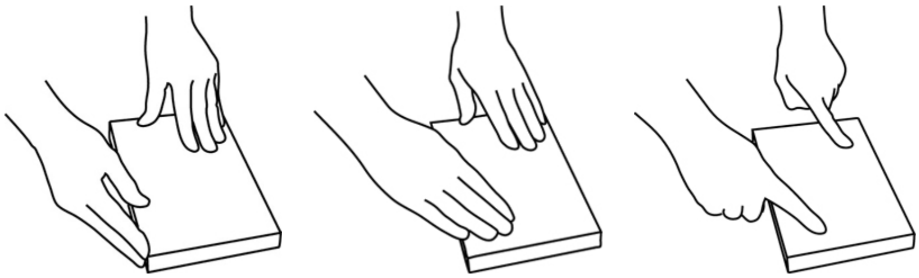


Figure 7.2: Different grasping strategies to rotate an object such as a book resting on a surface.

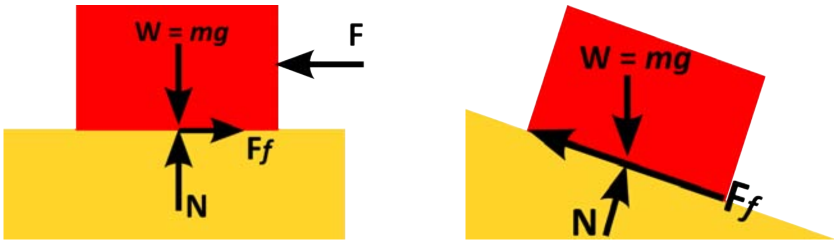
One approach for preserving more information about contact shape is to determine the bounding box of the contact, or the major and minor axes of an ellipse model that approximately fits the shape. These approaches work well for compact contacts (e.g., fingertips) and certain hand poses [WB03], but less so for complex shapes and their motion. Alternatively, the shape may be represented more precisely as a polygon mesh by calculating each contact's contour, represented as a closed path [GW06]. Another technique is to take pixels lying near the contour by computing the spatial gradient using a Sobel filter [GW06].

These approaches allow us to support even more sophisticated representations of user input. We would like to combine this broad fidelity of input with advanced physics simulations to expand the vocabulary with which we can manipulate digital objects. Our aim is to make manipulation of digital objects less scripted, using rich and varied interaction techniques and strategies.

7.1.3 Physics Simulations

Today's physics engines enable the creation of real-world mechanics and behavior in graphical applications while hiding computational complexity. They employ many physics concepts such as acceleration, momentum, forces, friction, and collisions. In addition to rigid bodies, many systems model particles (for smoke, dust, and so forth), fluids, hair, and cloths. Virtual joints and springs give "rag doll" characters and vehicles appropriate articulation, and materials can be programmed with specific properties so that ice is slick, for example. The present work primarily concerns contact forces, such as those due to collisions and friction between simulated bodies.

The handling of collisions is typically divided into collision detection, the determination of whether two rigid bodies are in contact, and collision response, the application of appropriate forces if they are in contact. For example, the collision of a cube falling on the floor may be detected by considering the intersection of the faces defining the cube with those of the floor. The change in motion of the cube as a result (the response) is a function of mass, inertia, velocity, the point of contact with the floor, and other factors.



(a) (b)

Figure 7.3: The effect of friction forces (a) Dynamic friction: N is the normal force preventing the red block from penetrating the floor; W weight force with which the block is pushed downwards by gravity; F is a force exerted onto the block to move it to the left; F_f is the friction opposing this motion because the magnitude $F_f < F$ the block moves. (b) Static friction F_f prevents the red block from sliding down the slope.

Friction forces resist motion when the surface of one body stays in contact with the surface of another. If two surfaces are moving with respect to each other, kinetic friction opposes the forces moving the bodies (Figure 7.3 a). If two surfaces are at rest relative to each other, static friction opposes forces that would otherwise lead to the motion of one of the bodies (Figure 7.3 b).

7.2 Surface Input within a Physics Simulation

In order to interact appropriately with virtual objects in a physics engine, surface contacts must be represented within the simulation. These engines have enormous potential and flexibility. Accordingly, there are many strategies for modeling surface input in the physics world. We briefly describe these strategies here, and give more detail later.

Direct forces: A force is applied where a contact point touches a virtual object. The force direction and magnitude is calculated from the contact's velocity and size if available.

Virtual joints and springs Each contact is connected to the virtual object it touches by a rigid link or spring, so that the object is dragged along with the contact.

Proxy objects: Contact points are represented as rigid bodies such as cubes or spheres. These bodies are an approximation of the contacts, and interact with other virtual objects by collisions and friction forces.

Particles: Where additional information about a contact's shape is available, multiple rigid bodies – or particles – are combined to approximate the shape and motion of the contact more accurately. This allows for better modeling of interaction with the whole hand or other contacts such as tangible objects.

Deformable 2D/3D mesh: Another approach for modeling more sophisticated shapes is to construct 2D or even 3D meshes if appropriate sensors are available.

It would seem that a deformable 3D mesh of the hand would achieve the highest degree of fidelity. But a number of difficulties exist with this approach. First, most interactive surfaces provide sensing at or near the surface only, not full 3D shape. Similarly, because the manipulated object exists only on the (flat) display surface, the 3D shape of the hand, if captured, would not conform to the object and so would not reflect the shape of a real hand grasping a real object. Finally, constructing such an animated mesh is difficult, requiring robust tracking of features and accurate deformation of the 3D object. Therefore, initially we do not consider this as an option in our exploration of the presented technique. However, in Chapter 9 we discuss two setups that approximate this "ideal" solution more closely.

That leaves us with a key question that motivates the rest of this chapter: How does one best use surface input to interact with advanced physics simulations in useful ways? We describe our rationale and experiences in implementing and evaluating the above alternatives. The main contribution is a novel interaction technique that retains most of the benefits of mesh-based representations – in particular, a high fidelity of interaction – but is considerably easier for application programmers to implement.

7.2.1 Applying Forces Directly

A typical strategy for moving an object on an interactive surface in response to touch is to continually update its position to match the touching contacts' position. Generally we will refer to this manner of moving objects by setting its position and orientation directly generally as kinematic control.

Within a physics simulation, however, the most common way for an application to control the movement of a rigid body is to apply one or more forces. For example, a spaceship in a game might have thrusters on either side of its body. The ship may be propelled forward by applying forward force at the location of both thrusters. If one of the forces is applied in the opposite direction, the ship will turn. Rotation is the by product of torque, which occurs when forces are applied off-center (of mass) because different "parts" of the body are moving at different speeds.

From a programmer's point of view, this approach is very different than moving the ship by setting its position. To effect kinematic control within a physics simulation, we must calculate the precise force and torque required to move the object into its target position (Figure 7.4 a). This method of positioning an object ensures correct collision response with other bodies in the simulation. In comparison, directly setting the position of the body within a simulation can lead to unstable and unpredictable results. Absolute positioning might be analogous to teleporting a real object from one location to another. Issues such as interpenetration whereby objects become partially embedded in each other, can arise. A natural strategy for moving an object to follow a contact on an interactive surface is therefore to consider that each contact imparts a friction force to the body it touches according to the contact's motion (and presumed mass). These multiple friction forces may be applied to the body, as in the example of the spaceship. Unfortunately, to calculate the forces necessary to match a contact's movement, all other external forces acting on the body must be taken into account and counteracted (Figure 7.4 b+c). These may include friction forces and collision responses that are difficult or impossible for application developers to obtain.

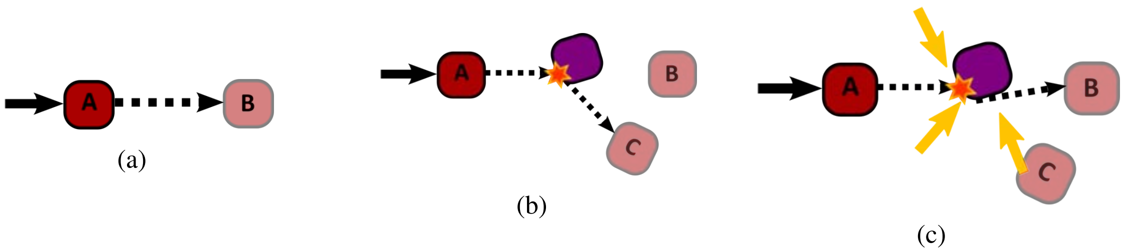


Figure 7.4: Applying forces directly. (a) To move an object from A to B we must apply a force of correct magnitude to the object. (b) Hidden (from the developer) forces such as collisions with other objects can lead to unwanted results. (c) To calculate counteracting forces all other external forces need to be taken into account. This can be complex and sometimes force values are difficult to obtain.

This difficulty extends to considering forces corresponding to surface contacts. In the case of multiple contacts, the correct friction forces corresponding to each contact must be determined simultaneously. Consider the case where one or more of the contacts exhibits static friction. Recall that static friction exerts a force that counteracts forces that would otherwise lead to a body's motion. For example, if one contact "pins" an object so that it will rotate due to the motion of another contact (e.g, Figure 7.2, a), the application of correct friction force due to one of the contacts requires knowing the friction force due to the other.

In fact, at the heart of any physics engine is a sophisticated constraint solver that addresses this very problem. Without essentially constructing a new solver within the physics engine, or without access to internals of the existing solver, it would seem impossible to correctly apply contact forces directly. Even if it were possible to change the solver or embed another, such an approach would go against the spirit of the present work, wherein an existing full featured physics engine is leveraged rather than built from scratch.

One possible solution is to treat all frictions as kinetic. But this poses a problem in the "pinning" example. Because kinematic friction forces only act in the presence of relative motion, the counteracting force that keeps the "pinned" part of the object stationary must first move. Thus, this approach results in a somewhat viscous and slightly unpredictable feel when moving objects.

7.2.2 Connecting to Objects with Joints and Springs

Another kinematic approach, used in systems such as BumpTop [AB06], is to connect virtual objects and an input contact using a joint. Think of this as an invisible piece of rope of predefined length that is tied to the object at a particular anchor point. The object is then pulled along using this rope.

By attaching a joint off-center, the object is subject to both force and torque – allowing the object to move and rotate using a single connection. In our earlier pinning example, one joint attaching a stationary contact point to one corner of the object would serve as a pivot point. A second joint attaching a second moving contact point to an opposing corner would cause the object to spin around the first contact point.

This approach is not well suited for multiple simultaneous contact points, particularly those pulling in opposite directions. While in the real world, multiple contacts pulling in opposite directions on an object would result in the fingers sliding, or the object deforming or tearing, neither behavior is supported by joint constraints on a rigid body. It is thus easy for multiple rigid constraints to overconstrain the simulation, resulting in numerical instability and unpredictable behavior.

Springs can in part alleviate some of these issues by providing more flex in the connection. However, a trade-off exists between the elasticity of the spring and how responsive the connected object is to contact motion (springs should be fairly short and rigid to allow for a faster response). Problems of numerical stability and uncontrolled oscillations are likely [10]. Another approach is to allow the joint or spring to break in these situations, but this can easily lead to situations where objects fly out of the user's reach.

7.3 Setting the Scene For a New Technique

We have so far described two techniques that one would typically employ in single-point physics-enabled applications, and discussed the limitations of both in terms of modeling multiple contacts. The modeling of such input is challenging but only part of the story with respect to the limitations of these approaches.

First, as we described earlier, contacts are not always discrete 2D points, and it may be desirable to match the shape of the contact input closely. It is unclear how one would model more sophisticated shapes and contours with either of these initial approaches. Second, the above techniques address the case where the user touches the object directly, thereby moving the object by friction forces. Neither of these approaches addresses the movement of objects by collision forces, i.e., from contact forces applied to the side of the object (as in Figure 7.2, b).

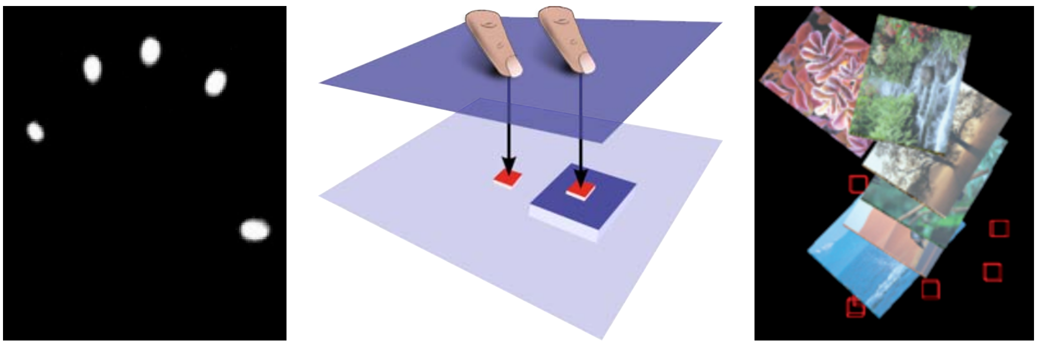
The next section presents a technique which handles friction and collision forces in the same framework and is easily extended to handle shapes of arbitrary contour. In doing so, it addresses many of the difficulties of the previous techniques.

7.3.1 Discrete Proxy Objects

The idea of *proxy objects* is to incorporate into the physics simulation a rigid body for each surface contact. These bodies are kinematically controlled to match the position of the surface contacts and can be thought of as incarnations of contact points within the physics simulation. Because they are regular rigid bodies, they may interact with other rigid bodies in the usual ways: either by collision or friction.

The proxy approach carries various benefits such as hiding the complexity of force calculations (in fact, hiding almost all physics aspects) from the programmer, while avoiding the difficulties of the previously described approaches. It leverages collision as well as friction forces (both static and kinetic) to model rich interactions such as pushing, grabbing, pinching, and dragging. Proxy objects interact with other objects in the simulation through the means provided by the engine. Finally, this approach avoids unnecessary strain on the solver (e.g., inserting extreme force values) and resulting unstable simulation states.

Proxy objects are created and positioned for each point of contact. Most simply, a single shape primitive such as a cube or sphere may be used for each contact. When a contact initially appears, a ray casting calculation is performed to determine the 3D position of the proxy so that it touches the underlying object, as shown in Figure 7.5. An interesting alternative to using a sphere or cube as a proxy shape is to create a thin capsule, box, or cylinder which stretches from the 3D camera near plane to the surface of the touched object (see Figure 7.6). This kind of proxy will collide not only with objects resting on the same plane as the touched object (or "floor"), but also objects that are in mid-air, or stacked on other objects. Such behavior may correspond more closely to user expectations.



(a) (b) (c)

Figure 7.5: Positioning of *proxy objects* works as follows: (a) Foreach surface contact a discrete *proxy object* is created. (b) A ray-casting operation returns intersection points with other virtual objects or the ground plane. (c) Proxy objects are positioned at these intersections. Surface motion is mapped to lateral motion of the *proxy objects*. Proxy objects interact with other virtual objects through collision and friction forces.

As the sensing system provides updates to a contact position, the corresponding *proxy object* is kinematically controlled to match the updated position. This is done, as described earlier, by applying the necessary forces to bring the proxy object (of known mass) to the updated position of the contact. This scheme allows users to leverage collision forces to push objects around or grab objects by touching them from two opposing sides.

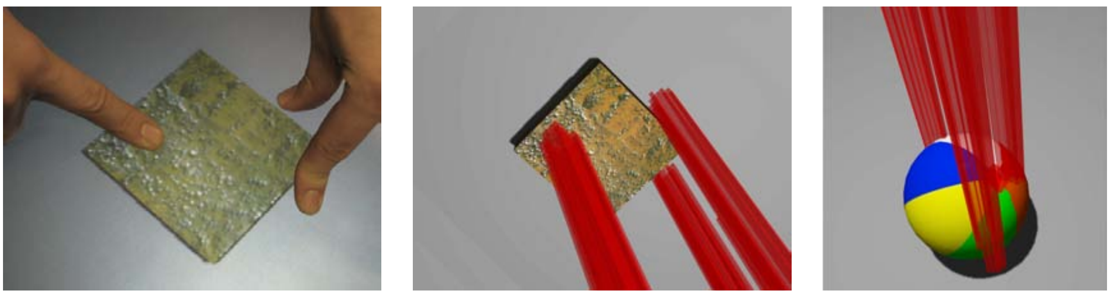
A small change in the kinematic control enables the *proxy object* to exert friction forces when it lies on top of another rigid body (as when the user touches the top surface of a virtual object, for example). In particular, only forces tangential to the touched object are applied to match the contact position. As with regular dynamic bodies, gravity is still included as an external force. In the case where gravity is directed into the surface, the proxies thus exert downward force onto other objects and cause friction forces. This hybrid kinematic-dynamic control of the object can be implemented by direct addition of forces to the proxy rigid body, or by a prismatic joint constraint on the body's motion. The simulated weight of the finger on the object may be adjusted by changing the mass of the *proxy object*, while the material properties of the virtual objects may be adjusted by changing static and kinetic friction coefficients.

The main advantage of the *proxy object* technique is that it leverages the built-in capability of the physics engine to simulate both friction and collision contact forces. Most significantly, because the calculation of contact forces is handled entirely by the built-in physics engine solver, the combined effect of simultaneous static and kinetic friction forces due to multiple *proxy objects* is handled correctly. These friction forces enable users to translate and rotate objects (through opposing forces) that they touch directly.

7.3.2 Particle Proxies

Thus far we have approximated each touch point as a single *proxy object*. This permits a simple, fast implementation, and lends itself to sensing systems that report only contact position and no shape information, as well as applications that favor interaction with the fingertip or stylus.

Some interactive surfaces provide shape information, such as an oriented ellipse, bounding

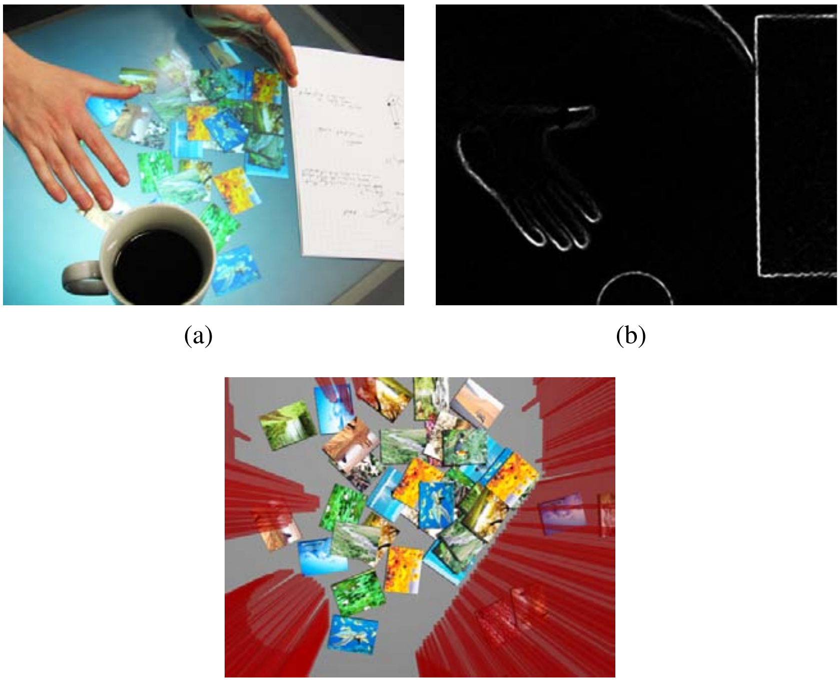


(a) (b) (c)

Figure 7.6: Particle proxies approximating the shape of various objects: (a) Interaction on the surface; applying friction from the top and collisions from the side to grip a virtual block. (b) Screenshot of the 3D scene. Long red objects are multiple particle objects approximating the shape of the surface contact. (c) Particle proxies accommodate arbitrarily shaped objects including non-flat objects such as this sphere.

box, or full polygonal contour. The idea of the *particle proxy* is to model the contact shape with a multitude of proxy objects ("particles") placed along the contour of the contact (see Figure 7.6). Particles are added and removed as contours change size and shape. A practical implementation involves creating a new set of proxy objects for the contour at the beginning of each simulation frame, and destroying all proxy objects after the physics simulation has been updated. Even though the proxies will be destroyed after the physics update, each enacts collision and friction forces during the update.

The advantage of the *particle proxy* approach is twofold. First, collisions appear to be more correct because they more closely follow the shape of the contact. This is particularly important when using the flat or side of the hand, tangible objects, or generally any contacts other than fingertips (see Figure 7.7). Similarly, the distribution and magnitude of friction forces on the top of an object are more accurately modeled. For example, the flat of the hand may exert more friction than the tip of a finger (Figure 7.2, c) by virtue of having more particles assigned to it. Likewise, a single contact turning in place can exert friction forces to rotate an object. Unlike the single proxy model, each particle is placed (ray cast) separately, so that a contact can conform to irregularly-shaped 3D virtual objects (Figure 7.6).



(a) (b) (c)

Figure 7.7: Particle proxies method overview. (a) Photograph of user interaction; shown are two hands a notebook and a cup. (b) Sobel image shows contours of surface contacts. (c) For each pixel on the contacts' contour a particle (red) is projected into the scene. Particles interact with virtual objects through friction and collision forces.

As in the single proxy object model, each particle is kinematically controlled to match the movement of the contact to which it belongs. Generally, the velocity of a point on the contour can be computed by examining the contact's contour in the previous frame. This calculation may be simple, as with an ellipse model, or more complex, as with a polygonal contour.

7.3.3 From Tracking to Flow

One difficulty in basing velocity calculations on tracked contacts is that tracking can fail, particularly when the user is using less constrained grasping postures such as the edge or flat of a hand rather than the more cursor like index finger. In these cases, components can split and merge in ways that do not correspond to how we see the physical input, leading to erroneous velocity calculations, and ultimately in the case of our physics simulation to unpredictable motion. An alternative approach is to calculate the motion of the particle independently of any tracked contact information. For example, local features of the image may instead be tracked from the previous frame to calculate velocity. Simple block matching of the sort used in optical flow [BFB94] is one such technique (see Figure 7.8).

When using local motion estimates, the tracking of discrete contact objects and exact contours may then be avoided altogether by placing proxy particles at image locations with high spatial gradient (e.g., Sobel filter [GW06]). These pixels will lie on contact contours. The *particle proxy* technique is summarized as:

```
compute Sobel image from surface input
for each pixel with high spatial gradient:
   ray cast into scene to determine initial particle position
   add particle rigid body to physics simulation
   compute contact motion at particle (e.g., from flow)
   compute corresponding tangential motion in scene
   apply force to particle to match scene motion
   apply downward force (gravity) to particle
update physics simulation
destroy all particle rigid bodies
```

The instantaneous, piecewise nature of the shape and motion calculations of the flow based *particle proxy* method possesses important advantages. First, the friction and contact forces lead to more stable physics simulation results than if shape and motion were calculated from discrete tracked objects. Second, because the technique makes few assumptions regarding the shape or movement of contacts, it imposes few limits on the manipulations a user may perform, whether leading to collisions, friction forces, or combination thereof.

7.4 New Physics-Based Interactions

Our goal in introducing more detailed surface input types into a physics simulation is to enable a wide variety of manipulation styles drawn from real-world experience. While we have only

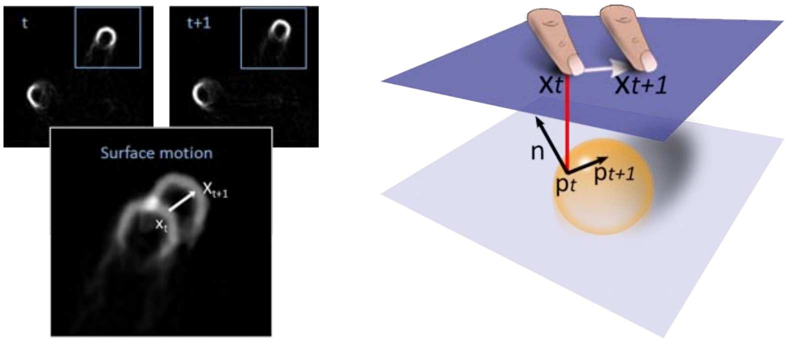


Figure 7.8: Computing flow of a particle. Surface motion at point x_t is computed by comparing successive edge images. Corresponding tangential motion in the scene is calculated by projecting image point x_t into the 3D scene, to obtain point p_t . Point p_{t+1} is found by projecting image point x_{t+1} onto tangent plane formed by normal n and point p_t. For brevity only one particle is shown while a fingertip in contact with surface would be approximated by many.

begun to explore some of the possibilities that these techniques afford, here we consider a few which we believe are noteworthy.

Manipulation Fidelity The ability to exploit detailed shape and motion information has broad consequences when considering the manipulation of even the simplest objects. Free moving virtual objects can be moved by any one of a variety of strategies that combine collisions against the contours of hands and fingers with static and kinetic frictions. Because all three kinds of forces can be employed simultaneously, the overall impression is one of unusually high fidelity. An interesting example is the manipulation of a ball that is free to roll on the surface: it may be compelled to roll, spin, stop, or bounce in a surprisingly precise fashion, using a single light touch, multiple touches, or the flat of the hand for stopping power (Figure 7.1). Physical objects can also be integrated at no cost, allowing a variety of interesting tangible behaviors (see Figure 7 for some examples).

The ability to sense and process contours, as well as distribute friction forces piecewise across the virtual space, enables the manipulation of many objects at once, much as one might move a group of small objects spread across a table (see Figure 7.1, a). Users may use the edges of their hands (or even arms) to collide against many objects at once, or use the flats of multiple hands to apply friction forces. For interactive surfaces able to sense physical objects, an interesting possibility is to use a ruler to move and align multiple objects.

3D Manipulations Modeling virtual objects and input in 3D enables interesting yet familiar interactions. For example, a flat object resting on a larger flat object may be moved by tapping its side or applying friction. Depending on the masses and frictions involved, it may be necessary to hold the larger object in place. It is thus important for the designer to tune masses, frictions, and appearances to match user expectations.

(a) (b) (c)

Figure 7.9: Interactions enabled by our model. (a) Sliding cambered cards on top of each other by pushing on by the side and holding the other in place. (b) A physical card is used to gather several pieces at once. (c) Pinning down a virtual cloth with a wooden torus.

If the interaction is limited to collision forces from the side and friction forces from the top, however, the manner in which a user may place the smaller object on top of another is unclear. Ramps, seesaws, and other constructions are possible, if somewhat contrived. In certain cases it may be possible to flip one object onto another through the application of sufficient friction forces to one side of the object.

When the objects to be stacked are thin, such as cards representing documents [RCL+98, AB06], one approach is to give the top and bottom surfaces of each object a cambered shape that allows the user to raise one side by pressing down on the other. The user may then move another like sized card under the tilted card (Figure 7.9, a). This behavior corresponds to our awareness that in the real world even "flat" objects such as cards and paper have some 3D shape that is often intuitively exploited to manipulate them.

Cloth and Soft Bodies We have used rigid bodies such as boxes and spheres to explain our interaction techniques. However, in the real world many objects are not rigid but are instead soft, malleable, and can deform or dissolve when forces are exerted on them. Examples include rubber, cloths, and paper.

In addition to rigid body dynamics, most available physics simulations offer some form of support for soft body, cloth, and fluid simulation. As all interactions in our model are conducted through collision or friction forces, the model can be applied to arbitrary virtual objects. For example, it is possible to crumple a piece of cloth with a grasping interaction using all the fingers of one hand. The crumpled cloth can then be straightened by pushing down with the flat hand.

Figure 7.10: Tearing a cloth apart by applying forces in opposing directions.

One can even tear paper like objects apart by applying forces in opposing directions on two corners (Figure 7.10).

Another possible application would allow soft volumetric bodies to be squished so as to fit into cavities or compressed so as to slide underneath other objects. Soft materials could also be used for terrains; deformation could be triggered by users digging their fingers into the terrain, using their whole hands to form valleys, or using a cupping gesture to create elevations. More open-ended and free-form interactions with particle systems (e.g., simulating sand) and fluids can be imagined in a gaming context.

7.5 User Study

To further understand and evaluate the utility of the techniques described in this paper, an exploratory experiment was performed. The following questions were addressed:

- Are users able to comprehend and exploit the openness of interaction that the physics model affords?
- Is the interaction of sufficient fidelity?
- Is it discoverable and predictable?
- Do users notice and value the added fidelity, or do they just expect kinematic control?
- Ultimately, how do users express their expectations in the physics-enabled manipulation?

The study exposed 6 participants to 3 simple physics-enabled tasks (as shown in Figure 7.11), and analyzed various behavioral and experiential aspects of interaction during task completion. The 3 male and 3 female participants came from a range of backgrounds, and all had normal (or corrected to normal) vision and no mobility impairments. The experiment was conducted on an early prototype of Microsoft Surface [Mic08], using the Nvidia PhysX gaming physics engine [Nvi08].

The experiment utilized a 3x3x2 within-subjects (repeated measures) design. Each participant worked through the three puzzles in each of three interaction techniques: Joints, Proxies, and Particles. Pilot testing included a fourth condition, Direct Forces, but this was dropped during further testing (as explained below). For techniques that do not intrinsically support collisions, a simple collision model based on kinematic objects was applied, allowing interactions from the top and the sides of an object. In addition, we hypothesized that the presence of visual feedback showing users precisely where their input is applied might improve the discoverability of each technique. Visual feedback of Joints was represented as red lines drawn from the contact point to the anchor point on the object (these disappeared if the joint was broken); Proxies as red cubes at each center point where contact was sensed; and Particles as smaller red cubes per pixel in the contour image. We ran each technique with and without visual feedback as our third independent variable.

The task setup (see Figure 7.11) was as follows.

- In Task 1, each of four spheres and rectangles were placed exactly on matching targets; each object disappeared upon proper placement.
- In Task 2, an assortment of objects of different shapes, sizes, and masses were sorted onto the left or right portions of the screen depending on their color.
- In Task 3, a cylindrical object was steered from a set starting position (far top right of photo) to a target (shown in red) by passing several waypoints (shown in blue) without dropping the object from a platform (which caused the task to restart).

(a) (b) (c)

Figure 7.11: (a) Task 1: Exact positioning of boxes and spheres (b) Task 2: Sorting by color. (c) Task 3: Steering a sphere (red) across narrow bridges (blue).

The tasks were presented in the same order to each participant, whereas the order of interaction techniques was counterbalanced across participants using a Latin-square design. Experimentation occurred in two main phases (with visual feedback of the input and without), presentation of which was, again, counterbalanced across participants. During the experiment, participants were not given any direct instruction, but had several attempts to try out each new puzzle. Participants performed each task twice (excluding any training), under experimental conditions, to provide an average completion time for each condition. Participants were interviewed informally after completing their session.

7.5.1 Early Issues with Direct Forces

Initially, the Direct Forces technique was implemented by applying a smooth velocity at a given contact point on the object, computed as a measure of the displacement between the contact's current and last positions (i.e., kinetic friction). This seemed a fair approximation for modeling surface input as direct forces. However, our pilots questioned the efficacy of this technique. Specifically, users found it difficult to complete tasks that involved accurately positioning objects; i.e., moving and then stopping an object at the target location. Moving the object could be performed reasonably, but to stop it the user needed to counteract the motion in the opposite direction. This often led to excessive velocity applied in the reverse direction, causing objects to "overshoot" the target. Consequently, performance with this technique was so poor that we felt it needed no further evaluation. Based on these issues and feedback from the pilots, we excluded this technique from analysis.

7.5.2 Initial Results and Observations

Although this was only an initial exploration, we observed many promising interactions and forms of gesture within the study. Users seemed aware of the potential of this new type of environment and exploited the physics-based system's facilitation of experimentation, and we observed many new interaction strategies.

Kinematic Control and the Curse of the Single Finger Figure 7.12 shows the completion times for all tasks. Joints provide kinematic control that closely mimics drag-and-drop behavior, and thus facilitate easy positioning of objects. This is reflected in the results. After some experimentation, there was a moment when users discovered that the object was under familiar kinematic control. Users commented that "my hands are like magnets" or "I can press hard and stick my fingers." Of course, pressure and magnetism were not factors at play here (in fact, post study interviews revealed that participants were unsure of the general principle behind the Joints technique). Nevertheless, users performed the task rapidly after discovering the object was somehow fixed to their fingers.

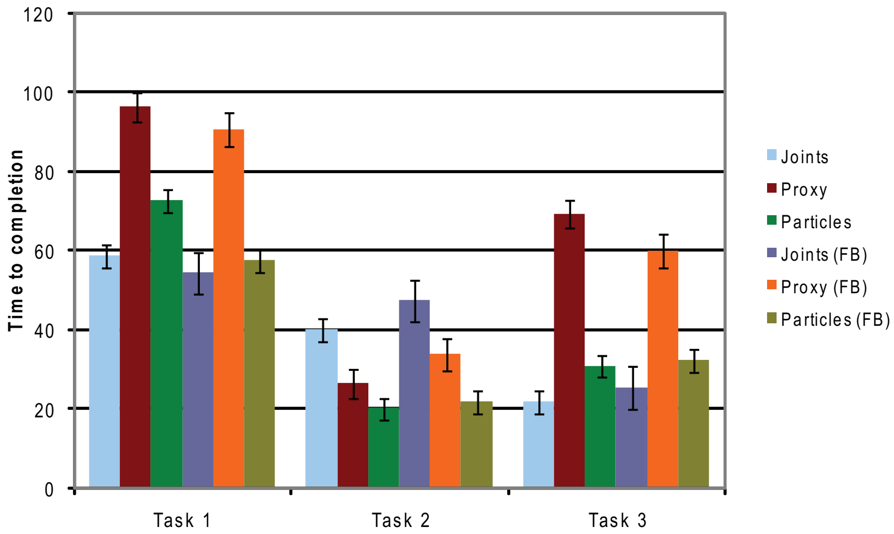


Figure 7.12: Task completion times. FB denotes conditions in which feedback of user's input was provided.

However, the quantitative results tell only part of the story. During the study we also observed many limitations with the kinematic approach. The discovery of this type of essentially drag-and-drop behavior in the Joints condition led users to predominately interact with a single finger and with a single object at a time. Even rotations of an object were predominantly undertaken using a single finger [KCST05].

Experimentation with multi fingered or bimanual techniques was therefore rare in the Joints condition. During informal interviews, users commented that the condition was "limited" and "less satisfying" than the other techniques even though they performed the tasks rapidly. Although it is too preliminary to draw significant conclusions, it does suggest the need to measure more than task completion time when evaluating such physics-based techniques.

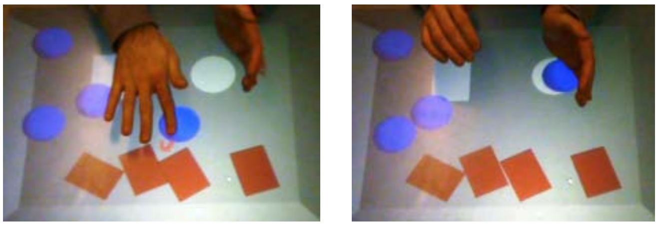
Users also had a poor understanding of how collisions were supported in a kinematic approach such as Joints. We observed many instances where accidental collisions caused by hit testing on the side as opposed to the top of the object would cause an object to move away from the user and cause a great deal of confusion. This makes us revisit whether a kinematic plus collision model makes any sense to the user at all: Why indeed should an object only be sticky when you touch its top as opposed to its sides? This actually led some users to infer that objects were magnetized in a way that supported both attraction (when touching the top) and repulsion (when colliding with the sides).

Using Feedback to Go Beyond Kinematic Control As shown in the results, feedback did not play a significant role in the Joints condition, as one might expect given the familiarity of the approach. Feedback played a more significant role for Particles in Task 1. After some training time, users discovered they could interact with more than just their fingertips. Bimanual "cupping" and "throwing and catching" techniques were devised to rapidly move objects to target positions (Figure 7.13). These strategies, and the general level of fine control, enabled users in the Particle condition to obtain completion times comparable to more kinematic approaches. During interviews, users reflected positively about the interactions Particles afforded.

(a) (b) (c)

Figure 7.13: Using contours of the hand to move (a) multiple boxes. (b) Providing a barrier to smoothly change direction of a sphere over the target area. (c) Fine-grained manipulations to complete sorting task.

However, these types of contour based bimanual interactions could not be utilized with Proxy objects – although participants did try. In fact, in many cases, a hand gesture on the surface would be poorly approximated as a single proxy (the center of mass of the contact shape), causing objects to slip through a hand or causing other peculiar hit testing behavior. Multiple fingers were used to reorient boxes effectively, but overall, bimanual control was rare.

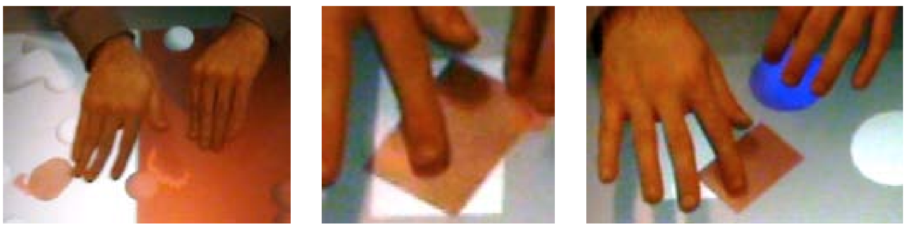


(a) (b)

Figure 7.14: Throwing and catching an object from a greater distance using both hands.

While the "drag and drop" nature of Task 1 clearly favored kinematic control such as that offered by the Joints approach, Task 2 offered a clear advantage to concurrent manipulation of multiple objects for rapid sorting. As might therefore be expected, use of both Proxy and Particles techniques, which seemingly promoted multi-touch interaction, led to faster completion times in this task (Figure 7.15).

Coming to Grips with Non-planar Objects Another specific trade-off in our design was that the rigid body cubes in the Proxy condition only provided an effective means for interacting with flat objects. They provided little grip of spherical objects (or more complex 3D meshes). This was clearly evident in the final task where the Proxy cubes struggled to keep the cylindrical object under control, as shown in Figure 7.11. In this task, we found users often reverted to point-based interaction to control the small non-planar object; the use of contours was infrequent. However, our initial results suggest that Particles still outperform Proxy objects for these purely point-based interactions. This suggests that for scenarios where touch-only input is available, the Particle model subsumes the single Proxy object model.



(a) (b) (c)

Figure 7.15: Two-handed and multi-fingered strategies adopted in the proxy and particle conditions. (a) Coarsely moving objects using both hands. (b) Two-fingered rotation by applying torque to align a box. (c) Fine-grained movement of two objects using a single finger of each hand.

7.6 Observations and Implications

The results of the user study and general experimentation suggest that while the more familiar kinematic approaches (somewhat inevitably) offer more predictable control in some situations, the *particle proxy* approach can offer comparable performance while providing new modes of interaction (such as cupping the ball in Figure 7.13). That our study participants were able to devise new manipulation strategies from limited feedback and training is encouraging. With more time, we expect users to further draw on their experience with real-world manipulations.

There are a number of ways in which our interactive surface simulation does not match the physics of the real world. In suggesting that we abandon familiar, kinematic point-based control in favor of strongly physics-based techniques, an important consideration is whether users are able to negotiate these differences.

First, while in the real world one might apply more or less force to control friction, our system has no sense of how hard the user is pressing. When using particle proxies, the amount of friction applied is instead proportional to the number of proxies applied to the object, which itself is related to the surface area of the touch. For example, a single finger moving across an object will apply less friction than multiple fingers. Not surprisingly, this distinction was generally lost on study participants, who often tried to press harder to bring an object under their control. Similarly, our users would sometimes apply multiple fingers to an object when they wanted precise movement. Because of the inevitable imprecision of the simulation, the object would move too unpredictably for fine control. In many of these cases, it would have been better to use a light (small) touch rather than a full grip.

Second, grasping a virtual object by placing contacts on either side can be difficult if not impossible in many of our techniques. Such manipulations require persistent resting contacts to be placed on virtual objects. The particle-based approach, in which each proxy is created every frame, places the proxy corresponding to a grasping finger on top of the object, thus defeating attempts to grasp it. The single proxy object approach uses persistent proxies, and so allows grasping of an object resting on the floor. It may be possible to extend the particle approach to allow proxies to persist at a given depth when it seems appropriate, or to explore a hybrid approach in which both the particle and single proxy techniques are used simultaneously.

Grasping may be difficult to replicate for more fundamental reasons. Virtual objects exert no counteracting force against the fingers, so it is difficult to know how “hard” the fingers are pressing on an object. Grasping an object in order to lift it out of the plane may be challenging to depict directly on an interactive surface with a 2D display. Similarly, the sensation of moving the hand across an interactive display surface is the same regardless of the simulated material, and whether the contact exerts static or kinetic friction.

Clearly one need not completely replicate the physics of object manipulation in order to construct useful applications exhibiting physics-inspired behavior. The appropriate degree to which the techniques in this paper are applied depends on the application. A game might naturally exploit detailed physics throughout, while a graphical layout application might be selective. Joint constraints provided by physics engines may be used to constrain motion, for example, to ease alignment tasks. While joints can be used to simulate the real-world counterparts of traditional GUI sliders, dials, buttons, and the like (as suggested by [GW06]), some aspects of traditional interactions do not naturally lend themselves to a physics implementation. Changing the size of an object dynamically, for example, does not lend itself to rigid-body simulation.

7.7 Summary

We have introduced a number of techniques to incorporate interactive surface input primitives into a real-time physics simulation. Our techniques take advantage of the fidelity of sensing provided by vision-based interactive surfaces, with the goal of enabling in a virtual domain the range of object manipulation strategies available to us in the real world. Thereby forming a new model for tabletop interaction that incorporates the physical aspects often accredited to direct-touch interaction and successful in other interaction styles with a flexibility and open-ended nature not possible with previous approaches. Our model allows for and encourages rich interactions with virtual objects, using not only multiple fingertips but novel, often in-situ designed interactions such as the cupping, throwing and raking gestures discussed in Section 7.5.2.

Our user-study and general observations have revealed that users are receptive for this new model of interaction. In general they had little problems in interacting with the virtual realm. However, kinematic more traditional interaction styles were still prevalent especially during initial contact with the system. We expect that users, as they become more familiar with the capabilities of the approach, will further draw on their real world knowledge and thus develop richer interaction strategies. However, this requires further study possibly in a long-term field study.

Of course the approach has several limitations and issues such as the lack of tactile feedback and collision response (cf. Section 7.5.2). Novel materials might be able to mitigate this problem through passive tactile feedback [HH09b] or novel, active (actuated) displays providing dynamic, computer controllable tactile feedback in addition to vision based multi-touch input [HH09a].

Another issue and possibly a more important issue is the mismatch of input and output fidelity. Even though the input data from vision-based interactive surfaces is rich it is, in most cases, limited to 2D – sometimes augmented with a narrow band of hover [SMA03] or pressure sensitivity [SGHB07, HKI08]. In contrast, in our interaction model the virtual world we interact with is three dimensional. Our mapping (especially the particle approach) allows for coarse 3D interactions (cf. Section 7.4) but these interactions are severely limited because scene motion is constrained to two dimensions. Virtual objects can be manipulated in various ways (see Figure 7.2) but these manipulations are limited to interacting with objects by touching them on their sides or touching them from top to drag them. On real tables we routinely pick objects up flip them around and put them into containers, all of which are interactions not possible with our current model. Therefore, interacting with the 3D scene sometimes can feel as if interacting with objects protected by a layer of transparent film while poking objects with fingers or chop-sticks.

The next two chapters of this dissertation we begin to address this issue by exploring several hardware and software solutions, in order to provide users with means to control objects in 3D. Chapter 8 discusses several novel display technologies that enable sensing beyond the usual surface constraint tabletop input. Chapter 9 introduces our approach to enabling interaction with 3D virtual worlds rendered on an interactive tabletop. The proposed solutions aim at providing interaction techniques that allow users to lift objects off the surface and control their position in 3D using as-direct-as possible interactions without instrumenting the user in any way.

Chapter 8

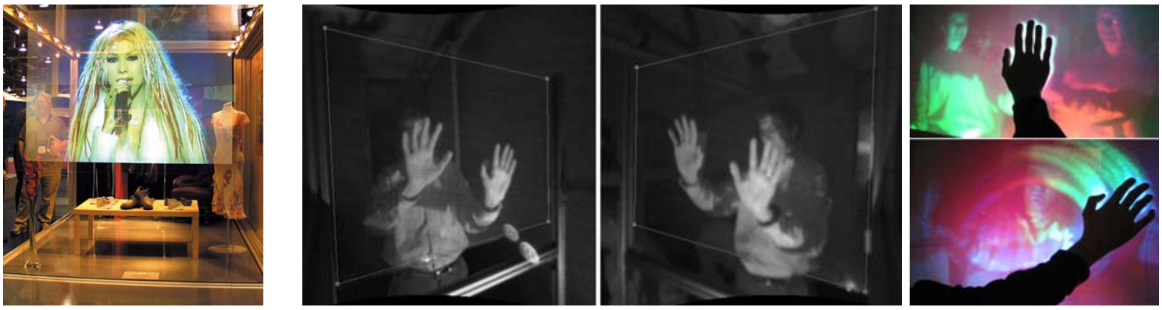
Sensing at a Distance

In Chapter 7 we introduced our new model for tabletop interaction. A model that fuses a sophisticated gaming physics simulation with rich sensor data from interactive surfaces. The model allows for fine-grained and flexible interactions with the virtual through a variety of real-world strategies because it does not constrain the behavior of virtual objects through pre-programmed or scripted behavior. Instead we have presented a set of techniques that model interactive surface input data so that users can interact with virtual objects through quantities such as force, mass, velocity and friction in an open-ended manner.

Early observations of system usage yielded promising results. Users were in general open to the new interaction model and, after getting used to its possibilities, started to explore several novel interaction strategies, including bi-manual, shape-based and multi-fingered interactions. However, a major limitation of the model became apparent during our experiments. Because the sensed input is bound to the two-dimensional (2D) display surface, interacting with three-dimensional (3D) content can be cumbersome or even impossible. If we consider how important the third dimension is for interactions on real tables (e.g., stacking, turning of pages) this problem puts our goal of providing an interaction model that is both *physicality* and *flexible* somewhat into perspective.

Recent emerging display technologies enable to project an image on the display surface and to simultaneously image through the surface to sense users' hands and other objects at a distance. These technologies could be used to extend the interaction space of interactive tabletops from purely surface bound 2D interaction to more holistic 2D on-surface and 3D above the surface interactions. A space that is naturally included in interactions with regular tables (e.g., leafing through a book, piling objects for storage and loose organization) and therefore an interesting direction for further research. We will discuss some of the recent projects and developments using these emerging technologies in the context of surface computing.

One category of these novel display technologies are holographic materials such as the DNP HoloScreen [DNP]. The Holoscreen is a rear projection screen coated with a holographic film that only displays images that are rear-projected from a certain angle (30°-35°). Light with an incident angle outside of that range passes right through the display. This material allows to project imagery onto what appears to be a regular sheet of acrylic (Figure 8.1 a). Furthermore, cameras can be placed behind the projection screen in order to capture the scene in front of the display. Another interesting quality of the material is its near transparency in the infrared spectrum (independent of incident angle) which makes the material a promising candidate for IR based multi-touch sensing.



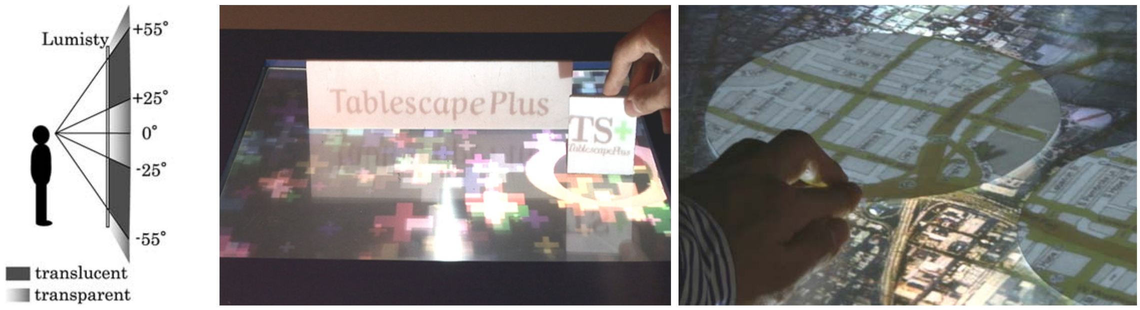
(a) (b) (c)

Figure 8.1: (a) Holoscreen displaying a video in a shop window. (b) Image after lens correction as seen by the two cameras. Applying perspective correction to rectify both views allows for obtaining a fused image that only shows objects close to or on the image plane. (c) Projection of both camera images using different color channels on top. Estimated motion rendered in a time-lapse drawing application.

In the **TouchLight** [Wil04] system Wilson demonstrates the utility of the projection screen as large vertical interactive surface. Here the DNP HoloScreen is used to display virtual imagery. Two off-axis cameras mounted behind the viewing plane are used to detect hands and objects on the projection screen. To decide whether an object is on the surface or not proximity to the viewing plane is computed by relating stereo disparity - the change in image position of a particular object when viewed from different positions (Figure 8.1 b). Since the projection screen is transparent, not only objects in close vicinity to the plane can be detected but also objects further away. An interactive drawing applicatiopn with decaying strokes and cycling colors is presented as sample application (Figure 8.1 c). Furthermore, the setup allows for high-resolution imaging on or through the screen. For example, documents present on the surface can be detected and “scanned” in order to create digital copies immediately available for further manipulation. Another possible application may be live video conferencing with direct eye-contact normally not possible with a usual separate screen-camera setup.

Our second prototype for in-the-air interactions discussed in Section 9.6 explores a DNP HoloScreen mounted horizontally in a tabletop setup in combination with an advanced depth sensing camera. This setup allows for accurate per-pixel depth estimation and therefore rich, high DOF sensing of the users’ hands on and above the table.

A related material is directional view-controlling film (Lumisty film [Sum04]). This material diffuses light with a certain incident angle while being transparent to light with a different incident angle (see Figure 8.2 a). In coordination with an adequate projector layout this allows for precise control over orientation and visibility of information. The **Lumisight Table** [MIO+04] allows for sensing of multiple hands and physical objects via vision-based sensing. In addition the system includes a unique optical engineering approach as it allows to display a) one group image visible to all users surrounding the table but correctly oriented for each user, and b) displaying of private information only visible to a specific user in addition to the shared view. The setup uses multiple projectors and two layers of Lumisty film. Input is sensed using two cameras; one operating in the visible light spectrum and one in the IR domain. Lumisty film is transparent for light coming from orthogonal incident angles therefore the cameras can detect objects on top of the table's surface. Hands (or palms) are tracked in the visible light spectrum using a color differencing technique. Since only one camera is used the presence of hands can be detected but no information is available whether the hand is on or above the surface. This information is used for a so-called shadowing interaction technique. Users can position their hand over buttons to interact with them but don't need to touch them. The infrared camera detects other physical objects equipped with retro-reflective, passive markers. An IR illuminant is used to reveal these markers to the camera.



(a) (b) (c)

Figure 8.2: Lumisty view control film and its application in tabletop computing.(a) Viewing angle and visibility of light passing through Lumisty film. (b) Displaying different images on the table's surface and objects above it. (c) Using tracked objects to reveal otherwise hidden information on secondary movable display surfaces in UlteriorScape.

Many tabletop systems use tangible input as the primary or a complementary input style and many different ways to sense these objects have been presented. Often the tangible objects remain passive and serve as input devices only. The **Tablescape Plus** system [KNM07] offers a different take on tangibles on tabletops. Based on the Lumisight table [MIO+04] the system is capable of displaying different images on the table's surface and on the surface of smaller and mobile surfaces positioned vertically on the tabletop. These smaller projection screens are identified and tracked by a camera mounted underneath the main surface using IR illumination and retro-reflective AR-toolkit markers [KB99]. The position and orientation information is used to correctly scale and align projected information in order to ensure legibility of text and graphics (see Figure 8.2 b). Furthermore, this information is used to turn the projection platforms into input devices. Different information can be displayed depending on position and orientation. For example, the tabletop can be turned into an interactive doll play where miniature figures are displayed on the mobile screens performing different actions depending on the platforms location. In another example the mobile screens are used to display cut views of 3D objects while a bird's-eye view of the same object is displayed on the table's surface.

UlteriorScape [KN08] is a direct descendant of both the Lumisight [MIO+04] table and the Tablescape Plus [KNM07] system. While Tablescape Plus is a single user system UlteriorScape adds multi user capabilities to the mix. The system is capable of showing different views to individual users located around the table as well as using movable objects on the tabletop as additional projection screens (Figure 8.2 c). The setup uses an additional projector, mounted perpendicular to the main projection screen, and a fresnel lens to enable projecting through the main screen (allowing to project onto objects further away from the surface than possible in Tablescape Plus). Several applications demonstrate the systems functionality. Including a map application (Figure 8.2 c) that reveals additional layers upon placement of any diffuse object.

The previously discussed materials, Lumisty film and the Holoscreen, are both passive. Their ability to control which light rays are scattered (and therefore become visible) and which pass through the screen depends on the incident angle of projected light. In some situations this may be a problem because the screen is not equally well viewable from all sides of the setup - or different information might be visible from varying viewpoints. However, there are materials available that allow to actively control whether they are transparent or opaque at the flip of a switch. Better known as privacy glass, polymer dispersed liquid crystal (PD-LC) is a material containing liquid crystal molecules which are normally randomly oriented and therefore the glass appears frosted. Applying a voltage across two parallel, transparent substrates on either side of the screen creates an electric field which causes the molecules to align along the electric field's potential. In this state the material is almost entirely transparent and appears as regular, transparent glass.

(a) (b) (c)

Figure 8.3: (a) Effect of the electronically switchable diffuser. (b) Any diffuse Material can be used as passive projection screen revealing the second image projected through the surface. Active objects (emitting an IR pattern) can be tracked through the surface and projected onto.

Based on a similar material (PSCT-LC [LC-07]) **SecondLight** [IHT+08] is a novel tabletop prototype that also allows to image and project through the display surface. Rather than using a static diffuser at the table's surface an electronically controlled projection screen is used (Figure 8.3 a). The screen can be switched between clear and diffuse states at high rates. This allows to rear-project a seemingly stable image during the diffuse states and project through the surface in the clear states. The second image may be projected onto objects on or above the surface

(Figure 8.3 b). In combination with an adequate timing scheme (synchronizing the screen and projectors) this can be achieved without any light-bleed or other interference between the two images, as is the case with other solutions based on Lumisty film. Furthermore, a thin sheet of IR edge-lit acrylic is mounted on-top of the switchable diffuser allowing for multi-touch sensing based on the FTIR principle. Finally, a second camera mounted underneath the surface may image through the surface to identify and track objects or even users within the space above the surface while the screen is in its clear state (Figure 8.3 c). These unique capabilities enable a variety of standard tabletop "on-surface" interaction techniques including multi-touch and tangible interaction but also enable new interaction techniques that make use of the space above the surface. SecondLight serves as hardware platform in our explorations of above-the-surface 3D interactions discussed in Chapter 9.

Chapter 9

Interactions in the Air - Adding More Depth to Interactive Tabletops

The goal of the work described in this chapter is to overcome the constraints opposed by the sensing limitations of traditional digital tabletops, ultimately we want to extend our physics-based interaction model into the third dimension. We present a new technique that combines the previously discussed "on-surface" interactions with the ability to, conceptually, pick virtual objects up from the tabletop and manipulate their 3D position and their orientation before placing or dropping them back onto the surface. Our goal has been to define a technique that feels as natural and as direct as possible, giving users the sense they are actually lifting the objects off the surface.

Recent emerging display technologies allow for projecting a stable image onto the display surface and at the same time for imaging the users hand at a distance (cf. Chapter 8). We explore how these technologies can be used to create 3D interaction techniques that unlock the space above digital tables for interaction. Our approach is threefold:

- We discuss two tabletop prototypes both using novel projection screen materials to allow sensing at significant depths beyond the display surface.
- Existing and new computer vision algorithms are employed to sense and interpret user interaction *above* the display surface.
- An evolution of shadow-based feedback mechanisms is introduced, in order to improve the "directness" of the enabled in-the-air interactions.

The novel combination of these technologies opens up the ability for the user to interact within the 3D space above the surface. Although this work is still a work-in-progress we close this chapter with a discussion of our initial experiences and observations with the system and feedback elicited from informal user observations.

Contribution Statement: A paper based on this Chapter has been accepted for publication at UIST '09 [HIW+09]. Most of this work has been done at the two Microsoft Research labs in Cambridge, UK and Redmond, USA. Many of the discussed models have been designed and implemented while I was a research intern in the Cambridge Lab. I am the lead author on that paper. Together with Shahram Izadi I initiated this project and designed and implemented many aspects of the first prototype including the depth estimation, shadow generation and height-based rendering techniques. Shahram Izadi, Steve Hodges and others created the original SecondLight platform [IHT+08]. I made several modifications to that hardware platform. Andy Wilson designed and implemented the second prototype including novel computer vision algorithms for the depth-camera enabled sensing. He also originally invented the algorithm to detect the pinching gesture [Wil06]. We used the code base from our previous work on physics-based interaction which was primarily implemented by Armando Garcia-Mendoza. Andreas Butz contributed many ideas throughout the planning and implementation phases (e.g., height-based rendering styles). All authors contributed to the writing of the submitted paper.

9.1 Motivation

Interactive surfaces [DL01, Han05, Mic08, Wil05, Wil04] and multi-touch tables in particular have received much attention in recent years. They allow us to directly manipulate digital information using the dexterity of multiple fingertips and even whole hands. As a result, these interfaces are often deemed more natural than their desktop counterparts. However, for all their compelling qualities, interaction with such surfaces is inherently constrained to the planar, 2D surface of the display. For many tabletop interactions this constraint may not appear to be a problem, particularly when direct manipulation with 2D content is desired. Recent research is however beginning to motivate the need for rendering 3D content on tabletops [HCC07, HC07]. One recent research direction is our own work exploring the notion of 3D physics-based tabletop user interfaces (UIs) [WIH+08]. Here the traditional UI elements are replaced with 3D objects, with properties such as depth, mass and friction, which are modeled by a physics simulation. Arguably, these systems are even more natural than traditional interfaces. However, because the interaction – the sensed input and the corresponding displayed output – is still bound to the 2D surface, there is a fundamental limitation in manipulating objects using the third dimension.

This makes some of the simplest real-world actions such as stacking objects or placing them in containers difficult if not impossible using these interfaces. For example, in Figure 9.1 we show a ball and cup rendered in such a physics-based tabletop UI. The simplest and most natural way to get the ball into the cup would be to pick it up, but this simply is not possible when interaction is bound to the surface. This is just one illustrative example, but it highlights that when considering full 3D interaction, tabletops are far from natural. To draw a comparison with the real-world, the current interaction fidelity offered by such systems is analogous to manipulating physical objects only by pushing them. Instinctively we would want to pick them up, tilt and angle them and so on.

(a) (b) (c)

Figure 9.1: An example demonstrating the limitations of current tabletops for 3D interaction. (a) Here we wish to pick the ball up and place it in the cup. (b + c) However, such a natural interaction is difficult when interactions are bound to the surface.

It is not just these types of physics-based interfaces that could benefit from such 3D interactions. In fact many 2D tabletops have a sense of 3D – for example notions of Z-ordering, stacking and layering are commonplace.

In this chapter we present new techniques that combine on-surface interactions with the ability to conceptually pick virtual objects up from the tabletop, manipulate their three dimensional position and potentially their orientation, and place or drop them back onto the surface. Our goal has been to define a technique that feels as *natural* and as *direct* as possible, giving users the sense they are actually lifting the objects off the surface. Hereby we mean interactions with the virtual that feel as close as possible to interacting with real objects (*natural*) and that are triggered by the same or similar interactions as in the real-world (*direct*). For example, a pick-up interaction should happen via grasping and then lifting of the hand/forearm to control the objects 3D position.

We chart the evolution of this work by describing two rear projection-vision prototypes we have built, based on a switchable diffuser [IHT+08] and a holographic projection screen [Wil04]. In both cases it is possible to rear-project an image onto the tabletop surface whilst simultaneously using a rear-mounted camera to detect the user's fingers and hands as they interact in the space above the surface. We have also experimented with two different types of camera system: a regular camera used in conjunction with a system of diffuse infrared (IR) illumination which allows us to both estimate the depth of hands in the image and to robustly detect a simple pinch gesture; and a true depth-sensing camera which generates more noisy data in our setup but nonetheless supports even higher degrees-of-freedom (DOF) interactions.

The novel combination of these technologies opens up the ability for the user to interact within the 3D space above the surface. However, a key challenge is the loss of *directness* when a user moves from interacting *on* the surface to the space *above* it. To combat this we present the evolution of a new shadow-based feedback technique. Preliminary feedback from a system that uses a naïve algorithm to generate shadows highlights the utility of the approach and points to some limitations. As a result of this, we have developed a much more sophisticated technique for rendering real-time computer generated shadows on the surface based on the true 3D sensing above the surface.

9.2 A Discussion of 3D Tabletop Interaction

3D carries many different connotations; some may immediately think of stereo displays which give users a perception of real depth [CPS+97], some may think more about sensing 3D gestures and others about a 3D rendering of content on a flat display. In the interactive tabletops and surfaces literature 3D has very specific meanings, which we elaborate upon in this section.

A great deal of research on 3D interaction has been conducted over the decades, from various fields such as Virtual Reality (VR), Augmented Reality (AR), and interactive and tangible surfaces (for an overview see [BKLP04]). It is difficult to touch upon all of these systems and concepts in this chapter. However, Grossman and Wigdor [GW07] provide an excellent overview and taxonomy of interactive 3D in the context of tabletop applications.

Perhaps one of the most important aspects in thinking about 3D on the tabletop is the separation of the *multi-touch* and *gestural input*, the *display technologies* used for output and the *graphics* itself.

Multi-touch and Gestural Input Input can be thought of as the user's physical actions in a defined space, which can be sensed by the system. For a standard tabletop this might be the display surface itself, where the user's fingertips can be sensed in 2D.

In defining the input capabilities of a system, it is often useful to consider the degrees-of-freedom (DOF) that can be sensed. For standard multi-touch screens, each fingertip offers 2DOF in terms of its position, plus a third (which could be interpreted as yaw) if orientation of the finger can be calculated. Certain surface technologies [MIO+04, Rek02] can sense the hover and pressure input, which can provide further, albeit limited, DOFs. We refer to these types of input as constrained 3D (following [GW07]) because they only support Z-based input in limited ways.

One way of extending the input space above the table is to instrument the user, for example using gloves augmented with sensors [BVBC04, CJK+92] or using tracked styluses [ABM+97, CPS+97]. However, this adds an artificial feeling to the interaction, moving away from the natural tabletop interaction typically afforded. A less intrusive approach is to place stereo [ML04] or depth cameras [Wil07] above the display surface. There are issues with such techniques however, including the space requirements, objects occluding the camera, and practicality of real-world deployment. Systems such as [IHT+08, KN08, KNM07, Wil04] have demonstrated the use of special projection screens, such as switchable diffusers or holographic materials, which open up the possibility to image the user at greater distances away from the surface using rear mounted cameras. These can be used for extended depth sensing, but to date these systems have not supported finger, hand and arm tracking for 3D user input.

Again it is important to recognize the differences regarding fidelity of 3D input. Most approaches sense depth as an estimation of distance of an object (such as a user's hand) in relation to the screen [ML04]. This gives 4DOF interaction when combined with regular on-surface interactions, allowing for Z-based input. To determine pitch and roll to support true 6DOF input more elaborate computer vision or sensing techniques are required.

Display Technologies For most tabletops the display used for rendering digital content to the user is a 2D planar device such as a LCD or projection screen. In past tabletop research, stereoscopic displays with shutter glasses [ABM+97, CPS+97], or AR and VR head-mounted displays [NMKT05] have been used to generate 3D output. These techniques require the user to be instrumented

There are emerging display technologies that allow for uninstrumented 3D output. One category is auto-stereoscopic displays [PPK00, SMG+05], which can project stereo image pairs into the users left and right eyes directly, without the need to wear shutter glasses. These displays tend to be single-user and heavily viewpoint dependent, making their use for tabletops less appealing. Volumetric displays [Fav05] do not have this limitation – because they render "voxels" (volumetric pixels) in a 3D physical volume they can be used simultaneously by different users with different viewpoints. However, whilst they support some forms of 3D interaction [GWB05, GB08] it is not possible for users to place fingers or hands inside the rendered volume or support direct manipulation.

Other display possibilities include projection of 2D imagery onto the surfaces of physical objects that are placed on the surface or held above it [IHT+08, KN08, KNM07], a term referred to as constrained 3D [GW07] or tabletop spatially augmented reality [RWC99]. Both front- [IRP+04, PRI02, UI99b, Wil07] and rear-projection tabletops [IHT+08, KN08, KNM07] have been demonstrated with these possibilities.

The Graphics The graphics rendered on the display are typically 2D, which is perhaps not surprising given typical sensing and display technologies. However, many 2D GUIs have some notions of constrained 3D through the Z-ordering they use to layer 2D widgets.

For 3D graphics, one important factor for the user is the *perceived display space*. In [GW07] this is defined as *"the possible spatial locations for which displayed imagery can exist based on stereoscopic depth cues"*. However, even for a standard 2D display rendering 3D content (what is typically termed as 3D onto 2D) this notion of perceived display space is an important one. For example, depending on the virtual camera position, graphical projection and other depth imagery, it is possible to create the perception of a 3D volume inside the tabletop.

3D Tabletop Interaction Techniques In this section we give an overview of the existing work exploring 3D on tabletops, and attempt to categorize them based on the definitions introduced previously. We first introduce two further concepts that allow us to reason more deeply about these systems:

Input and output coupling: This defines the extent to which the input and output are spatially coupled. For regular multi-touch tabletops [DL01, Han05, Mic08, Rek02, Wil05] there is a tight coupling between input and output spaces. That is information is touched directly and output is rendered at the same location and in direct response to user input.

Input to graphics mapping: This defines how naturally the sensed input maps onto manipulations with the UI (which in these examples is 3D or constrained 3D). This is an important consideration, particularly when fidelity of output and input differ. In a scenario where high DOF input data is available a literal mapping would be when a virtual rotation about Y would be performed by the user through a rotation of the wrist about the forearm (“roll”).

Perhaps the highest fidelity of 3D tabletop interaction comes in the form of stereoscopic systems, such as [ABM+97, CPS+97] which combine 3D sensing via sensor augmented data gloves and styluses, 3D displays and a 3D UI. Here there is a straightforward mapping and coupling between the elements. However this comes at a cost in that the user must be instrumented. As [GW07] mentions *“such devices can be uncomfortable, reduce the ubiquity of the system (as they will no longer be walk-up-and-use), and can cause the user to lose the context of their surrounding environment or collaborators.”* Crucially these systems as well as AR and VR-based tabletops move away from the notion of interacting naturally with the tabletop. Based on these issues we specifically desire to explore un-instrumented 3D interactions with tabletops.

Hancock et al. [HCC07] demonstrate a set of one-, two- and three-fingered touch techniques to manipulate 3D objects in an uninstrumented manner. They use a regular multi-touch tabletop with 2D input sensing and display, but 3D graphics. A major contribution of the work is the mapping of 2D input to 3D manipulations. Given the differences in fidelity of input and output, interaction techniques are defined to map from 2D translations on the surface to 5 and 6DOF manipulations. Although the results of a study showed that these interactions can be readily learned, they cannot be considered natural, in that they do not directly resemble the ways we manipulate objects in the real-world.

Davidson and Han [DH08] present a pressure-based technique for manipulating the Z-order of objects on a large interactive surface. A regular 2D display is used, but the sensing and UI can be considered as constrained 3D. The pressure data provides an additional DOF to give the user a more natural mapping for pushing objects below one another.

Subramanian et al. [SAL06] define a multi-layer interaction technique using a 3D tracked stylus for input above a tabletop with 2D output and a constrained 3D UI. Here the user can maintain multiple layers of visual content and move between layers by moving their pen in the space above the tabletop. This system uses a single stylus to interact, leading to symbolic gestures for switching between layers. We are interested in more natural touch and whole hand gestures for interacting on and above the tabletop surface.

Tangible user interfaces have also explored extending tabletop interaction space into the physical 3D environment [FIB95, IU97]. Some use physical objects as props to interact with digital UIs [MIO+04, UI97, UI99b], others top-project virtual imagery onto 3D objects and surfaces [IRP+04, PRI02]. Other work has explored rear-projected setups to display information on tracked, moveable objects above the surface [IHT+08, KN08, KNM07]. Although, these offer powerful real-world metaphors, our aim is to give users a more direct sense of interacting with the virtual in 3D, without using specialize tracked objects as interaction proxies.

	Input			Output			Data			Instrumentation		Coupled		Mapping	
	2D	2D+	3D	2D	2D+	3D	2D	2D+	3D	User	Table	Y	N	L	S
Stereoscopic Tables			×			×			×	×		×		×	
Hancock et al.	×			×					×		×	×			×
Davidson et al.		×		×				×			×	×		×	
Subramanian et al.			×		×			×		×			×		×
Tangible tabletops			×		×				×		×	×		×	
Physics	×			×					×		×	×			×
Wilson (Depth Cam)			×		×				×		×	×		×	
DepthShadows			×	×					×		×		×	×	

Figure 9.2: Comparison of approaches to 3D on digital tabletops. Last row shows uncharted space explored in this chapter.

9.3 Natural Interactions Beyond the Surface

This motivation for exploring natural gestures for 3D interactions comes from our prior work [WIH+08] (discussed in Chapter 7) which explored the use of physics engines to bring real-world dynamics to interactions with standard 2D digital tabletops. We achieved this through a novel mapping between surface input and the 3D UI. The sensed 2D input was projected into the 3D scene as a series of rigid bodies that interacted with other 3D objects. This allowed a literal mapping between input and manipulations for the 3DOFs sensed on the surface (x and y translation and yaw). Although this provided a direct way to interact with 3D objects by pushing on the sides or tops of them, the approach reached its limitations whenever objects needed to be manipulated with higher DOFs. The 3D physics engine and renderer support these higher DOF manipulations, but these were not naturally available to us through the surface input sensing.

The goal of the work reported here is to enable more natural 3D interactions with tabletops. We are interested in enabling interactions with 4DOFs and higher without the need to instrument the user in any way. Given the limitations of existing 3D display technologies in the context of tabletop computing, we wish to support traditional 2D displays for output. We are particularly interested in exploring integrated rear projection-vision form factors because of their ability to mitigate some of the issues of top-down tabletop systems, including occlusion, bulky form-factors and complexity of deployment.

This configuration carries some interesting characteristics that, as shown in table 9.2, have yet to be explored in the context of 3D interaction. In particular our input is 3D but our output limited to the 2D display. This provides an interesting challenge in compensating for the lack of direct interaction when the user moves to interacting off the surface. The closest work to ours [Wil07] is based on a depth-sensing camera and overhead projection, providing 3D input and constrained 3D output, which means that output can be provided to the user even off the surface.

9.3.1 In the Air Interactions

Our first system is a rear projection-vision system that uses a switchable diffuser to extend the input space for interaction beyond the tabletop. A 3D scene is rendered as a birds-eye view on the switchable projection screen. Users can interact with objects in the 3D using standard 3DOF interactions defined in [WIH+08]. Additionally users can gesture directly above a virtual object in the 3D scene, which allows the object to be picked up. Subsequent changes in the position of the user's hand in 4DOF will result in the virtual object being repositioned in 3D space. A release gesture is recognized to drop the virtual object back down were upon it can be manipulated using rich on-surface touch interactions.

To implement this technique we use a modified version of the SecondLight system [IHT+08]. This uses frustrated total internal reflection (FTIR) for sensing multi-touch [Han:2005:FTIR]. Our setup introduces five main changes to the standard SecondLight setup:

1. In place of the layer of edge-lit clear acrylic in front of the display surface, we use a similar material known as EndLighten [Pro09] which provides a certain amount of diffuse illumination across its surface in addition to supporting FTIR multi-touch sensing.

2. The EndLighten is simultaneously edge-lit with two wavelengths of infrared light, namely 850nm and 950nm.

3. A system of diffuse 950nm illumination is introduced by mounting strips of IR LEDs behind the Endlighten display surface.

4. A second infrared sensitive camera is also mounted directly behind the display surface. This camera is fitted with a 950nm pass filter, whilst the first is fitted with an 850nm pass filter.

5. Only one projector is used, so that the SecondLight unit is only capable of on-surface projection.

(a) (b)

Figure 9.3: (a) Detection of thumb and forefinger pinch gesture. A computer vision algorithm observes hole enclosed by fingers. (b) Pixel intensity based height estimation. Objects close to the surface appear very bright. Objects further away are increasingly dimmer.

The first camera in this modified system images IR light reflected from touching fingers, whereas the second images IR reflected from the diffuse illumination of environment by the EndLighten and LEDs under the display. This second camera is used for gesture recognition and sensing the depth of the user's hand.

To detect when users want to pick-up objects (and later release them) we use a robust and real-time computer vision algorithm which detects when the user brings their thumb and index finger together in a "pinch" gesture, as reported in [Wil06] and shown in (Figure 9.3 a). The algorithm uses a simple connected components analysis to identify the hole that is formed when the thumb and index finger are touching. The algorithm reports the 2D center of mass of the hole plus the major and minor axis to determine the orientation. Upon detection we perform a raycast operation from the 2D position of the hole and determine which virtual object the ray intersects with first (if an object intersects at all). We then pick up the object by defining a virtual joint within the physics engine from the object to a proxy created directly above it. This joint ensures that we have kinematic control over the object, allowing us to position it in the 3D scene, without it being affected by gravity and other friction effects. The joint is destroyed once the tracked hole disappears from the image, which results in the object returning to a dynamic state and falling back down towards the ground.

Whenever a pinch gesture is detected, the average intensity of a region of pixels around the hole is calculated. This gives a simple measure of depth of the user's hand. Hands in close proximity to the screen (and hence the IR light sources) will have brighter pixels in the camera image, and this begins to fall off as the hand moves away from the IR light sources as shown in Figure 9.3 (b).

These different technologies come together to allow the user to pinch over a virtual object, control the height of object by lifting their hand up or down (whilst maintaining the pinch gesture), reposition the object, and release the object back down into the 3D scene, as highlighted in Figure 9.4.

9.3.2 Depth Shadows

Interacting with virtual objects rendered on the surface in this way opens up 4DOF interaction capabilities on and above the surface. However, there is also a key challenge when facilitating

Figure 9.4: Using the combined depth sensing and pinch to place an object into a virtual container. Height of virtual objects is directly controlled by distance of user's hand to the screen.

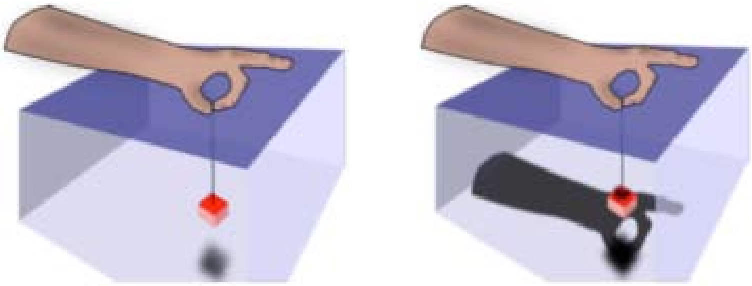


Figure 9.5: Depth Shadows are conceptually shadows cast from the user's hand into the 3D scene to allow closer coupling between input and output spaces.

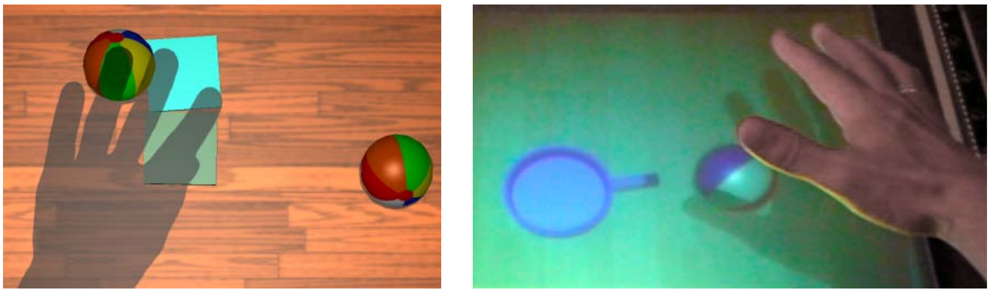
this type of interaction – the user's hands and rendered content are only in direct visual contact when interacting directly on the surface. A key challenge arises in the loss of *directness* when a user moves from interacting on the surface to above it.

Returning to our earlier discussion around perceived display space, it becomes clear that we are conceptually creating a 3D volume inside the tabletop. Clearly the user's hands sit outside of this virtual space, separated by the actual physical bounds of the display. Using raycasting and the virtual joint metaphor means that users have even a greater sense that their hands are decoupled from the 3D volume rendered on the tabletop. To try and compensate for this decoupling, we describe a shadow-based technique that helps connect the user's hand in the real-world with the objects in the 3D scene. We do this by conceptually casting a shadow of the user's hand into the 3D scene, fusing this with the shadows cast by the virtual objects in the scene. This provides a real-world metaphor to map between actions in the physical space and interactions inside the virtual 3D scene, as shown in Figure 9.5. These shadows can also function as additional depth cues for the user when adjusting an object's position along the Z-axis.

We use a GPU based shadow mapping approach to create a realistically shaded scene. In a first render pass the scene is drawn from the light's position. Only the z-test is performed, resulting in a depth-map of the scene. In a second render pass the scene is drawn to screen this time from the cameras point-of-view. For each pixel the 3D position is converted into light-space coordinates and its distance to the light is compared to the corresponding value in the depth map. For most pixels this distance will be equal unless an object blocks the line of sight from the current pixel to the light. In these cases the pixel is in the shadow of another object. Additionally we smooth the edges of shadowed areas using the percentage-closer filtering (PCF) algorithm [RSC87].

Generating a shadow of the user's hands could potentially require additional sensing and illumination. However, we are already imaging the hands of the user from the second tabletop camera used for gesture recognition and depth estimation, and there are several viable options to create realistic renderings of hand shadows in the 3D scene from this camera image.

A "naïve" solution would be to render the raw, binarized camera image onto the ground plane of the scene or as overlay on top of the entire scene – a technique often used in tabletop remote collaboration setups (see [Kir07] for an overview). Our aim however is to heighten the user's perception that they are interacting directly in the 3D scene. With this in mind, a more



(a) (b)

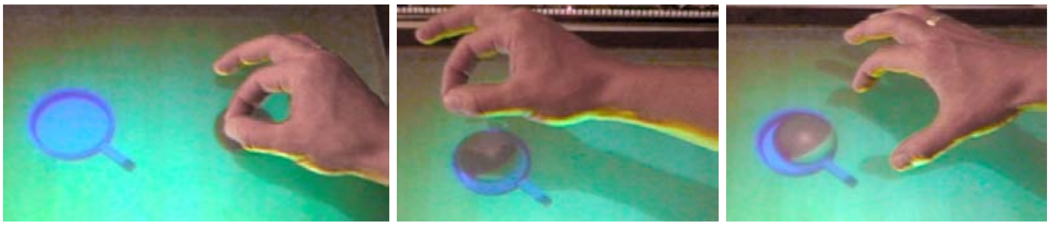
Figure 9.6: (a) Virtual shadow cast by the users hand. Hands geometrical correctly shadow virtual objects underneath them. (b) Virtual shadows are aligned with real hand.

elegant solution involves computing the 3D geometry of the user's hand based on the height values calculated from pixel brightness in the raw image and introducing this 3D mesh into the scene. A shadow mapping technique could then be used to generate shadows for both the user's hand and other virtual objects in the scene. In practice however this mesh is difficult to generate using diffuse illumination alone. The main issue here is that pixel intensity fall-off is relatively accurate when hands move mostly along the Z-axis but the brightness of the captured image is not uniform throughout the space above the table. This problem is mitigated in our depth estimation by averaging pixel intensity over a region around the detected holes. In order to create a detailed 3D mesh this solution would not be viable because accurate depth information would be necessary on a per-pixel level.

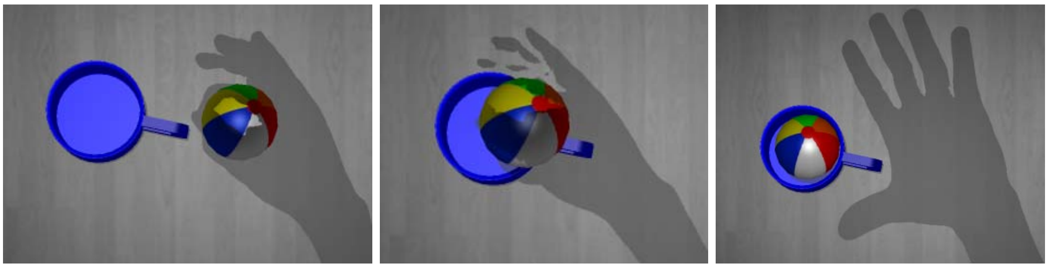
An alternative approach is to first generate a shadow map for all virtual objects in the scene and then fuse this with the raw image from the camera. To do this we leverage the pixel intensity to compute an estimated Z-value for each pixel. Transforming this position into light-space coordinates produces a depth-map of the users hand as seen by the light. This depth-map can then be merged with the shadow-map by comparing the Z-value for each pixel. The larger Z-value is stored in the final shadow-map. Figures 9.6, 9.7 illustrate the hand-shadows generated by this approach.

Why Shadows for 3D Tabletop Interaction? Researchers have explored the use of shadows to support a number of interactive systems. In [HZR$^+$92] the notion of shadows were used as depth cues and as a means to open up additional DOFs for 3D manipulations on the desktop. Shoemaker et al. [STB07] present the idea of real shadows as a mechanism for reaching across large displays. Shadows have perhaps been most extensively used for remote collaboration, where renderings of hands and arms of remote participants act as additional feedback mechanism for remote awareness (for an excellent literature overview see [Kir07]).

The use of shadows for 3D tabletop interaction has yet to be fully explored however, and it presents many compelling aspects. Perhaps most importantly, it gives users a natural feedback mechanism for representing their hands in the virtual scene (see Figure 9.7).



(a)



(b)

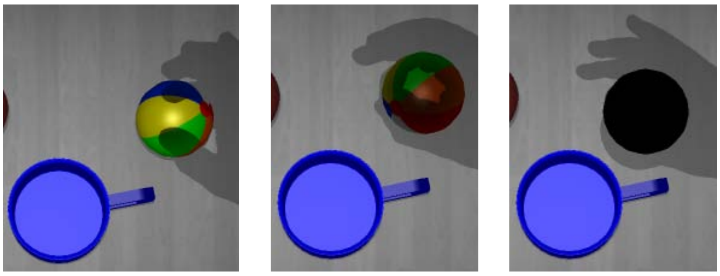
Figure 9.7: Combining depth sensing, pinch and virtual hand shadows to place an object into a virtual container. (a) Sequence as seen by the user. (b) Screenshots of the same interaction.

We are often unaware of our shadows when interacting in the real-world, and so they offer a subtle, non-intrusive form of feedback. However, the feedback can also be rich. For instance, how the hand shadows end up being cast in the scene and in relation to other virtual objects and shadows gives users additional depth cues allowing them to get a better sense of the 3D nature of the scene they are interacting in. For example, a user knows their hand is over a virtual object if the shadow is cast onto the top of it (Figure 9.6 a) whereas if the object occludes the shadow the hand is clearly underneath.

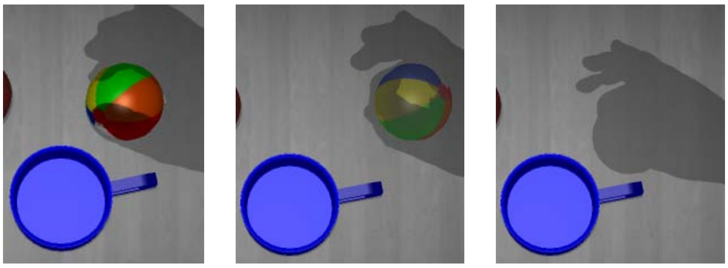
9.3.3 Conveying Depth

To provide additional depth feedback and potentially strengthen the coupling between the input and output spaces we have also explored several techniques to add additional depth cues to the 3D scene.

The first technique compares the height value of each pixel on the surface of any object in the scene with a configurable threshold. Once the threshold is surpassed the object is gradually de-saturated. Objects that are high in the scene are rendered entirely black as shown in Figure 9.8, a. The remaining two techniques modify the opaqueness of objects rather than their color. Once the entire object is higher than the threshold only the shadow of the object is rendered. In a variation of this technique fades pixel opacity in and out gradually, illustrated in Figure 9.8, b.



(a)



(b)

Figure 9.8: (a) Object color is de-saturated based on height. (b) Objects turn into their own shadow as they are lifted off the surface.

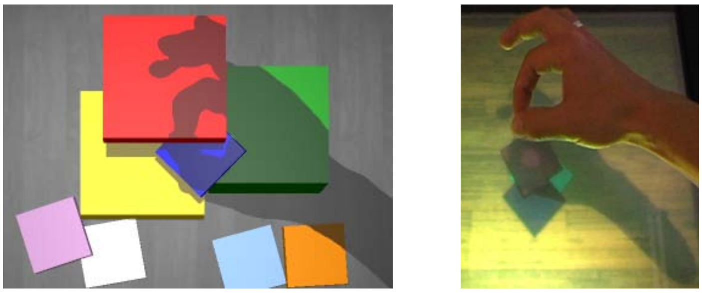
9.4 Interaction and Application Scenarios

We have really only just begun to explore the interactions that are enabled by our shadow and in-air techniques. The focus of our work to date, and this chapter, is the core underlying concepts and technical implementations. However, in this section we touch briefly on some of the interactive possibilities.

Layering and Stacking Fundamentally our technique allows users to pick objects up from the surface and directly control their position in 3D. In traditional GUIs, fine control of object layering involves dedicated, often abstract UI elements such as a layer palette (e.g. Adobe Photoshop) or context menu elements (e.g. Microsoft Powerpoint) Our technique allows for a more literal layering control. Objects representing documents or photographs can be stacked on top of each other in piles and selectively removed as required.

Our technique may also be applied in application domains that directly involve or benefit from 3D data such as gaming, medical visualizations and CAD applications. In the architectural domain our technique may be used to construct complex 3D models by picking-up various building blocks and then placing them on top of each other, akin to using Lego™.

We are of course excited by the potential that our work brings to physics-based tabletop interactions. Figure 9.9 shows some simple examples of stacking and finer grained layering control in this context. We can make use of the additional degrees of freedom to mimic popular storage strategies applied in the real-world: using containers such as shoeboxes, bowls shelves



(a) (b)

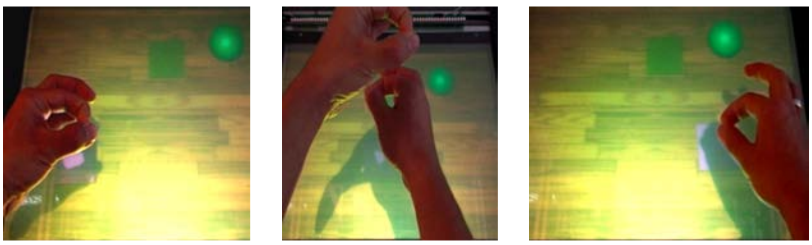
Figure 9.9: (a) Moving objects over and underneath other objects. (b) Creating piles of related objects for organization.

for storage of digital content. It is also possible to in interact with non-rigid objects in a much richer way, for example stretching pouring fluids out of containers.

It is also important to note that our technique can also handle multiple hands interacting at the same time. The algorithms can identify and track several hands simultaneously as long as the pinch holes are not occluded. This allows users to pass objects, such as a virtual document, to one another using the free space above the surface (Figure 9.10).

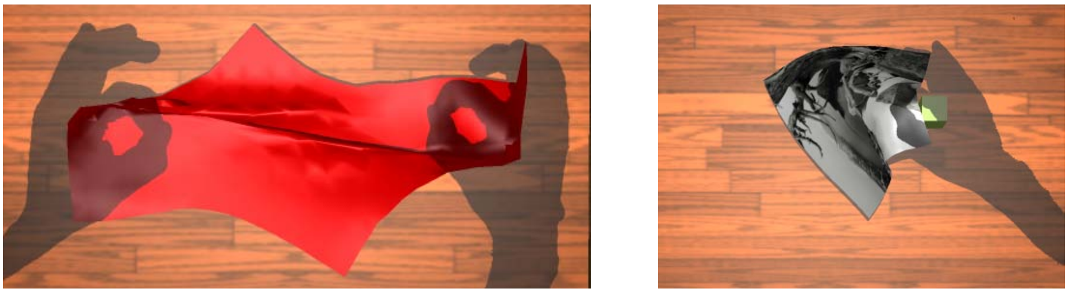
Interaction with Soft Bodies In addition to rigid-body dynamics, many third-party physics engines are capable of simulating non-rigid dynamics such as cloth and fluids. Up to now we have only discussed examples involving rigid bodies but our technique also works with soft-bodies such as cloth, rubber sheets or paper-like documents.

A user may grasp such objects with one or two hands to move objects around and to manipulate them in rich ways. For example, two hands may be used to stretch or tear a cloth or crumple it up into a soft ball. Our technique also allows users to control the height of the attachment point (the part of the soft body they are holding onto), enabling users to drape cloth over rigid-bodies to quickly experiment with different clothing in a 3D game editor for example.



(a) (b) (c)

Figure 9.10: Bi-manual in-the-air interactions (a) Object is held by the left hand; both hands approach each other. (b) Left hand releases pinch gesture right hand grabs the object. (c) Object is held by right hand.



(a) (b)

Figure 9.11: (a) Bi-manual stretching of cloth. (b) Draping a textured cloth over a solid cube.

9.5 Initial Reflections

We have demonstrated our prototype to literally hundreds of colleagues on various occasions, using several scenarios where depth-based interactions are mandatory or greatly eased the task at hand. During these occasions we had the opportunity to enable and disable the shadow rendering and depth-based feedback mechanism described in this chapter. While this use of our system can in no way be considered a formal user evaluation, we have nonetheless had the opportunity to observe hundreds of users interacting with it, often with little or no instructions at all. Here we report some of the observations we found noteworthy.

In our demonstrations users have commented that the shadows gave them a greater sense of interacting with the virtual objects. We have yet to evaluate this quantitatively. From observing users it does become apparent that the shadow technique greatly improves the usability of the system. While users could pick-up objects following detailed instructions and with practice when the shadow rendering was turned off, the technique proved to be difficult and cumbersome. It was difficult for users to discover how to operate the system with shadows disabled. However, enabling the shadow rendering made a big difference. In this condition users were able to learn the technique by either observing other users perform the pick-up (and release) gestures or sometimes by simply exploring different gestures to lift objects off the surface. The shadows provide an additional cue for depth and a way of rationalizing about what the system is sensing. This seemed useful in particular when using the pinch gesture, where if the user saw a broken hand shadow on the surface they assumed the gesture would not work.

It is also interesting to note that our shadows are inverted in that they get smaller the further away the hand is from the screen. Users we have demoed to seem less aware of this in aspect, and have commented that it might feel unusual to have the hand shadow get larger as it moves away from the surface. In some senses, the further the hand gets from the device the less the feedback should be portrayed on the screen. Of course, this is just a hypothesis, one we hope to evaluate in the future.

Another observation was that users did not necessarily think the pinching gesture were the most intuitive choice. Grabbing gestures, for example using all fingers of one hand to grip the object from its sides, were observed more frequently, which are not sensed by our system. Some users tried to perform a pinching gesture but in the wrong orientation such that the system could not observe an apparent “hole”.

Some of the users had problems in judging how high objects were away from the surface. Enabling the object to fade as it moved off the ground plane certainly improved the users’ depth cues. However, some of our demonstrations, for example, steering objects through a 3D maze, required fine-grained depth adjustments. Here some users had difficulties controlling the object’s height when only the shadow was rendered (and the object had faded away).

Finally, users often asked for additional degrees of freedom in the 3D manipulation. In particular carrying out 3D orientation such as tilting objects or reorienting more complex shapes (such as the cup) when these had become knocked over – this is something that is difficult to achieve just with 4DOF.

9.6 Exploring a New 3D Tabletop Configuration

To address some of these issues we have recently begun to explore another tabletop configuration, which augments some of the in-the-air interactions in our previous prototype. One of the main rationales for this work was to more accurately emulate grasping, rather than the iconic pinch gesture, and also to think about how to enable the other available DOFs. Early experience with this system shows the promise of some of these new features as well as fresh challenges.

Hardware Configuration For display, we use a DNP HoloScreen, a holographic diffuser mounted on an acrylic carrier, in combination with a NEC WT610 short throw projector. As in [Wil04] the HoloScreen material was chosen because it is nearly transparent to IR light, while the projector was chosen to meet the projection angle requirement of the HoloScreen material. Our HoloScreen measures 40” diagonal (compared to 20” for SecondLight).

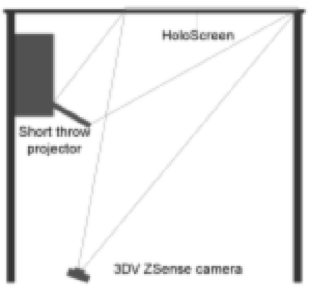


Figure 9.12: Overview of our second tabletop hardware configuration.

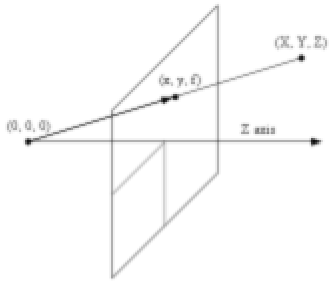
We also use a 3DV ZSense depth camera [Wil07] to image objects above the table. The ZSense is placed behind the HoloScreen, in a vertical configuration. For the holographic nature of the HoloScreen not to interfere with the operation of the ZSense, the camera must be placed off axis to prevent any IR illumination reflecting directly back from the underside of the acrylic. Like SecondLight, the combination of camera, display material and projector results in a completely self contained waist-high table, illustrated in Figure 9.12.

From Range-sensing to World Coordinates The 3DV ZSense camera uses pulsed infrared laser light and a very fast solid-state shutter to construct a per-pixel depth map of the scene (320x240, 30Hz). One of the main features of the camera is the ability to compute the world coordinates of any point within its configurable near and far clipping planes D_{near} and D_{far}. An 8-bit value d at depth map location (x, y) may be converted to depth in real units (cm):

$$D = D_{near} + \frac{255-d}{255} * (D_{far} - D_{near}).$$

Consider the vector V originating at the center of the camera and passing through (x,y,f), with focal length f, x and y in cm (the pixel width is known). World coordinate (X,Y,Z) is then D units along V: $(X,Y,Z) = D \cdot \frac{V}{\|V\|}$.

(a)



(b)

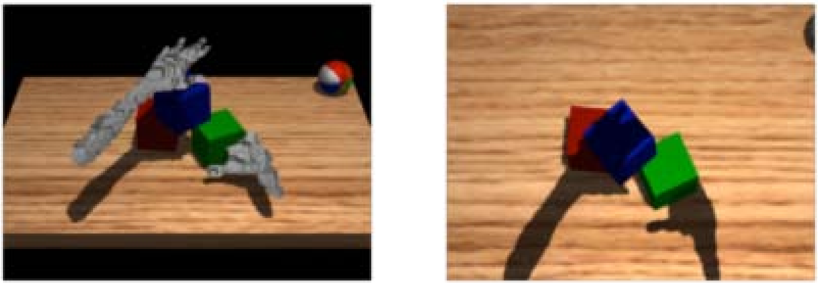
Figure 9.13: (a) Raw ZSense depth image. (b) Conversion to world coordinates.

More correct hand shadows Our SecondLight-based prototype creates hand shadow effects by attenuating the light falling on the scene on a per-pixel basis according to the observed image of hands above the table. This approximation of shadows has limits: for example, a hand will shadow objects that are known to be above it. As we explore more realistic grasping models, such limitations may be troublesome.

Our second prototype improves the simulation of shadows by constructing a mesh from world coordinate values computed as above. This mesh is rendered when computing the shadowmap, but is not rendered with the shadowed scene. An example is shown in Figure 9.14.

Grasping Model The pinch detection technique has important advantages described earlier, but as a gross simplification of human grasping behavior it can be a poor model, particularly when the user is unaware of its restrictions.

With our second prototype we are exploring a more accurate model of grasping behavior that, rather than raycasting the center of holes formed by pinching, determines when the user touches an object in multiple places. Touching an object is determined by hit testing the geometry of each object with the world coordinates of the user's fingertips.



(a) (b)

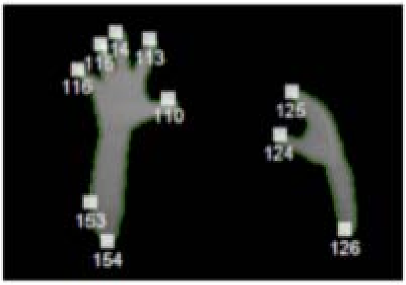
Figure 9.14: 3D mesh and shadows. (a) Illustration of computed world coordinate mesh used in shadowing algorithm. (b) Table top view shows left hand fully above the blocks, right hand penetrating green block.

While it is tempting to perform all calculations (e.g., finding fingertips) in world coordinates, it is important to note that depth estimates are noisier than the (x, y) location of an object that appears against a far background (such as a hand above the table). This is in part due to the ZSense's approach of computing independent depth estimates for each pixel location. For this reason, it is often better to precisely locate the depth discontinuity due to the edges of such an object using traditional image processing techniques on the 8-bit depth map, followed by area averaging of depth values and finally conversion to world coordinates.

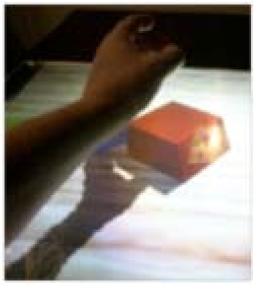
Accordingly, we detect fingertips by analyzing the depth map only. While there are many ways to perform such shape detection (e.g., [ML04]) we proceed by finding the contour of every connected component in the binarized version of the depth map [CCL04]. Each external contour is then walked twice: first to compute a Hough transform histogram to select circular shapes of typical finger radius, and second to locate the points on the contour corresponding to the maxima of the histogram. Multiple such maxima are eliminated via a standard nonmaximal suppression technique, where maxima are considered overlapping if they lie within some arclength distance along the contour (see Figure 9.15). The depth value of each remaining fingertip location is computed by sampling a neighborhood in the depth map. This is then converted to world coordinates, tracked from frame to frame and smoothed by a Kalman filter.

A user's attempt to grasp an object is detected by first determining which fingertips (if any) are contained within the 3D shape of each dynamic body in the scene. If a body not previously under grasping control is found to contain exactly two fingertips, it enters grasping control. Thenceforth, the body remains under grasping control if the same fingertips are contained with the body, regardless of the number of fingers in the body. The body is dropped when either of the original fingertips leaves the body, as when, for example, the user opens their grasp (see Figure 9.15, b).

This grasping model does not consider where each fingertip touches or penetrates the body as it would if it were a true simulation of grasping behavior. However, it improves upon the pinch detection and raycasting approach by respecting the geometry of the grasped body while using a similar gesture, and by performing 3D picking. With this model, it is possible to grasp an object that is sitting under another object.



(a)



(b)

Figure 9.15: (a) Contour detection (green) and finger tracking. (b) Grasping with fingertips.

5 DOF Interactions Once under grasping control, the body may be manipulated in 3D by analyzing the combined motion of the two grasping fingertips. Translation in three dimensions, yaw about Z and roll about the wrist are easily computed from the motion of two points. Pitch cannot be computed in this way, but rather via a least-squares fit to a plane of a number of pixels in the neighborhood of the grasp.

While the contour-based detection of fingertips allows easy determination of whether two fingertips are on the same hand, bimanual manipulations may be performed when the two fingertips are on different hands.

More fidelity requires more control The more detailed modeling of shadows, grasping and manipulations suggests a higher fidelity interaction than possible with our first prototype. Indeed, a number of interactions are possible that were not before: precisely orienting an object and grasping an object at a given height are two examples.

However, the same improvements in fidelity demand that the user be more aware of the 3D position of their grasp and the objects they are attempting to manipulate. Initial early experience with this tabletop system suggests that the rendered shadows are extremely important, perhaps more so than in the earlier prototype. The more accurate modeling of shadows may be helpful in certain situations.

Errors in finger tracking can make objects harder to grasp or cause objects to fall from grasp. In particular, when the grasped object is small or the grasp is too tight, the fingertip contours will merge and disappear. To combat this effect we have experimented with increasing the effective size of the object for hit testing. Another option is to fall back to the pinch gesture in this case (it is easily identified as an internal contour). Perhaps rather than rely on fragile finger tracking, an approach based on contour or mesh tracking is feasible. Ultimately we would like to more closely simulate the physics of grasping, after the style of [WIH+08].

Grasping in 3D also depends on the user's ability to see more than the tops of objects. This in turn depends on the choice of graphics projection transformation. A standard perspective transformation allows the sides of an object to appear if it is not near the center of the table. Moving the camera to one side addresses this limitation, but makes it impossible for the simulated table and the physical table surface to coincide. We suggest an "off-center" perspective projection (also known as "perspective control lens" in photography) to restore this correspondence, so that objects on the table plane will appear at the correct location on the physical table, while objects with height exhibit perspective effects.

9.7 Observations and Implications

Our second system was developed to address some of the shortcomings of the first, which were uncovered by observing literally hundreds of users interacting with it. However, it turns out that both systems have their own strengths and weaknesses and we therefore thought it valuable to present both setups in some detail in this chapter. Perhaps the most obvious difference between the two systems is the input fidelity afforded by each. The SecondLight setup can only approximate the distance of objects above the surface, and it only provides 4DOF input which was one of the main limitations according to user feedback. Switching to our second prototype, and in particular the ZSense camera, provides higher DOFs and enables exciting new interaction techniques that we have only just begun to explore.

However, the added sensing flexibility of the system comes at a cost – foremost speed and robustness. The ZSense camera provides calibrated depth data but only at 30Hz and a lower resolution. The image provided by the two tabletops also differs significantly in terms of noise. The ZSense depth image requires extensive smoothing and processing further reducing the tracking frame rate. So there is a clear trade-off between system responsiveness and input fidelity.

These differences in sensing fidelity also impact the interaction style. In SecondLight, raycasting into the scene upon detecting a pinch gesture always picks the topmost object. The more accurate depth data in our new tabletop allows for more precise 3D grasping such as pulling objects from the middle of a pile. While this is sufficient for most basic 3D tasks the interaction still remains somewhat indirect as if the user was manipulating the scene through a glass window. Although somewhat limited in terms of 3D interaction fidelity we feel that this interaction style could easily be mapped to other scenarios, especially applications that use extended 2D data (2D plus z-layering). The more accurate depth data in our second prototype allows for precise 3D grasping which makes the interaction feel more direct and the coupling of input and output is more literal. Therefore we feel that this interaction style is the more appropriate model for true 3D interactions.

However, the SecondLight platform has some compelling qualities absent from our new tabletop. Firstly, the on-surface image is much higher quality in terms of viewing angle, than it is with the holoscreen. Also, the possibility of projecting a second image through the display, which is much more practical with a switched diffuser, is interesting. Whilst we haven't explored this in our current work, projecting onto the user's hands to provide coarse information about objects under manipulation is an interesting avenue of exploration.

Finally, on surface interactions are readily available in the SecondLight setup via FTIR multitouch sensing this enables interesting combinations of fine-grained interactions 2D data and a seamless transition to 3D manipulations. Currently the depth camera setup does not sense on surface interactions at all. Wilson has demonstrated how the HoloScreen can be turned into a multi-touch screen in earlier work [Wil04]. We plan to add on-surface sensing capabilities to our current setup to further explore the combination of on- and above the surface interactions.

9.8 Summary

We have implemented and demonstrated two prototype systems, motivated by a desire to use the space above an interactive tabletop to enable richer depth-based interactions, without compromising an integrated hardware form factor. Thereby we have extended our interaction model proposed in Chapter 7. The work discussed here extends the sensed interaction space from pure "on-surface" interaction to "above-the-surface" interaction. We have demonstrated a couple of enabled interactions previously cumbersome or impossible. In addition to the two novel tabletop prototypes we have designed a mapping of high-DOF input to 3D data (displayed on a traditional 2D screen). Of course we have only begun to explore this space and further study is necessary to judge the utility of our approach. To this end, we feel that our approach holds the potential to effectively extend the interaction space of interactive surfaces from a 2D plane to a 2D-3D continuum. Thus further blurring the line between the display and the environment it exists in.

We feel that this work builds on the existing literature through a number of distinct contributions:

- We present a number of extensions to SecondLight to support sensing up to half a meter beyond the tabletop.
- We have developed a novel shadow-based technique to provide feedback during mid-air interactions.
- We have built a second tabletop system based on a depth camera and holoscreen.
- We have implemented a tabletop system with high DOF interactions without requiring any user instrumentation.

Currently our work builds on our physics-based UI to emphasize the naturalness of the interaction afforded. However, we feel that the techniques described here can be generalized to other 3D systems and even to 2D tabletop UIs with notions of Z-ordering and layering.

Part IV

Conclusion and Future Work

This last part concludes this dissertation. In Part I we introduced the reader to the general field of tabletop computing and specified the problem statement for this dissertation. This introductory discussion was then followed by a discussion of related work. Part II discusses our own explorations into common tabletop interaction paradigms. In this part we also analyzed the suitability of these approaches as generic interaction model for tabletop computing. Part III builds on the insights from the previous parts and introduces a new model for tabletop interaction in Chapter 7. This model is then extended to allow for richer, more direct manipulations of 3D data using novel display technologies and computer vision algorithms Chapter 9.

In this final Part IV we summarize in Section 10.1 all the aspects discussed in detail earlier on. Section 10.2 draws conclusions across all previous parts of the dissertation. in order to structure this discussion we revisit the two main questions postulated in the introduction: *what is the interface?* and *where is the interface?* Finally we discuss directions for future work in the areas of tabletop hardware, interaction models and 3D interaction on tabletops.

Chapter 10

General Discussion and Conclusion

This chapter summarizes the areas of our research described in this thesis. We discuss advancements in the field of interactive tabletop computing presented by this dissertation – to do so we compare problems and issues described in Chapter 1 with the proposed solutions outlined in the following Chapters. We also discuss how well our studies and analyses of proposed prototypes helped answering the two main questions *what is the interface?* and *where is the interface?* We then highlight areas for future work not covered in this thesis and new challenges and areas for research that have been opened up by the work described herein.

10.1 Summary

The work described in this thesis has two main components of interest to the field: First, an analysis of existing interaction techniques, based on literature review (Chapter 2) and studies of our own prototypes (Chapters 3, 4, 5). Second, a new model for tabletop interaction (Chapter 7) and an extension of this model, allowing for as-direct-as-possible interactions with 3D data from above the display surface.

In the introduction (Chapter 1) we have outlined many of the compelling qualities that digital tabletops feature. The horizontal display and table-shaped form make this class of computing devices a compelling one because amongst other things, they naturally enable users to sit around the device and interact with the digital while also engaging in face-to-face conversations with other group members. The ability to interact with the digital through direct touch, and in some cases physical interaction handles, adds a direct almost natural quality to the mix. We have shown that these devices bring with them not only many appealing qualities but novel challenges and uncertainties.

We have also highlighted that the field of interactive tabletop computing is still in its infancy and therefore many developments are still in flux. If we look at the way we interact with the more traditional desktop PC a well defined (and more refined) model for interaction exists, with its main ingredients being Windows, Icons, Menus and Pointers (WIMP). This model became dominant because the hardware platform became standardized and input typically happens via mice and keyboards. For tabletops this clearly defined model has yet to emerge. One reason is that a standardization process on the hardware front has yet to happen. Many different approaches to user input sensing have been proposed, demonstrated and studied (cf. Section 6.1). One important aspect, is that tabletop hardware has become increasingly capable of sensing rich user input.

Our analysis of related work in Chapter 2 has identified two major interaction styles that have begun to emerge for tabletop computing. First, the use of pen- or finger-trace gestures to invoke commands and thus control the behavior of on-screen objects and the state of entire applications (Chapter 2.2.2). Second, the combination of physical objects with direct-touch input (Chapter 2.2.3).

In Part II we have discussed and analyzed our own explorations into these interactions styles. Chapter 3 details the motivations, design and study of BrainStorm. A system designed to support collaborative problem solving using several interactive surfaces embedded into walls and a table. The chapter explores the suitability of a gesture-based interaction style as general model for tabletop computing.

Our analysis based on the criteria of *physicality* and *flexibility* (Section 3.5) has shown that the real world resemblance of virtual objects in the system helped users to understand and learn how to operate the system. However, interpreting Newtonian physics in an overly simple fashion proved to be problematic because users developed expectations that they could apply strategies learned in the real world to manipulate the virtual. However, when designing **Brainstorm** we only implemented a set number of gestures that could be recognized by the system. Many manipulations possible in the real world were not translated into gestures. Albeit an extended gesture recognition algorithm would somewhat mitigate this problem and reduce the observed user frustration.

In some situations we believe there exists a more fundamental problem. When considering the rich and flexible ways we manipulate real world objects using various strategies as we see fit, it becomes apparent that this flexibility can not be matched by pre-programmed, scripted behavior. In the real world subtle changes in manipulation can have different outcomes (e.g., imagine the broad repertoire of tricks good ball players possess). Often these subtle changes are lost to a gesture recognition algorithm due to the need to reliably differentiate and classify various gestures. Finally, recognized gestures need to be mapped to commands within the application. This mapping often happens a-priori and remains fixed. This again stands in contrast to the dynamic and flexible way humans manipulate objects in the real-world. Interaction breakdowns can occur whenever users try to interact with the system in ways not anticipated.

In Chapter 4 we explore how tangible objects on interactive surfaces might help to address the lack of flexibility observed with gesture-based approaches. Photohelix 4.2 is an application designed to support the co-located browsing and sharing of pictures on a digital tabletop. A physical handle is used to position and browse the entire picture collection while direct-touch interaction is used to interact with individual pictures. We had various assumptions about the benefits of using a physical handle for interaction (cf. Section 4.1). Our observations from a number of lab-based user-studies yielded interesting results but could not confirm all of them

(Section 4.5). We could observe that the physical affordances and 3D nature of the input device allowed for richer and more flexible interactions than observed in the direct-touch condition. These benefits however, are limited by problems with sensing and interpreting the manipulations (not unlike those in the gesture-based interface). Here the problem is that the virtual objects do not behave in a realistic manner and are not subject to the same laws as the physical interaction handle – therefore the manipulation of the virtual is again limited by what was anticipated by the designer in advance. Finally, we could not find particular strong evidence speaking in support of other assumed benefits such as *eyes-free* manipulations and natural support for bi-manual interaction.

Another problem that spanned across the two explorations became apparent during many hours of system usage and observations during user studies. Both prototypes, **Brainstorm** 3 and **Photohelix** 4, were built and deployed using the same hardware – DViT enabled Smartboards [SMA03]. This hardware platform is only capable of tracking two simultaneous contact points. Due to the construction principle (four IR cameras – one in each corner – and a ring of infrared LEDs hidden in the bezel) objects on the surface (e.g., fingers) can easily occlude other objects. In case two objects are aligned along one of the display diagonals they become difficult to disambiguate. These sensing limitations caused user frustration because the system would often report wrong finger positions or not recognize fingers at all. More important the limitation to two contact points (which was perceived as arbitrarily chosen by many users) posed severe constraints on the interaction style. We could often observe that users – especially novice users – tried to use more than one finger, sometimes both hands at a time to interact with on-screen objects. We could also frequently observe how users tried to interact with other body parts than their fingers, for example whole hands, the edge of the hand and even whole forearms (e.g., in an attempt to stop sliding post-its in **Brainstorm**).

One of the main motivations for our later projects discussed in Part III was to overcome these hardware limitations to enable richer interactions with the digital more akin to the ways we manipulate objects in the real world. In Chapter 6 we discussed a variety of approaches to sense this richer interaction. We have discussed approaches that embed some sort of sensing electronics into the display surface itself (cf. Section 6.1.1). For example, electronics that sense changes in capacitance or resistance but we have also seen approaches that embed infrared illumination and sensing optics into or behind thin form factor displays. Another popular approach are sensors based on a digital camera and computer vision algorithms. This approach is particularly popular because setups are relatively easy to construct – necessary parts, such as digital cameras and filters, are abundant and last but not least the possibility to leverage a wealth of existing computer vision techniques makes this approach flexible and powerful. For example many vision-based systems can track multiple fingertips and capture the shapes and outlines of arbitrary objects in contact with the surface such as whole hands. These capabilities promise interaction styles that mimic the ways we interact with the real world more closely.

In Section 6.2 we discuss our own approach to vision-based sensing of multiple simultaneous surface contacts. It works by liquid displacement inside a malleable projection surface. Our approach allows recognition of shapes and outlines of many different objects touching the surface with high precision. Our approach compliments the existing range of approaches 6.1.2. The signal is fairly unique, enabling advanced techniques such as pressure sensing or even some 3D shape reconstruction to be captured from imprints of hands and other objects. Section 6.3 summarizes our experiences from building different interactive prototypes and discusses various aspects of the systems described in the literature. It would stand to reason for tabletop computing[1] hardware matters especially; to a degree where hardware and software are interlinked and need to be considered in unison when designing and evaluating interaction styles and techniques.

Drawing both on hardware developments (cf. Chapter 6), that enable rich sensing of user interaction, and on our observations and learnings from Part II we debuted our new model for tabletop interaction in Chapter 7. This new model aims at providing more *flexibility* of interaction, allowing users to appropriate an re-purpose user interface elements and apply strategies learned in the real world. Another goal was to maintain the positive aspects we could accredit to *physicality* in our earlier prototypes. We demonstrated various ways to map rich input from vision-based interactive surfaces within virtual worlds where object behavior is controlled by a sophisticated physics simulation. The model enables a variety of open-ended, non-scripted interactions allowing users to leverage their fine-grained manipulation skills in ways similar to the real world. Allowing for and encouraging rich interactions with virtual objects, using not only multiple fingertips but novel, often in-situ designed interactions such as cupping, throwing and raking gestures. The technique allows users to interact with the virtual using multiple fingers, whole hands and physical objects as handles for interaction. We have also discussed the evolution of our technique and shown how different incarnations could be used with other hardware platforms than the one used in our setup. Our user study and general observations have revealed that users are receptive to this new model of interaction. In general they had little problems in interacting with the virtual realm. However, kinematic more traditional interaction styles were still prevalent especially during initial contact with the system. We expect that users, as they become more familiar with the capabilities of the approach, will further draw on their real world knowledge and thus develop richer interaction strategies.

Our model makes digital tabletop interaction even more “natural”. However, because the interaction – the sensed input and the displayed output – is still bound to the surface, there is a fundamental limitation in manipulating objects using the third dimension. To address this issue, we started with a discussion of several emerging projection screen materials in Chapter 8. Materials that allow for simultaneous projection onto the surface and for imaging through the surface – thus enabling us to sense the users interactions at much greater depth than possible in standard tabletop setups.

Finally, in Chapter 9 we discussed our approach to creating a continuous interaction space – allowing for rich, natural interaction *on* the surface and *above* the surface. We presented several ways to enable in-the-air interactions that allow users to interact with 3D data (rendered on a 2D display) in a as-direct-as possible fashion without the need for user instrumentation. The chapter introduces two novel, vision-based back projection tabletop setups that allow for sensing of on and above the surface interaction. Existing and new computer vision techniques have been proposed to estimate the position of hands in 3D. Our technique demonstrates how this can be achieved using an electronically switchable diffuser, a monocular camera, IR illumination and

[1] but possibly for all areas of non-desktop HCI

pixel intensity based depth estimation in our first setup. The second tabletop configuration shows how in-the-air interactions can be facilitated using a holographic projection screen and a per-pixel depth sensing camera.

In addition, various rendering techniques have been proposed to provide better feedback when interacting with 3D data on the surface from above the surface. We have also showcased some of the interactions enabled by our techniques. Clearly we have only begun to explore the possibilities unlocked by the gained interaction fidelity. While we have explored the high DOF interaction techniques in a literal context – that is 3D input and 3D data – we are convinced that our, and similar, techniques can be useful in a broad range of applications including 3D and (extended) 2D applications.

10.2 Conclusion

Going back to our problem statement presented in Section 1.2 the main goal of this thesis was to define a new model for tabletop interaction. When launching into the research for this dissertation many hardware and software solutions had been proposed and studied such as pen-based, multi-touch and tangible interaction. These interaction techniques, however have often been designed in an ad-hoc manner and studied in isolation. We wanted to gain a better more structured understanding of the design space. In order to achieve this goal we followed a threefold approach. First: a literature analysis with special focus on tabletop interaction styles. Second: explorations into the identified two main interaction styles (gesture-based and tangibles). Third: proposal, evaluation and refinement of our own model for tabletop interaction.

In order to structure and focus our investigation of tabletop interaction styles we aspired to provide answers to the following two questions:

What is the Interface? Real world interactions benefit from tactile and other rich sensory feedback. As well as from our aptitude to manipulate objects in various ways. This *physicality* is only poorly represented by most direct-touch interfaces. Often it is assumed that digital tabletops are inherently flat and made of glass or acrylic. Also it is often assumed that multi-touch simply means having the equivalent of multiple cursors controlled by several fingertips simultaneously. In this thesis we have shown that interacting through direct-touch carries much richer meaning and interfaces for this emerging class of computing devices. It may be required to think of user interfaces for digital tabletops not as an evolution of the standard desktop interaction paradigm, but one that has a whole new set of unique parameters.

In Part II we have seen that both tangible and direct-touch based interaction have promising aspects and limitations as interaction styles. Both promise a natural and graspable interaction experience, both are accredited with a low learning threshold and both offer support for co-located collaborative work and social interaction. Our aim was to explore this rich design space as a whole in order to provide a better understanding how to best use them. We would argue that our criteria of *flexibility* and *physicality* has been helpful to generalize some of the findings specific to our implementations to a broader understanding of the design space. Especially with regards to uncovering the limited flexibility afforded by these interaction styles.

In particular we have seen that physical aspects play an important role in interactive surface computing and can be supported through a variety of means. For example, those explored in this thesis such as real world metaphors in the UI, tangible objects as well as virtual objects that behave more realistically. Another way to provide users with physical affordances are means to create richer feedback at the interface level be it through passive haptic feedback – partially explored in our malleable interactive surface prototype (cf. Section 6.2 and in the literature [VMK+05, KVM+05, Sin97, SGHB07] – or through active feedback via actuated tangible objects [PMAI02] or via dynamically changeable physical buttons on direct-touch enabled surfaces [HH09a].

We have also seen that is important to compliment this physicality with a great deal of flexibility in the interface. In our explorations it has been especially problematic combining pseudo-physical behavior and appearance with scripted and therefore constrained interaction techniques. Our new model for tabletop interaction has shown ways to create more open-ended and flexible interaction techniques that allow for re-purposing and appropriation of user interface elements through rich parameters such as friction, velocity and collisions known from the real-world. We have also seen that this model was well received by users in a lab-based study. Clearly we have only begun to explore this interaction model and many questions remain unanswered as of now. For example, we have yet to build applications based on this model that go beyond literal translations of object positions. We also have to consider how many aspects that go beyond the physically possible can be modeled in with our approach. For example, copying, scaling and zooming of documents is not very well represented in our model. In general it is worthwhile considering how this model could be crossbred with more abstract, traditional interaction models such as the WIMP paradigm.

Where is the Interface? Because tabletops have planar 2D displays it is often assumed that applications and interaction should be 2D as well. Interaction with such surfaces is inherently constrained to the planar, 2D surface of the display. For many tabletop interactions this constraint may not appear to be a problem, particularly when direct manipulation with 2D content is desired. Recent research is however beginning to motivate the need for rendering 3D content on tabletops [HCC07, HC07]. Also during the evaluation of our own work on physics enabled tabletop applications [WIH+08] it became apparent that because the interaction – the sensed input and the corresponding displayed output – is still bound to the 2D surface, there is a fundamental limitation in manipulating objects using the third dimension.

We have outlined in Chapter 8 that emerging display technologies can enable sensing that goes beyond these 2D limitations. In Chapter 9 we have started to explore the space above the table as interaction space with equal value as the display surface. While we haven't fully explored this space we are convinced that tabletop setups that allow for continuous interaction on *and* above the surface are a compelling area for research and promise to be useful in many exciting application domains such as architectural or medical imaging but also 3D gaming. In the light of our initial exploration we would argue that we have done an initial step toward blurring the line between the (interactive) display and the environment it exists in. We think it is worthwhile thinking of the entire space surrounding interactive devices as potential interface with the digital. This could involve physical (inherently 3D), possibly actuated, objects in combination with tabletops supporting advanced sensing and feedback mechanisms [IHT+08, KN08, Wil07] or via rich natural hand gestures sensed in-the-air as discussed in Chapter 9.

10.3 Main Publications

Key concepts discussed in this dissertation have been published (or submitted for review) in the following peer reviewed papers:

Description	Main Publication	Additional Publications	Chapter
Gesture-based Interaction across multiple interactive surfaces.	**O. Hilliges,** L. Terrenghi, S. Boring, D. Kim, H. Richter, A. Butz **Designing for Collaborative Creative Problem Solving** *In Proceedings of 6th Creativity & Cognition, Washington D.C., USA, June 13-15, 2007*	S. Boring, **O. Hilliges**, A. Butz **A Wall-sized Focus plus Context Display** *In Proceedings of 5th IEEE Conference on Pervasive Computing and Communications, New York, NY, USA, Mar. 2007*	3
Hybrid interaction on digital tabletops. Browsing and sharing of personal photo collections.	**O. Hilliges**, D. Baur, A. Butz **Photohelix: Browsing, Sorting and Sharing Digital Photo Collections** *In Proceedings of the 2nd IEEE Tabletop Workshop, Newport, RI, USA, October 10 - 12, 2007*	**O. Hilliges**, D. Kirk **Getting Sidetracked: Display Design and Occasioning Photo-Talk with the Photohelix** *In Proceedings of 27th ACM SIGCHI Conference on Human Factors in Computing Systems, Boston, MA, USA, April 4 - 9, 2009*	4
A different approach to multi-touch sensing using liquid displacement. Provides rich sensor data and unique, organic feel when touched.	**O. Hilliges**, D. Kim, S. Izadi **Creating Malleable Interactive Surfaces using Liquid Displacement Sensing** *In Proceedings of 3rd IEEE Tabletop and Interactive Surfaces, Amsterdam, the Netherlands, October 1 - 3, 2008*		6
A new model for tabletop interaction. Exploring the intersection of rich surface input data and gaming physics simulation.	A. D. Wilson, S. Izadi, **O. Hilliges**, A. Garcia-Mendoza, D. Kirk **Bringing Physics to the Surface** *In Proceedings of 21st ACM UIST, Monterey, CA, USA, October 19 - 22, 2008, (Best Paper Award)*		7
An extension to our interaction model. Enabling interaction with 3D data on tabletops using “as-direct-as-possible” interaction techniques	**O. Hilliges**, S. Izadi, A.D. Wilson, S. Hodges, A. Garcia-Mendoza, A. Butz. **Interactions In the Air - Adding more Depth to Interactive Tabletops.** *To appear in ACM UIST '09, 2009.*		8

Figure 10.1: Overview of publications describing key aspects of this dissertation.

10.4 Future Work

Based on our experiences in developing and studying the systems described in this thesis we have identified several areas for improvement and interesting directions for future work. Obviously many of the systems discussed in Chapters 3, 4, 6.2, 7 and 9 are prototypes and each of them could benefit from additional development iterations and extended functionality. However, in this Chapter we want to highlight areas of interest for our own future research as well as interesting aspects worth consideration for the general field of tabletop and interactive surface computing.

10.4.1 Tabletop Hardware and Form Factor

In Chapter 6 we have already given an extensive overview of various tabletop setups, approaches to sensing and displaying of information. Throughout the entire Part III we have also seen how interwoven hardware and software are. Novel sensing possibilities can enable new interactions not possible before such as our in-the-air interactions which were enabled by our approaches to depth estimation (sensing). These sensing developments themselves relied on emerging display technologies to become available. Together, these changes to the "standard" tabletop computing setup have a large impact on the user experience. We would therefore argue that in the future we should not take the current status-quo of tabletop hardware as a given but actively think about shortcomings of current solutions as well as possible extensions to them. When comparing some of the more recent developments in interactive surface sensing [IHT+08, HIB+07, Wil07, KN08] with the current generation of commercially available platforms [DL01, Mic08, SMA09] we can already see that tabletop computing does not necessarily have to remain bound to the 2D display surface in terms of input and output.

The advent of advanced depth sensing cameras such as the 3DV ZSense camera used in our second prototype in Chapter 9 as well as stereo cameras open up completely new interaction possibilities, including natural hand gestures and full-body interaction. Furthermore, interaction using tangible objects could be significantly changed through sensing schemes based on these cameras. They are more capable of capturing the 3D nature of tangible objects and therefore might unlock more of the manipulation vocabulary afforded by them.

In addition to input and sensing aspects, providing richer visual feedback is an important area for future work. First, many current tabletop setups still suffer from poor resolution and limited screen-size (e.g., text legibility is often a problem) better display technologies might increase the utility of tabletop computers for many application domains. Second, tabletop interaction has often been studied in isolation but it would be interesting to look deeper into setups combining several interactive surfaces but also small, mobile devices – especially in the context of emerging hardware that allows the sensing, recognition and identification of and communication with other devices [IHT+08, HIB+07, KN08].

Third, most tabletop setups today are flat and don't provide much tactile feedback. This can be a disadvantage in terms of *eyes-free* operation when compared to standard physical buttons. Novel technologies that enable tactile feedback for direct-touch enabled devices would certainly be an interesting area for future work and some have already surfaced be it through sensing of reconfigurable passive interaction props [WWJ+09] or active controllable physical buttons [HH09a]. Finally haptic feedback could improve the sensation and therefore realism when interacting with interfaces that contain virtual objects that behave in a more realistic fashion as discussed in Chapter 7. Feeling when a finger or hand collides with an object or being able to feel when one's fingers start to intersect an object while attempting to pinch or grab that object would certainly improve the usability of our model. At the time of writing the necessary technology to provide localized, independent haptic feedback at several on-screen locations simultaneously did not exist.

10.4.2 Interaction Model

In this thesis we have proposed a new model for tabletop interaction. Although our lab-based studies have yielded encouraging results the model is far from being finished (or proven to be of real world utility).

One aspect to consider is how this model can be integrated into actual applications. Here the main challenge is to find ways to model or interface our physics-based interactions with abstract concepts only possible in the virtual (or laborious in the physical) such as creating infinite copies of objects and sending them over great distances. Also, performing simple changes to object appearance, frequent in painting, modeling or other content creation tools, such as re-coloring, applying of textures or even scaling are not very well modeled in our current solution.

Another point of interest would be to investigate ways to integrate more traditional concepts such as windows, buttons and sliders into our model. A possible solution for some of these concepts would be to mechanically construct complex machinery from individual primitives. For example, a slider could be built by using two virtual joints attached to the constraining limits and several boxes to guide and constrain the motion of the actual sliding knob.

Finally, a long-term user study, potentially a field deployment, of a tabletop device with an application based on our interaction model could be informative. It would be of great value to observe whether users would adapt richer, more natural interaction styles over the single-finger, kinematic style they are used to from desktop interaction. Furthermore, it would be interesting to find out whether the novelty of realistically behaving virtual objects wears out and if the, sometimes inevitable, imprecision of the approach would get into the way of users trying to complete their current task efficiently.

10.4.3 3D on Tabletops

The in-the-air interactions discussed in Chapter 9 are the newest development undertaken during this dissertation and are still very much a work-in-progress. Obviously sensing, rendering and interaction techniques in both discussed setups could benefit from further iterations without adding new features to the setup.

More fundamentally, a controlled study comparing the various interaction and feedback techniques is a next step in order to better understand the specifics, shortcomings and strength of our solutions. In particular, a better understanding about which influence the shadow rendering has on performance in 3D manipulation tasks would be of value. In addition to the two techniques discussed in Sections 9.3.2 and 9.6 other feedback mechanisms are conceivable. For example, the current "shadows" are not consistent with the lighting of the remaining scene where a virtual light source is positioned centered above the scene. In such a setup hand shadows should increase in size when hands travel away from the surface. In our implementation they become smaller, hence they are technically inverted. However, none of our users seemed to pick-up on this issue therefore it would be interesting to compare these two conditions. Other conditions could be reflections of the users hand (as if one was looking into a mirror) or stylized versions of the shadow or 3D representation of the entire arm (e.g., a rendering of just the fingertips).

Furthermore, it may be worthwhile exploring ways to provide additional feedback about the position of virtual objects *within* the physical space above the normal projection screen. One first step may be to project the color of objects currently under control by the user onto the users' hands – either from top or from underneath. For example, using the a switchable diffuser and two synchronized projectors as in the original SecondLight setup [IHT+08]. Other compelling scenarios may include auto-stereoscopic displays to provide even richer visual feedback.

When designing our model for tabletop interaction one of the main motivations was to get rid of scripted behavior and predefined gestures. When addressing the limitations on interaction imposed by the 2D sensing limitations on interactive surfaces, we have retreated to detecting specific pick-up and drop-off gestures. Initial results from usage observations revealed that this rigid approach – and the chosen gestures – weren't necessarily the best solution. Often users had difficulties performing the right gesture and tried to pick objects up using a variety of strategies. The approach to track fingertips and test for (virtual) object penetration somewhat mitigates the problem. Ultimately the goal would be to have a model free solution, that is an approach that does not test for a specific logical condition to be fulfilled. Instead a model that enables users to apply the necessary forces onto virtual objects in order to lift them off the surface and control their position in 3D as well as "yaw", "pitch" and "roll" would be the most literal solution. Extending the particle proxy model discussed in Section 7.3 certainly is a promising but challenging possibility.

Finally, applying the gained degrees of freedom from our sensing approaches to actual applications that operate on 3D data would be interesting as alluded to earlier in this Chapter. However, we would also argue that the additional control could be put to good use in applications that don't necessarily have to be 3D. For example, controlling z-layering in applications such as Photoshop or Powerpoint might be controlled much easier and more intuitively using our techniques.

"Iucundi Acti Labores"

– Cicero

Index

Bibliography

[AB06] Anand Agarawala and Ravin Balakrishnan. Keepin' it real: pushing the desktop metaphor with physics, piles and the pen. In *CHI '06: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1283–1292, New York, NY, USA, 2006. ACM.

[ABM+97] Maneesh Agrawala, Andrew C. Beers, Ian McDowall, Bernd Fröhlich, Mark Bolas, and Pat Hanrahan. The two-user responsive workbench: support for collaboration through individual views of a shared space. In *SIGGRAPH '97: Proceedings of the 24th annual ACM Conference on Computer Graphics and Interactive Techniques*, pages 327–332, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co.

[Ama96] Teresa M. Amabile. *Creativity in context: Update to "The social psychology of creativity."*. Westview Press, 1996.

[App08] Apple. iPhone. `http://www.apple.com/iphone/specs.html`, 2008. [cited 2008/12/19].

[AR92] N. Ambady and R. Rosenthal. Thin slices of expressive behavior as predictors of interpersonal consequences: A meta-analysis. *Psychological Bulletin*, 111(22):256–274, 1992.

[Bar89] D. Baraff. Analytical methods for dynamic simulation of non-penetrating rigid bodies. In *SIGGRAPH '89: Proceedings of the 16th annual ACM Conference on Computer Graphics and Interactive Techniques*, pages 223–232, New York, NY, USA, 1989. ACM.

[Bar94] David Baraff. Fast contact force computation for nonpenetrating rigid bodies. In *SIGGRAPH '94: Proceedings of the 21st annual ACM Conference on Computer Graphics and Interactive Techniques*, pages 23–34, New York, NY, USA, 1994. ACM.

[BBL93] Thomas Baudel and Michel Beaudouin-Lafon. Charade: remote control of objects using free-hand gestures. *Commun. ACM*, 36(7):28–35, 1993.

[BFB94] J. L. Barron, D. J. Fleet, and S. S. Beauchemin. Performance of optical flow techniques. *Int. J. Comput. Vision*, 12(1):43–77, 1994.

[BGM04] Benjamin B. Bederson, Jesse Grosjean, and Jon Meyer. Toolkit design for interactive structured graphics. *IEEE Transactions on software engineering*, 30:535–546, 2004.

[BHB07] Sebastian Boring, Otmar Hilliges, and Andreas Butz. A wall-sized focus plus context display. In *PerCom '07: Proceedings of the 5th Annual IEEE Conference on pervasive computing and communications*, 2007.

[BIH08] Alex Butler, Shahram Izadi, and Steve Hodges. Sidesight: multi-"touch" interaction around small devices. In *UIST '08: Proceedings of the 21st annual ACM Symposium on User Interface Software and Technology*, pages 201–204, New York, NY, USA, 2008. ACM.

[BKLP04] Doug A. Bowman, Ernst Kruijff, Joseph J. Jr. LaViola, and Ivan Poupyrev. *3D user interfaces: theory and practice*. Addison-Wesley/Pearson Education, 1 edition, 2004.

[BM97] Ravin Balakrishnan and I. Scott MacKenzie. Performance differences in the fingers, wrist, and forearm in computer input control. In *CHI '97: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 303–310, New York, NY, USA, 1997. ACM.

[BSK$^+$05] Steve Benford, Holger Schnädelbach, Boriana Koleva, Rob Anastasi, Chris Greenhalgh, Tom Rodden, Jonathan Green, Ahmed Ghali, Tony Pridmore, Bill Gaver, Andy Boucher, Brendan Walker, Sarah Pennington, Albrecht Schmidt, Hans Gellersen, and Anthony Steed. Expected, sensed, and desired: A framework for designing sensing-based interaction. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 12(1):3–30, 2005.

[BVBC04] Volkert Buchmann, Stephen Violich, Mark Billinghurst, and Andy Cockburn. Fingartips: gesture based direct manipulation in augmented reality. In *GRAPHITE '04: Proceedings of the 2nd international Conference on Computer Graphics and Interactive Techniques in Australasia and South East Asia*, pages 212–221, New York, NY, USA, 2004. ACM.

[BWB06] Hrvoje Benko, Andrew D. Wilson, and Patrick Baudisch. Precise selection techniques for multi-touch screens. In *CHI '06: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1263–1272, New York, NY, USA, 2006. ACM.

[CB03] Xiang Cao and Ravin Balakrishnan. Visionwand: interaction techniques for large displays using a passive wand tracked in 3d. In *UIST '03: Proceedings of the 16th annual ACM Symposium on User Interface Software and Technology*, pages 173–182, New York, NY, USA, 2003. ACM.

[CCL04] Fu Chang, Chun-Jen Chen, and Chi-Jen Lu. A linear-time component-labeling algorithm using contour tracing technique. *Computer Vision and Image Understanding*, 93(2):206–220, 2004.

[CDJG02] Ken Camarata, Ellen Yi-Luen Do, Brian R. Johnson, and Mark D. Gross. Navigational blocks: navigating information space with tangible media. In *IUI '02: Proceedings of the 7th Conference on Intelligent User Interfaces*, pages 31–38, New York, NY, USA, 2002. ACM.

[CJK+92] Christopher Codella, Reza Jalili, Lawrence Koved, J. Bryan Lewis, Daniel T. Ling, James S. Lipscomb, David A. Rabenhorst, Chu P. Wang, Alan Norton, Paula Sweeney, and Greg Turk. Interactive simulation in a multi-person virtual world. In *CHI '92: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 329–334, New York, NY, USA, 1992. ACM.

[Cla00] Andy Clark. *Mindware: An introduction to the philosophy of cognitive science*. Oxford University Press, USA, 2000.

[CMR98] Stuart K. Card, Jock D. MacKinlay, and George G. Robertson. A morphological analysis of the design space of input devices. *Readings in Intelligent User Interfaces*, pages 597–609, 1998.

[CPS+97] M. Czernuszenko, D. Pape, D. Sandin, T. DeFanti, G. L. Dawe, and M. D. Brown. The ImmersaDesk and infinity wall projection-based virtual reality displays. *ACM SIGGRAPH Computer Graphics*, 31(2):46–49, (1997).

[CRM04] A. Crabtree, T. Rodden, and J. Mariani. Collaborating around collections: informing the continued development of photoware. In *CSCW '04: Proceedings of the 2004 ACM Conference on Computer Supported Cooperative Work*, pages 396–405, New York, NY, USA, 2004. ACM Press.

[CRM+06] Mary Czerwinski, George Robertson, Brian Meyers, Greg Smith, Daniel Robbins, and Desney Tan. Large display research overview. In *CHI '06: extended abstracts of Conference on Human Factors in Computing Systems*, pages 69–74, New York, NY, USA, 2006. ACM.

[CS95] G. Crampton Smith. The hand that rocks the cradle. ID Magazine, May/June 1995.

[Csi96] M. Csikszentmihalyi. *Creativity: Flow and the psychology of discovery and invention*. Harper Collins, 1996.

[CU93] Bay-Wei Chang and David Ungar. Animation: from cartoons to the user interface. In *UIST '93: Proceedings of the 6th annual ACM Symposium on User Interface Software and Technology*, pages 45–55, New York, NY, USA, 1993. ACM.

[CWB+08] Xiang Cao, Andrew D. Wilson, Ravin Balakrishnan, Ken Hinckley, and Scott E. Hudson. Shapetouch: Leveraging contact shape on interactive surfaces. In *Proceedings of TABLETOP 2008, IEEE International Workshop on Tabletops and Interactive Surfaces*, pages 139–146, Oct 2008.

[DH08] Philip L. Davidson and Jefferson Y. Han. Extending 2d object arrangement with pressure-sensitive layering cues. In *UIST '08: Proceedings of the 21st annual ACM Symposium on User Interface Software and Technology*, pages 87–90, New York, NY, USA, 2008. ACM.

[DL01] Paul Dietz and Darren Leigh. Diamondtouch: a multi-user touch technology. In *UIST '01: Proceedings of the 14th annual ACM Symposium on User Interface Software and Technology*, pages 219–226. ACM, 2001.

[DM87] Stroebe W. Diehl M. Productivity loss in brainstorming groups: toward the solution of a riddle. *Journal of Personality and Social Psychology*, 53(3):497–509, 1987.

[DNP] DNP Denmark. DNP Holo Screen. http://www.en.dnp.dk/get/472.html. [cited 14/05/2009].

[DR04] A. R. Dennis and B. Reinicke. Beta versus vhs and the acceptance of electronic brainstorming technology. *MIS Quarterly*, 28(1), 2004.

[Dra04] Pierre Dragicevic. Combining crossing-based and paper-based interaction paradigms for dragging and dropping between overlapping windows. In *UIST '04: Proceedings of the 17th annual ACM Symposium on User Interface Software and Technology*, pages 193–196, New York, NY, USA, 2004. ACM.

[ESHD05] Kenny Erleben, Jon Sporring, Knud Henriksen, and Kenrik Dohlman. *Physics-based Animation (Graphics Series)*. Charles River Media, Inc., Rockland, MA, USA, 2005.

[Fav05] G. E. Favalora. Volumetric 3d displays and application infrastructure. *Computer*, 38(8):37–44, 2005.

[FIB95] George W. Fitzmaurice, Hiroshi Ishii, and William A. S. Buxton. Bricks: laying the foundations for graspable user interfaces. In *CHI '95: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 442–449, New York, NY, USA, 1995. ACM Press/Addison-Wesley Publishing Co.

[Fin08] FingerWorks. iGesture Pad. `http://www.fingerworks.com/index.html`, 2008. [cited 2008/12/19].

[Fis04] Kenneth P. Fishkin. A taxonomy for and analysis of tangible interfaces. *Journal on Personal and Ubiquitous Computing*, 8(5):347–358, 2004.

[FS05] Clifton Forlines and Chia Shen. Dtlens: multi-user tabletop spatial data exploration. In *UIST '05: Proceedings of the 18th annual ACM Symposium on User Interface Software and Technology*, pages 119–122, New York, NY, USA, 2005. ACM.

[FTB+00] Bernd Fröhlich, Henrik Tramberend, Andrew Beers, Maneesh Agrawala, and David Baraff. Physically-based manipulation on the responsive workbench. In *VR '00: Proceedings of the IEEE Conference on Virtual Reality 2000*, page 5, Washington, DC, USA, 2000. IEEE Computer Society.

[GB08] Tovi Grossman and Ravin Balakrishnan. Collaborative interaction with volumetric displays. In *CHI '08: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 383–392, New York, NY, USA, 2008. ACM.

[Gei98] Jörg Geißler. Shuffle, throw or take it! working efficiently with an interactive wall. In *CHI '98: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 265–266, New York, NY, USA, 1998. ACM.

[Ges78] H. Geschka. Introduction and use of idea-generating methods. *Research Management*, 21(3), 1978.

[GSW01] François Guimbretière, Maureen Stone, and Terry Winograd. Fluid interaction with high-resolution wall-size displays. In *UIST '01: Proceedings of the 14th annual ACM Symposium on User Interface Software and Technology*, pages 21–30, New York, NY, USA, 2001. ACM.

[GW06] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing (3rd Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2006.

[GW07] T. Grossman and D. Wigdor. Going deeper: a taxonomy of 3D on the tabletop. In *Proceedings 2nd IEEE International Workshop on Horizontal Interactive Human-Computer Systems TABLETOP '07*, pages 137–144, Oct 2007.

[GWB05] Tovi Grossman, Daniel Wigdor, and Ravin Balakrishnan. Multi-finger gestural interaction with 3d volumetric displays. In *SIGGRAPH '05: Proceedings of the 32nd annual ACM Conference on Computer Graphics and Interactive Techniques*, pages 931–931, New York, NY, USA, 2005. ACM.

[Han05] Jefferson Y. Han. 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*, pages 115–118, New York, NY, USA, 2005. ACM.

[HB06] Eva Hornecker and Jacob Buur. Getting a grip on tangible interaction: a framework on physical space and social interaction. In *CHI '06: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 437–446, New York, NY, USA, 2006. ACM.

[HBB07] Otmar Hilliges, Dominikus Baur, and Andreas Butz. Photohelix: Browsing, sorting and sharing digital photo collections. In *Proceedings 2nd IEEE International Workshop on Horizontal Interactive Human-Computer Systems TABLETOP '07*, pages 87–94, Los Alamitos, CA, USA, 2007. IEEE Computer Society.

[HC07] Mark Hancock and Sheelagh Carpendale. Supporting multiple off-axis viewpoints at a tabletop display. In *Proceedings 2nd IEEE International Workshop on Horizontal Interactive Human-Computer Systems TABLETOP '07*, pages 171–178, Los Alamitos, CA, USA, 2007. IEEE Computer Society.

[HCC07] Mark Hancock, Sheelagh Carpendale, and Andy Cockburn. Shallow-depth 3D interaction: Design and evaluation of one-, two- and three-touch techniques. In *CHI '07: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1147–1156, New York, NY, USA, 2007. ACM Press.

[HCS06] Uta Hinrichs, Sheelagh Carpendale, and Stacey D. Scott. Evaluating the effects of fluid interface components on tabletop collaboration. In *AVI '06: Proceedings of the Working Conference on Advanced Visual Interfaces*, pages 27–34, New York, NY, USA, 2006. ACM.

[HH09a] Chris Harrison and Scott E. Hudson. Providing dynamically changeable physical buttons on a visual display. In *CHI '09: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 299–308, New York, NY, USA, 2009. ACM.

[HH09b] Chris Harrison and Scott E. Hudson. Texture displays: a passive approach to tactile presentation. In *CHI '09: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2261–2264, New York, NY, USA, 2009. ACM.

[HHCC07] U. Hinrichs, M. Hancock, C. Collins, and S. Carpendale. Examination of text-entry methods for tabletop displays. In *Proceedings 2nd IEEE International Workshop on Horizontal Interactive Human-Computer Systems TABLETOP '07*, pages 105–112, 10–12 Oct. 2007.

[HIB+07] Steve Hodges, Shahram Izadi, Alex Butler, Alban Rrustemi, and Bill Buxton. Thinsight: versatile multi-touch sensing for thin form-factor displays. In *UIST '07: Proceedings of the 20th annual ACM Symposium on User Interface Software and Technology*, pages 259–268, New York, NY, USA, 2007. ACM.

[Hil07] O. Hilliges. Informed Browsing: Scaling Up Co-Experienced Access to Digital Media. In *Doctoral symposium of 20th ACM UIST, Newport, RI, USA*, October 2007.

[HIW+09] Otmar Hilliges, Shahram Izadi, Andrew Wilson, Steve Hodges, Armando Garcia-Mendoza, and Andreas Butz. Interactions in the air - adding more depth to interactive tabletops. In *UIST '09: Proceedings of the 22nd annual ACM Symposium on User Interface Software and Technology (to appear)*, 2009.

[HK09] Otmar Hilliges and David Kirk. Getting sidetracked: display design and occasioning photo-talk with the photohelix. In *CHI '09: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1733–1736, New York, NY, USA, 2009. ACM.

[HKI08] Otmar Hilliges, David Kim, and Shahram Izadi. Creating malleable interactive surfaces using liquid displacement sensing. In *Horizontal Interactive Human Computer Systems, 2008. TABLETOP 2008. 3rd IEEE International Workshop on*, pages 157–160, Oct. 2008.

[HRL99] Lars Erik Holmquist, Johan Redström, and Peter Ljungstrand. Token-based acces to digital information. In *Proc. HUC*, pages 234–245, London, UK, 1999. Springer-Verlag.

[HTB+07] O. Hilliges, L. Terrenghi, S. Boring, D. Kim, H. Richter, and A. Butz. Designing for collaborative creative problem solving. In *Proceedings of the 6th International Conference on Creativity & Cognition, Washington D.C., USA*, July 2007.

[HWTB07] Otmar Hilliges, Maria Wagner, Lucia Terrenghi, and Andreas Butz. The Living-Room: Browsing, Organizing and Presenting Digital Image Collections in Interactive Environments. In *Proceedings of the 3rd International Conference on Intelligent Environments, Ulm, Germany*, September 2007.

[HZR+92] Kenneth P. Herndon, Robert C. Zeleznik, Daniel C. Robbins, D. Brookshire Conner, Scott S. Snibbe, and Andries van Dam. Interactive shadows. In *UIST '92: Proceedings of the 5th annual ACM Symposium on User Interface Software and Technology*, pages 1–6, New York, NY, USA, 1992. ACM.

[IBH+08] Shahram Izadi, Alex Butler, Steve Hodges, Darren West, Malcolm Hall, Bill Buxton, and Mike Molloy. Experiences with building a thin form-factor touch and tangible tabletop. In *3rd IEEE International Workshop on Horizontal Interactive Human Computer Systems, 2008. TABLETOP 2008.*, pages 181–184, Oct. 2008.

[IHT+08] Shahram Izadi, Steve Hodges, Stuart Taylor, Dan Rosenfeld, Nicolas Villar, Alex Butler, and Jonathan Westhues. Going beyond the display: a surface technology with an electronically switchable diffuser. In *UIST '08: Proceedings of the 21st annual ACM Symposium on User Interface Software and Technology*, pages 269–278, New York, NY, USA, 2008. ACM.

[IRP+04] H. Ishii, C. Ratti, B. Piper, Y. Wang, A. Biderman, and E. Ben-Joseph. Bringing clay and sand into digital design — continuous tangible user interfaces. *BT Technology Journal*, 22(4):287–299, 2004.

[IU97] Hiroshi Ishii and Brygg Ullmer. Tangible bits: towards seamless interfaces between people, bits and atoms. In *CHI '97: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 234–241, New York, NY, USA, 1997. ACM.

[Jaz08] JazzMutant. Lemur. http://www.jazzmutant.com/lemur_overview.php, 2008. [cited 2008/12/19].

[JGAK07] Sergi Jordà, Günter Geiger, Marcos Alonso, and Martin Kaltenbrunner. The reacTable: exploring the synergy between live music performance and tabletop tangible interfaces. In *TEI '07: Proceedings of the Conference on Tangible and Embedded Interaction*, pages 139–146, New York, NY, USA, 2007. ACM.

[KB99] Hirokazu Kato and Mark Billinghurst. Marker tracking and hmd calibration for a video-based augmented reality conferencing system. In *IWAR '99: Proceedings of the 2nd IEEE and ACM International Workshop on Augmented Reality*, page 85, Washington, DC, USA, 1999. IEEE Computer Society.

[KCSG03] Russell Kruger, Sheelagh Carpendale, Stacey D. Scott, and Saul Greenberg. How people use orientation on tables: comprehension, coordination and communication. In *GROUP '03: Proceedings of the 2003 international ACM SIGGROUP conference on Supporting Group Work*, pages 369–378, New York, NY, USA, 2003. ACM.

[KCST05] Russell Kruger, Sheelagh Carpendale, Stacey D. Scott, and Anthony Tang. Fluid integration of rotation and translation. In *CHI '05: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 601–610, New York, NY, USA, 2005. ACM.

[KGH85] Myron W. Krueger, Thomas Gionfriddo, and Katrin Hinrichsen. Videoplace—an artificial reality. In *CHI '85: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 35–40, New York, NY, USA, 1985. ACM.

[KHT06] Scott R. Klemmer, Björn Hartmann, and Leila Takayama. How bodies matter: five themes for interaction design. In *Proc. DIS*, pages 140–149, New York, NY, USA, 2006. ACM.

[Kir07] David Kirk. *Turn It This Way: Remote Gesturing in Video-Mediated Communication*. Phd thesis, University of Nottingham, 2007.

[KN08] Yasuaki Kakehi and Takeshi Naemura. Ulteriorscape: Interactive optical superimposition on a view-dependent tabletop display. In *Proceedings 3rd IEEE International Workshop on Horizontal Interactive Human-Computer Systems TABLETOP '08*, pages 189–192, Oct. 2008.

[KNF+01] Scott R. Klemmer, Mark W. Newman, Ryan Farrell, Mark Bilezikjian, and James A. Landay. The designers' outpost: a tangible interface for collaborative web site. In *UIST '01: Proceedings of the 14th annual ACM Symposium on User Interface Software and Technology*, pages 1–10, New York, NY, USA, 2001. ACM.

[KNM07] Yasuaki Kakehi, Takeshi Naemura, and Mitsunori Matsushita. Tablescape plus: Interactive small-sized vertical displays on a horizontal tabletop display. In *Proceedings 2nd IEEE International Workshop on Horizontal Interactive Human-Computer Systems TABLETOP '07*, pages 155–162, Los Alamitos, CA, USA, 2007. IEEE Computer Society.

[KSRW06] D. Kirk, A. Sellen, C. Rother, and K. Wood. Understanding photowork. In *CHI '06: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 761–770, New York, NY, USA, 2006. ACM Press.

[KVM+05] Kazuto Kamiyama, Kevin Vlack, Terukazu Mizota, Hiroyuki Kajimoto, Naoki Kawakami, and Susumu Tachi. Vision-based sensor for real-time measuring of surface traction fields. *IEEE Computer Graphics and Applications*, 25(1):68–75, 2005.

[LBS85] S. K. Lee, William Buxton, and K. C. Smith. A multi-touch three dimensional touch-sensitive tablet. In *CHI '85: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 21–25, New York, NY, USA, 1985. ACM.

[LC-07] LC-Tech. FOS-PSCT optical-shutter datasheet. http://www.lctecdisplays.com/files/datasheets/FOS-PSCT.pdf, 2007. [cited 15/05/09].

[Lin06] S.E. Lindley. *The effect of the affordances offered by shared interfaces on the social behaviour of collocated groups*. Phd thesis, University of York, 2006.

[LPS+06] Jun Liu, David Pinelle, Samer Sallam, Sriram Subramanian, and Carl Gutwin. Tnt: improved rotation and translation on digital tables. In *Proceedings Graphics Interface*, pages 25–32, Toronto, Ont., Canada, Canada, 2006. Canadian Information Processing Society.

[LZB98] Andrea Leganchuk, Shumin Zhai, and William Buxton. Manual and cognitive benefits of two-handed input: an experimental study. *ACM Trans. Comput.-Hum. Interact.*, 5(4):326–359, 1998.

[Mar95] David A. Martin. Interactive display system (smartboard). us patent 5,448,263, smart technologies inc., calgary, alberta, canada, september 5 1995. `http://patft.uspto.gov/netacgi/nph-Parser?patentnumber=5448263`, 1995. [cited 2009/01/21].

[MCP+06] Meredith Ringel Morris, Anthony Cassanego, Andreas Paepcke, Terry Winograd, Ann Marie Piper, and Anqi Huang. Mediating group dynamics through tabletop interface design. *IEEE Computer Graphics & Applications*, 26(5):65–73, 2006.

[MH06] Tomer Moscovich and John F. Hughes. Multi-finger cursor techniques. In *Proceedings Graphics Interface*, pages 1–7, Toronto, Ont., Canada, Canada, 2006. Canadian Information Processing Society.

[MHPW06] Meredith Ringel Morris, Anqi Huang, Andreas Paepcke, and Terry Winograd. Cooperative gestures: multi-user gestural interactions for co-located groupware. In *CHI '06: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1201–1210, New York, NY, USA, 2006. ACM.

[Mic08] Microsoft. Microsoft Surface. `http://www.microsoft.com/surface/`, 2008. [cited 2008/09/19].

[Min84] Margaret R. Minsky. Manipulating simulated objects with real-world gestures using a force and position sensitive screen. In *SIGGRAPH '84: Proceedings of the 11th annual ACM Conference on Computer Graphics and Interactive Techniques*, pages 195–203, New York, NY, USA, 1984. ACM.

[MIO+04] Mitsunori Matsushita, Makoto Iida, Takeshi Ohguro, Yoshinari Shirai, Yasuaki Kakehi, and Takeshi Naemura. Lumisight table: a face-to-face collaboration support system that optimizes direction of projected information to each stakeholder. In *CSCW '04: Proceedings of the 2004 ACM Conference on Computer Supported Cooperative Work*, pages 274–283, New York, NY, USA, 2004. ACM.

[ML04] Shahzad Malik and Joe Laszlo. Visual touchpad: a two-handed gestural input device. In *ICMI '04: Proceedings of the 6th International Conference on Multimodal Interfaces*, pages 289–296, New York, NY, USA, 2004. ACM.

[MPW06] Meredith Ringel Morris, Andreas Paepcke, and Terry Winograd. Teamsearch: Comparing techniques for co-present collaborative search of digital media. In *Proceedings 1st IEEE International Workshop on Horizontal Interactive Human-Computer Systems TABLETOP '06*, pages 97–104, Washington, DC, USA, 2006. IEEE Computer Society.

[MPWS06] Meredith Ringel Morris, Andreas Paepcke, Terry Winograd, and Jeannie Stamberger. TeamTag: exploring centralized versus replicated controls for co-located tabletop groupware. In *CHI '06: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1273–1282, New York, NY, USA, 2006. ACM.

[MR97] Nobuyuki Matsushita and Jun Rekimoto. Holowall: designing a finger, hand, body, and object sensitive wall. In *UIST '97: Proceedings of the 10th annual ACM Symposium on User Interface Software and Technology*, pages 209–210, New York, NY, USA, 1997. ACM.

[MRD06] Ali Mazalek, Matthew Reynolds, and Glorianna Davenport. Tviews: An extensible architecture for multiuser digital media tables. *IEEE Computer Graphics & Applications*, 26(5):47–55, 2006.

[MRD07] A. Mazalek, M. Reynolds, and G. Davenport. The tviews table in the home. In *Proc. Second Annual IEEE International Workshop on Horizontal Interactive Human-Computer Systems TABLETOP '07*, pages 52–59, 10–12 Oct. 2007.

[MSW92] Richard Mander, Gitta Salomon, and Yin Yin Wong. A "pile" metaphor for supporting casual organization of information. In *CHI '92: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 627–634, New York, NY, USA, 1992. ACM.

[NMKT05] Kousuke Nakashima, Takashi Machida, Kiyoshi Kiyokawa, and Haruo Takemura. A 2D-3D integrated environment for cooperative work. In *VRST '05: Proceedings of the annual ACM Symposium on Virtual Reality Software and Technology*, pages 16–22, New York, NY, USA, 2005. ACM.

[Nvi08] Nvidia Corporation. Nvidia PhysX. `http://www.nvidia.com/object/nvidia_physx.html`, 2008. [cited 04/05/2009].

[Osb53] AF Osborn. *Applied imagination: Principles and Procedures of Creative Thinking*. Charles Scribner's Sons, 1953.

[Per06] Perceptivepixel. `http://www.perceptivepixel.com/`, 2006. [cited 2009/01/21].

[PKS+08] Peter Peltonen, Esko Kurvinen, Antti Salovaara, Giulio Jacucci, Tommi Ilmonen, John Evans, Antti Oulasvirta, and Petri Saarikko. It's mine, don't touch!: interactions at a large multi-touch display in a city centre. In *CHI '08: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1285–1294, New York, NY, USA, 2008. ACM.

[PMAI02] Gian Pangaro, Dan Maynes-Aminzade, and Hiroshi Ishii. The actuated workbench: computer-controlled actuation in tabletop tangible interfaces. In *UIST '02: Proceedings of the 15th annual ACM Symposium on User Interface Software and Technology*, pages 181–190, New York, NY, USA, 2002. ACM.

[POMW06] Anne Marie Piper, Eileen O'Brien, Meredith Ringel Morris, and Terry Winograd. Sides: a cooperative tabletop computer game for social skills development. In *CSCW '06: Proceedings of the 20th Conference on Computer Supported Cooperative Work*, pages 1–10, New York, NY, USA, 2006. ACM.

[PPK00] Ken Perlin, Salvatore Paxia, and Joel S. Kollin. An autostereoscopic display. In *SIGGRAPH '00: Proceedings of the 28th annual ACM Conference on Computer Graphics and Interactive Techniques*, pages 319–326, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.

[PRI02] Ben Piper, Carlo Ratti, and Hiroshi Ishii. Illuminating clay: a 3-d tangible interface for landscape analysis. In *CHI '02: Proceedings of the SIGCHI Conference on*

Human Factors in Computing Systems, pages 355–362, New York, NY, USA, 2002. ACM.

[Pro09] Professional Plastic. Acrylite®endlighten. http://www.professionalplastics.com/ACRYLITEENDLIGHTEN, 2009. [cited 07/05/2009].

[PSJ+07] Peter Peltonen, Antti Salovaara, Giulio Jacucci, Tommi Ilmonen, Carmelo Ardito, Petri Saarikko, and Vikram Batra. Extending large-scale event participation with user-created mobile media on a public display. In *MUM '07: Proceedings of the 6th international Conference on Mobile and Ubiquitous Multimedia*, pages 131–138, New York, NY, USA, 2007. ACM.

[RCB+05] George Robertson, Mary Czerwinski, Patrick Baudisch, Brian Meyers, Daniel Robbins, Greg Smith, and Desney Tan. The large-display user experience. *IEEE Computer Graphics and Applications*, 25(4):44–51, 2005.

[RCL+98] George Robertson, Mary Czerwinski, Kevin Larson, Daniel C. Robbins, David Thiel, and Maarten van Dantzich. Data mountain: using spatial memory for document management. In *UIST '98: Proceedings of the 11th annual ACM Symposium on User Interface Software and Technology*, pages 153–162, New York, NY, USA, 1998. ACM.

[Rek02] Jun Rekimoto. Smartskin: an infrastructure for freehand manipulation on interactive surfaces. In *CHI '02: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 113–120, New York, NY, USA, 2002. ACM.

[RFSM04] Kathy Ryall, Clifton Forlines, Chia Shen, and Meredith Ringel Morris. Exploring the effects of group size and table size on interactions with tabletop shared-display groupware. In *CSCW '04: Proceedings of the 2004 ACM Conference on Computer Supported Cooperative Work*, pages 284–293, New York, NY, USA, 2004. ACM.

[RGS+06] Adrian Reetz, Carl Gutwin, Tadeusz Stach, Miguel Nacenta, and Sriram Subramanian. Superflick: a natural and efficient technique for long-distance object placement on digital tables. In *Proceedings Graphics Interface*, pages 163–170, Toronto, Ont., Canada, Canada, 2006. Canadian Information Processing Society.

[RMB+98] Mitchel Resnick, Fred Martin, Robert Berg, Rick Borovoy, Vanessa Colella, Kwin Kramer, and Brian Silverman. Digital manipulatives: new toys to think with. In *CHI '98: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 281–287, New York, NY, USA, 1998. ACM Press/Addison-Wesley Publishing Co.

[RMRS+04] Meredith Ringel-Morris, Kathy Ryall, Chia Shen, Clifton Forlines, and Frederic Vernier. Beyond "social protocols": multi-user coordination policies for co-located groupware. In *CSCW '04: Proceedings of the 2004 ACM Conference on Computer Supported Cooperative Work*, pages 262–265, New York, NY, USA, 2004. ACM.

[RRS+04] Meredith Ringel, Kathy Ryall, Chia Shen, Clifton Forlines, and Frederic Vernier. Release, relocate, reorient, resize: fluid techniques for document sharing on multi-user interactive tables. In *CHI '04: extended abstracts on Human factors in computing systems*, pages 1441–1444, New York, NY, USA, 2004. ACM.

[RS00] Jun Rekimoto and Eduardo Sciammarella. Toolstone: effective use of the physical manipulation vocabularies of input devices. In *UIST '00: Proceedings of the 13th annual ACM Symposium on User Interface Software and Technology*, pages 109–117, New York, NY, USA, 2000. ACM.

[RSC87] William T. Reeves, David H. Salesin, and Robert L. Cook. Rendering antialiased shadows with depth maps. In *SIGGRAPH '87: Proceedings of the 14th annual ACM Conference on Computer Graphics and Interactive Techniques*, pages 283–291, New York, NY, USA, 1987. ACM.

[RUO01] Jun Rekimoto, Brygg Ullmer, and Haruo Oba. DataTiles: a modular platform for mixed physical and graphical interactions. In *CHI '01: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 269–276, New York, NY, USA, 2001. ACM.

[RW03] K. Rodden and K.R. Wood. How do people manage their digital photographs? In *CHI '03: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 409–416, New York, NY, USA, 2003. ACM Press.

[RWC99] R. Raskar, G. Welch, and Wei-Chao Chen. Table-top spatially-augmented realty: bringing physical models to life with projected imagery. In *Proceedings 2nd IEEE and ACM International Workshop on Augmented Reality (IWAR '99)*, pages 64–71, 20–21 Oct. 1999.

[SAL06] Sriram Subramanian, Dzimitry Aliakseyeu, and Andrés Lucero. Multi-layer interaction for digital tables. In *UIST '06: Proceedings of the 19th annual ACM Symposium on User Interface Software and Technology*, pages 269–272, New York, NY, USA, 2006. ACM.

[SBD99] Jason Stewart, Benjamin B. Bederson, and Allison Druin. Single display groupware: a model for co-present collaboration. In *CHI '95: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 286–293, New York, NY, USA, 1999. ACM.

[SC94] Olin Shivers and Brian Carlstrom. *The scsh manual*. MIT Laboratory for Computer Science, December 1994. Current version available at `http://scsh.net`.

[SGHB07] J. D. Smith, T. C. N. Graham, D. Holman, and J. Borchers. Low-cost malleable surfaces with multi-touch pressure sensitivity. In *Proceedings 2nd IEEE International Workshop on Horizontal Interactive Human-Computer Systems TABLETOP '07*, pages 205–208, 10–12 Oct. 2007.

[SGM03] Stacey D. Scott, Karen D. Grant, and Regan L. Mandryk. System guidelines for co-located, collaborative work on a tabletop display. In *ECSCW'03: Proceedings of the 8th European Conference on Computer Supported Cooperative Work*, pages 159–178, Norwell, MA, USA, 2003. Kluwer Academic Publishers.

[SH03] Abigail J. Sellen and Richard H. R. Harper. *The myth of the paperless office*. MIT Press, 2003.

[Sin97] Michael Sinclair. The haptic lens. In *SIGGRAPH '97: Proceedings of the 24th annual ACM Conference on Computer Graphics and Interactive Techniques*, page 179, New York, NY, USA, 1997. ACM.

[SlH+99] Norbert A. Streitz, Jörg Geißler, Torsten Holmer, Shin'ichi Konomi, Christian Müller-Tomfelde, Wolfgang Reischl, Petra Rexroth, Peter Seitz, and Ralf Steinmetz. i-land: an interactive landscape for creativity and innovation. In *CHI '99: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 120–127, New York, NY, USA, 1999. ACM.

[SLM+01] Chia Shen, Neal Lesh, Baback Moghaddam, Paul Beardsley, and Ryan Scott Bardsley. Personal digital historian: user interface design. In *CHI '01: extended abstracts on Human Factors in Computing Systems*, pages 29–30, New York, NY, USA, 2001. ACM.

[SLV03] Chia Shen, Neal Lesh, and Frédéric Vernier. Personal digital historian: story sharing around the table. *interactions*, 10(2):15–22, 2003.

[SMA03] SMART. SMART Technologies DViT. `http://www.smarttech.com/DViT/`, 2003. [cited 2009/12/09].

[SMA09] SMART. SMART Technologies, SMART Table. `http://www2.smarttech.com/st/en-US/Products/SMART+Table/`, 2009. [cited 2009/01/21].

[SMG+05] Daniel J. Sandin, Todd Margolis, Jinghua Ge, Javier Girado, Tom Peterka, and Thomas A. DeFanti. The varrier$^{\text{TM}}$autostereoscopic virtual reality display. In *SIGGRAPH '05: Proceedings of the 32nd annual ACM Conference on Computer Graphics and Interactive Techniques*, pages 894–903, New York, NY, USA, 2005. ACM.

[SRF+06] Chia Shen, Kathy Ryall, Clifton Forlines, Alan Esenther, Frederic D. Vernier, Katherine Everitt, Mike Wu, Daniel Wigdor, Meredith Ringel Morris, Mark Hancock, and Edward Tse. Informing the design of direct-touch tabletops. *IEEE Computer Graphics and Applications*, 26(5):36–46, 2006.

[SS91] Andrew Sears and Ben Shneiderman. High precision touchscreens: Design strategies and comparisons with a mouse. *International Journal of Man-Machine Studies*, 34:593–613, 1991.

[STB07] Garth Shoemaker, Anthony Tang, and Kellogg S. Booth. Shadow reaching: a new perspective on interaction for large displays. In *UIST '07: Proceedings of the 20th annual ACM Symposium on User Interface Software and Technology*, pages 53–56, New York, NY, USA, 2007. ACM.

[Sum04] Sumitomo. Sumitomo chemical, lumisty film. `http://www.sumitomo-chem.co.jp/english/research/develop_basic.html`, 2004. [cited 2009/1/21].

[SWS+02] Olov Ståhl, Anders Wallberg, Jonas Söderberg, Jan Humble, Lennart E. Fahlén, Adrian Bullock, and Jenny Lundberg. Information exploration using the pond. In *Proc. CVE*, pages 72–79, New York, NY, USA, 2002. ACM.

[Tac05] Tactiva. TactaPad. `http://tactiva.com/tactapad.html`, 2005. [cited 2008/12/19].

[TKR+08] Lucia Terrenghi, David Kirk, Hendrik Richter, Sebastian Krämer, Otmar Hilliges, and Andreas Butz. Physical handles at the interactive surface: exploring tangibility and its benefits. In *AVI '08: Proceedings of the Working Conference on Advanced Visual Interfaces*, pages 138–145, New York, NY, USA, 2008. ACM.

[TKSI07] Lucia Terrenghi, David Kirk, Abigail Sellen, and Shahram Izadi. Affordances for manipulation of physical versus digital media on interactive surfaces. In *CHI '07: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1157–1166, New York, NY, USA, 2007. ACM.

[UI97] Brygg Ullmer and Hiroshi Ishii. The metaDESK: models and prototypes for tangible user interfaces. In *UIST '97: Proceedings of the 10th annual ACM Symposium on User Interface Software and Technology*, pages 223–232, New York, NY, USA, 1997. ACM.

[UI99a] Brygg Ullmer and Hiroshi Ishii. mediablocks: tangible interfaces for online media. In *CHI '99: extended abstracts on Human Factors in Computing Systems*, pages 31–32, New York, NY, USA, 1999. ACM.

[UI99b] John Underkoffler and Hiroshi Ishii. Urp: a luminous-tangible workbench for urban planning and design. In *CHI '99: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 386–393, New York, NY, USA, 1999. ACM.

[UI00] Brygg Ullmer and Hiroshi Ishii. Emerging frameworks for tangible user interfaces. *IBM Systems Journal*, 39(3-4):915–931, 2000.

[VG88] A.B. Van Gundy. *Techniques of structured problem solving*. John Wiley & Sons Inc, 1988.

[VG07] Nicolas Villar and Hans Gellersen. A malleable control structure for softwired user interfaces. In *TEI '07: Proceedings of the 1st international conference on Tangible and Embedded Interaction*, pages 49–56, New York, NY, USA, 2007. ACM.

[VGREG06] Nicolas Villar, Kiel Mark Gilleade, Devina Ramduny-Ellis, and Hans Gellersen. The voodooio gaming kit: a real-time adaptable gaming controller. In *ACE '06: Proceedings of the 2006 ACM SIGCHI Conference on Advances in Computer Entertainment Technology*, page 1, New York, NY, USA, 2006. ACM.

[VMK+05] Kevin Vlack, Terukazu Mizota, Naoki Kawakami, Kazuto Kamiyama, Hiroyuki Kajimoto, and Susumu Tachi. Gelforce: a vision-based traction field computer interface. In *CHI '05: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1154–1155, New York, NY, USA, 2005. ACM.

[WB03] Mike Wu and Ravin Balakrishnan. Multi-finger and whole hand gestural interaction techniques for multi-user tabletop displays. In *UIST '03: Proceedings of the 16th annual ACM Symposium on User Interface Software and Technology*, pages 193–202, New York, NY, USA, 2003. ACM.

[WC05] Andrew D. Wilson and Edward Cutrell. Flowmouse : A computer vision-based pointing and gesture input device. In *INTERACT'05: International Conference on Human-Computer Interaction*, pages 565–578, 2005.

[Wel93] Pierre Wellner. Interacting with paper on the digitaldesk. *Communications of the ACM*, 36(7):87–96, 1993.

[Wes99] Wayne Westerman. *Hand Tracking, Finger Identification and Chordic Manipulation on a Multi-Touch Surface*. PhD thesis, University of Delaware, 1999.

[WFB+07] Daniel Wigdor, Clifton Forlines, Patrick Baudisch, John Barnwell, and Chia Shen. Lucid touch: a see-through mobile device. In *UIST '07: Proceedings of the 20th annual ACM Symposium on User Interface Software and Technology*, pages 269–278, New York, NY, USA, 2007. ACM.

[WIH+08] Andrew D. Wilson, Shahram Izadi, Otmar Hilliges, Armando Garcia-Mendoza, and David Kirk. Bringing physics to the surface. In *UIST '08: Proceedings of the 21st annual ACM Symposium on User Interface Software and Technology*, pages 67–76, New York, NY, USA, 2008. ACM.

[Wil04] Andrew D. Wilson. Touchlight: an imaging touch screen and display for gesture-based interaction. In *ICMI '04: Proceedings of the 6th Conference on Multimodal Interfaces*, pages 69–76, New York, NY, USA, 2004. ACM.

[Wil05] Andrew D. Wilson. PlayAnywhere: a compact interactive tabletop projection-vision system. In *UIST '05: Proceedings of the 18th annual ACM Symposium on User Interface Software and Technology*, pages 83–92, New York, NY, USA, 2005. ACM.

[Wil06] Andrew D. Wilson. Robust computer vision-based detection of pinching for one and two-handed gesture input. In *UIST '06: Proceedings of the 19th annual ACM*

Symposium on User Interface Software and Technology, pages 255–258, New York, NY, USA, 2006. ACM.

[Wil07] A. D. Wilson. Depth-sensing video cameras for 3d tangible tabletop interaction. In *Proceedings 2nd IEEE International Workshop on Horizontal Interactive Human-Computer Systems TABLETOP '07*, pages 201–204, 10–12 Oct. 2007.

[WMW09] Jacob O. Wobbrock, Meredith Ringel Morris, and Andrew D. Wilson. User-defined gestures for surface computing. In *CHI '09: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1083–1092, New York, NY, USA, 2009. ACM.

[WPR+07] Daniel Wigdor, Gerald Penn, Kathy Ryall, Alan Esenther, and Chia Shen. Living with a tabletop: Analysis and observations of long term office use of a multi-touch table. In *Proceedings 2nd IEEE International Workshop on Horizontal Interactive Human-Computer Systems TABLETOP '07*, pages 60–67, Los Alamitos, CA, USA, 2007. IEEE Computer Society.

[WSR+06] Mike Wu, Chia Shen, Kathy Ryall, Clifton Forlines, and Ravin Balakrishnan. Gesture registration, relaxation, and reuse for multi-point direct-touch surfaces. In *Proceedings 1st IEEE International Workshop on Horizontal Interactive Human-Computer Systems TABLETOP '06*, pages 185–192, Washington, DC, USA, 2006. IEEE Computer Society.

[WWJ+09] Malte Weiss, Julie Wagner, Yvonne Jansen, Roger Jennings, Ramsin Khoshabeh, James D. Hollan, and Jan Borchers. Slap widgets: bridging the gap between virtual and physical controls on tabletops. In *CHI '09: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 481–490, New York, NY, USA, 2009. ACM.

[WWL07] Jacob O. Wobbrock, Andrew D. Wilson, and Yang Li. Gestures without libraries, toolkits or training: a $1 recognizer for user interface prototypes. In *UIST '07: Proceedings of the 20th annual ACM Symposium on User Interface Software and Technology*, pages 159–168, New York, NY, USA, 2007. ACM.

[ZM98] Shumin Zhai and Paul Milgram. Quantifying coordination in multiple DOF movement and its application to evaluating 6 DOF input devices. In *CHI '98: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 320–327. ACM Press, 1998.