



INPUT TECHNOLOGIES AND TECHNIQUES

Ken Hinckley and Daniel Wigdor
Microsoft Research

INTRODUCTION: A SHIFTING LANDSCAPE

The barometer has fallen and a brisk wind blows. Input and interaction is not what it used to be.

The gusts drift over the line in the sand that separates input from output. The academic division between input and output as topics of study was never a principled one to begin with. But this is perhaps clearer now than ever before: many people now primarily experience computing by direct-touch input on their displays. The shifting sands are rapidly unearthing a future where any surface on which we can display information will also serve as a surface that sees, hears, feels, and otherwise senses all of our interactions. As some would have it, the encroaching dunes have all but buried the mouse, keyboard and other indirect input devices as archeological curiosities, to be unearthed by a future generation of researchers, explorers, and scoundrels.

What is the prospector of innovative devices and interaction techniques to make of this new landscape? Is it a desert or is it an oasis? In our view it is a little bit of each. Everyone is now familiar with multi-touch as the defining example of direct interaction, but it is not the whole story. As our colleague Bill Buxton constantly likes to remind us, a key lesson to remember is the following:

Everything, including touch, is best for something and worst for something else.

The goal for this chapter is to help you understand why, how, and under what circumstances, by way of a survey of illustrative examples, devices, techniques, and conceptual tools. It will also help you understand why direct interaction is about much more than just multi-touch. Direct interaction not only includes many variations on touch itself, but also includes modalities such as pen input, motion and postural sensing, and proximal and spatial interactions in the physical space beyond the display. Each of these in turn exhibits its own idiosyncratic strengths and weaknesses.

Every Coin Has Two Sides

Having said that, is it even desirable to think of devices and modalities in terms of strengths and weaknesses? Perhaps not, and

indeed we encourage the reader to move beyond this categorical mindset. Input modalities share a number of common properties. If the designer understands these properties thoroughly, then nearly any property of an input device can be turned to one's advantage when used appropriately in an interface design.

For example, take the property of *mechanical intermediary*. A modality either requires a mechanical intermediary, such as a pen, or it does not, as is the case with direct-touch input. Think of this property of *mechanical intermediary* as a coin with two sides: heads requires an intermediary, tails does not. Which side of the coin should we choose? Well, of course, it depends.

If we choose heads, then *the pen* has an advantage because it employs a mechanical intermediary. A stylus is a familiar tool to which users may bring highly developed skills for handwriting, sketching, and drawing. The body of the pen provides a lever-arm that affords a tripod grip for precise control, and its tapered tip enables one to indicate small objects on the screen. It also affords a purchase for secondary controls such as buttons, an eraser-head, or perhaps a trigger (as in that of an air-brush, for example, which modulates the flow of paint). A skilled interaction designer can leverage these attributes to produce compelling user experiences.

If we instead choose tails, on the other side of the coin, then *touch* has an advantage because it does *not* require a mechanical intermediary. Unlike a stylus, there is nothing to lose. The user always carries their finger with them. Furthermore, the user can start interacting immediately. There is no additional *acquisition time* required to grasp or unsheathe a stylus. This becomes a critical factor in mobile interaction, where interacting with the real world and other persons is the user's primary task, forcing users to wedge their interactions with a device into ever-narrowing fragments of time and attention.

That is, the property of *mechanical intermediary* shared by both the pen and touch input modalities is neither inherently good nor inherently bad. It is just a property. Correctly choosing heads or tails depends on what the user is trying to accomplish, as well as how that task fits into the larger set of activities that a designer seeks to support with an interface. The *exact same property* of a modality that offers a marked advantage for one task, or context of use, turns about face and becomes a crippling liability in another. In our example, the requirement of a mechanical intermediary is both a strength and a weakness for the pen—as is the lack of an intermediary for touch. Having these two opposing sides makes the coin no less valuable. Indeed, our ability to trade off these two opposing facets against one another is precisely what gives the coin its value in the currency of design.

Hopefully by now the lesson is clear. The designer should not just extol the virtues of a particular modality. The designer does not truly understand an input until he can articulate its shortcomings just as thoroughly as he can articulate its advantages. The emerging trend of user experiences that integrate both pen and touch inputs (as well as other modalities) illustrates the practical need for this perspective (Hinckley et al. 2010; Zeleznik, Bragdon, Ko, et al. 2010; Frisch, Heydekorn, and Dachsel 2009; Brandl et al. 2008).

From a Competitive to a Cooperative Landscape of Devices

Let's now step away from the computer for a moment and look at the example of handwriting with pencil and paper. These are "devices" that one interacts with in the real world, after all. Consider this simple question:

Which hand do you write with, right or left?

When we give talks on this subject, we have the audience raise their hands. Who uses their right hand? Do we have any left-handers in the audience? Which hand would you, the reader, raise? Now, of course, we have led you into a trap, because:

You are all wrong. No matter which hand you raised, you are wrong.

This is not a trick question. Rather the question is fundamentally ill-posed. People write with both hands, as demonstrated by Guiard (Guiard 1987):

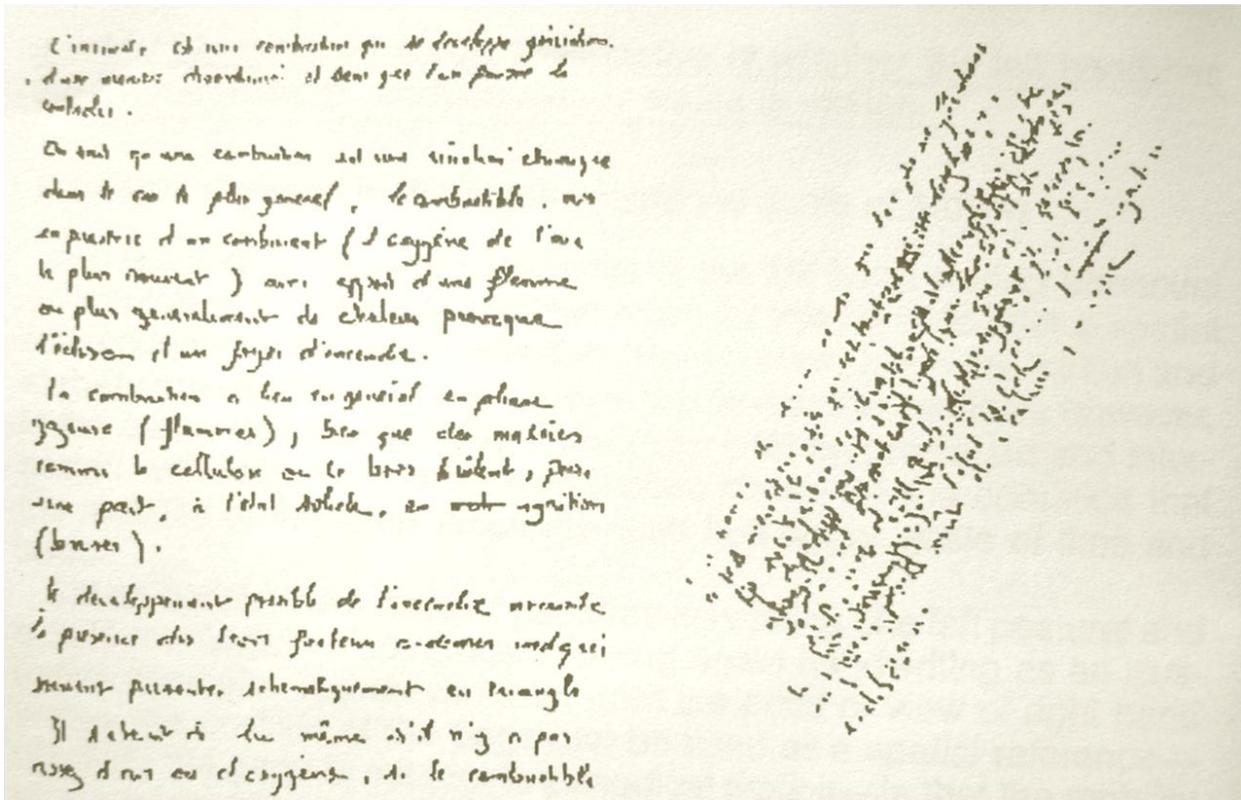


Figure 9.1. *Guiard's transfer paper experiment (Guiard 1987), with the full sheet of paper shown on the left, and the impressions left on an underlying sheet of transfer paper on the right. This shows how the nonpreferred hand orients and shifts the sheet of paper to suit the action of the preferred hand while writing.*

Figure reprinted from "Asymmetric division of labor in human skilled bimanual action: The kinematic chain as a model," by Yves Guiard, in The Journal of Motor Behavior, 19 (4), 1987, Taylor & Francis Ltd, <http://www.informaworld.com>. Reprinted by permission of the publisher.

What the above figure shows is the result of writing on a sheet of paper (**Error! Reference source not found.**). But on the right, we see the impressions left by the pen on a sheet of transfer paper surreptitiously left underneath. That is, it records the movements of the pen relative to the desk, rather than relative to the paper, and it reveals that the nonpreferred hand dynamically adjusts the position of the paper to suit the action of the preferred hand. Hence, both hands play a role in handwriting, but each hand plays a *specialized* role that is *complementary* to the other hand.

In the same way that one's hands are specialized, so too we can think of input modalities and devices as taking on specialized roles in a larger ecosystem of interaction.

In both the popular press and in research papers, one often sees input modalities framed in *competitive* terms. How many times have we seen headlines proclaiming that the next magical device will now make the mouse and keyboard obsolete? How many studies have we seen that compare device A to device B for some task? This implies a winner, as in a football match, where either A or B achieves victory. These type of headlines and studies beg research questions, even if only implicitly, of the form:

Which is better, touch or stylus? (or the mouse, or motion sensing, or freehand gesture – or name your favorite input modality, device, or interaction technique?)

If we find ourselves asking a *Which is best?* question of this sort then we must once again recognize it as an ill-posed query. We are not saying that studies that compare individual techniques cannot be illuminating. They can be, if done in a principled manner that carefully assesses and controls for the factors underlying performance, rather than just comparing two arbitrary techniques. However, isolated studies cannot be the only things that we publish, nor should they be framed and extrapolated

without due consideration of the ecology of devices and techniques in which they reside.

Following Guiard, we believe the correct question to ask is one which takes a *cooperative* perspective, as well as a broad view of input modalities and form factors. There are multiple modalities and devices. We shouldn't try to do everything with any one of them. We should instead seek to understand how input modalities and techniques can complement one another, such that the advantages of one make up for the shortcomings of others.

Interactive system design should encourage a division of labor between a collection of input devices, sensing modalities, and interaction techniques that together sum to more than the whole of the parts. We should seek out logical design principles (or rules-of-thumb) that help to guide the ways in which we combine techniques. That is, we should frame our input research and system designs around questions of the form:

What is the logic of the division of labor between touch, pen, motion and postural sensing, proximal and spatial interactions beyond the display, and a diverse ecology of devices and form factors?

If we can succeed in bringing about this shift in perspective, then perhaps this chapter can form a more enduring survey of the terrain than the gold-rush that the industry is witnessing to do anything and everything with touch, and with touch alone.

A Multiple-Modality Perspective

The discussion above implies that the reader should take a multiple-modality perspective on the design of interactive systems. The iPhone has been lauded for its multi-touch interface, but don't forget the additional modalities such as the use of proximity sensing to avoid "ear dialing" (Dietz and Yerazunis 2001), or the use of orientation sensing to automatically rotate the screen (Hinckley et al. 2000; Schmidt, Beigl, and Gellersen 1999), which both enrich the experience the device offers.

If we accept, then, that multi-touch is likely the tip of the iceberg in terms of what is possible for a natural and compelling direct-input user experience, then the combination and infusion of touch with rich sensory information from other modalities represents the great frozen massif beneath the water-line in terms of future innovation and research contributions. Rather than focusing on one modality, such as touch, our collective goal as a community of researchers, designers, and practitioners should be to understand how to most effectively design systems that use input modalities in combination, where the strengths of one compensate for the limitations of others. When a system does not have to provide coverage of all possible functions with a single input modality, implicitly this leads one to ask where each modality should be used to best advantage, where a particular modality should not be used, and in what cases modalities should perhaps be treated interchangeably (Hinckley et al. 2010).

Nonetheless, we must be pragmatic and recognize that at times an interface design must focus on a single modality. But even if your intention is to design a single-modality user experience, such as for a touch-screen device, our belief is that you will design a better and more compelling experience if you do so with a broad perspective of the limitations of that modality, as well as good working knowledge of the capabilities of other modalities. Perhaps you can even anticipate future developments, and dovetail your design to intersect with capabilities that other modalities may add to the user experience in future generations of a device or system, or enable richer capabilities in the presence of secondary sensors or peripherals that augment a system's core capabilities.

The Breadth/Depth Dichotomy

With regards to systems and devices, there are two basic design strategies. We can design for breadth, supporting a wide ecosystem of hardware and modalities with a one-size-fits-all design. Designing for breadth creates a common platform that unifies many different devices and experiences, with all the benefits (as well as the drawbacks) that are inherent in an accretion of features and interfaces. Or, we can design for depth, with a core-sample carefully drilled through the many strata of target market, user experience, input modality, display size and form factor, operating system, and application design. The market is showing that design-for-depth can yield simplification of user interfaces as well as compelling experiences. Yet the market has also shown that breadth has immense value when done well. Which way to turn? We refer to this fundamental design dilemma as the *breadth/depth dichotomy*.

This dichotomy pervades every element of design as well as software and hardware development. Successfully addressing this tension will be a key challenge for designers of software and devices supporting direct interaction. It is as if we are lost at sea and we await the arrival of an automated method to 'convert' applications and experiences for different contexts of use-- yet none is apparent on the horizon. This chapter aims to provide the reader with a modest navigational tool to help find your way through this unsettled ocean of design decisions and their subtle implications.

Whether one adopts a uni-modal or multi-modal perspective, excellence in user interface design requires tailoring the interface

to the input method. This tailoring extends to form factor as well as modality. Multi-touch interaction with a phone is often conducted with thumbs and while holding the device. Multi-touch interaction with wall-mounted displays is conducted with fingertips and arms extended. The physicality of the interaction is clearly different, as is the context of use and likely also the tasks that the user will be performing. As another example, many common websites and applications have both a desktop and a mobile edition. Students of Human-Computer Interaction (HCI) should consider these, and ask: *Does each have a different user interface model? What are commonalities, and what are the differences?* The answer will vary depending on the application, because no standard has yet emerged.

This further extends to a society of devices that users expect to work together, with consistent experiences that are nonetheless tailored to the form factor. User interface consistency should not be viewed as a rigid construct, but rather a flexible one that maintains *consistency of user expectations* given the form-factor, input modality, context of use, and the specific task at hand. Handhelds, e-readers, tablets, booklets, slates, desktops, table-tops, laptops, and wall-mounted displays each have their own unique affordances, and each may be encountered by the same user in varying contexts. Thus, from the user's perspective, the most "consistent" user experience may in fact be one that is *not consistent* with regards to the specific actions that the user must articulate to use a given form-factor.

Designing User Interfaces: The Loss of a Closely Held Abstraction

The widespread shift to direct interaction, in service of mobility as well as other concerns, has ushered in a storm front that batters us on our voyage of discovery. We have arrived at a strange land lashed by impetuous gales that one must weather while avoiding the uncharted seamounts and rocky shoals that abound everywhere on its wilderness coast. Yet, in the lee of the tempest, islands of compelling new interactions, form factors, and user experiences can be discovered that tempt both the adventurous and the foolhardy to ply these treacherous waters. The result, certainly, will be many shipwrecks, but also some wondrous new systems and devices that were never before imagined.

Our voyage has led us far from the "old world" of the so-called "WIMP" (Windows, Icons, Menus, and Pointer) desktop user interface. This paradigm has been painstakingly designed over more than four decades for multi-modal control with a keyboard and with a mouse, achieved in no small part through the cursor. The cursor is a logical abstraction of the user's locus of attention, known by many other names (originally "the bug," alternately as the *telepointer*, *pointer*, or *tracking symbol*). The abstract representation of the cursor serves as an intermediary that separates the concerns of input and output. Direct interaction does away with this separation and abstraction, leading us towards a world where each device requires specific and particular software design.

One must keep in mind that the WIMP graphical user interface has been highly optimized for the particular pairing of mouse and keyboard input devices. While it is possible to control the pointer and enter text using other input devices, doing so is more difficult and less efficient than with mouse and keyboard. Doing so with a direct-touch interface can even be painful. Many modern platforms rely on the accoutrements of the WIMP interface (buttons, sliders, checkboxes, windows, and so forth), while simultaneously forgetting that these are optimized for a pair of input devices that are not actually present on the systems their platform supports. In short, new modalities require new user interfaces and interaction techniques if one wishes to make the most of their capabilities.

To the extent that we succeed, then, this chapter forms a chart, which despite being but a rough guide that is riddled with many blanks of *terra incognita*, can still assist explorers of this new interactive world. Through this exercise, we seek to equip the explorer with what we believe to be the most valuable tool of all: the confidence in knowing which questions to ask. This will be based on as deep an understanding of the properties of input devices as we are able to impart in this short chapter, and which will be rooted in perhaps the only indisputable fact in the ever-changing landscape of design: *everything is best for something, and worst for something else.*

The following sections of this chapter first consider in a little more depth what is actually meant (in our opinion) by "natural" user interfaces. We discuss terminology, enumerate some common properties of input devices, and provide examples of how these properties apply to familiar examples of devices such as touch screens, pens, and mice. However, our emphasis is on the particular concerns of direct input devices, and less so on indirect devices. We then turn our attention to state-transition models of devices and how these relate to interactive techniques, as well as the hierarchies of fundamental tasks that delineate the units of interaction in a user interface. We briefly examine a number of models and theories that are commonly used to evaluate human performance with interaction techniques and devices. We discuss how to transform input signals for use in applications, and analyze some factors and considerations for the design of appropriate feedback in response to inputs. The chapter then takes a brief sojourn into the specialized realm of discrete symbolic entry, including mobile and keyboard-based text entry. In the closing sections of the chapter we consider some higher-level topics such as input modalities and general strategies for interaction that transcend any particular input device, as well as the impact on input techniques of a diverse and evolving ecosystem of form-factors. We conclude with some thoughts about future trends and opportunities.

UNDERSTANDING INPUT TECHNOLOGIES

Input devices sense physical properties of people, places, or things. Yet any treatment of input devices without regard to the corresponding visual feedback is like trying to use a pencil without paper. This chapter treats input technologies at the level of interaction techniques, which provide a way for users to accomplish tasks by combining input with appropriate feedback. An interaction designer must consider (a) the physical sensor, (b) the feedback presented to the user, (c) the ergonomic and industrial design of the device, and (d) the interplay between all of the interaction techniques supported by a system-- as well as the interplay between a specific device and others in its surrounding digital ecology.

A designer who understands input technologies and the task requirements of users has a better chance of designing interaction techniques that match a user's natural workflow and which take appropriate advantage of users' innate abilities. Making an optimal choice for tasks in isolation often leads to a poor design, so the designer must weigh competing design requirements, as well as transitions between tasks.

Is This Input Method Natural? Why This Is an Ill-Posed Question

The word 'natural' might be applied to interactions with systems. A reasonable working definition for 'natural' is that the experience of using a system matches expectations such that it is always clear to the user how to proceed, and that few steps (with a minimum of physical and cognitive effort) are required to complete common tasks. It is a common mistake to attribute the naturalness of a product to the underlying input technology. A touch-screen, or any other input method for that matter, is not inherently natural.

A common naïve reaction to direct touch and gestural input, for example, is to suggest that it is "more natural" than other input methods because interactions can be accomplished by hand gestures that correspond to movements that users "naturally" make as a result of their everyday real-world experiences. Wobbrock et al. explored this hypothesis (Wobbrock, Morris, and Wilson 2009). They presented user study participants with "before" and "after" images depicting a particular interaction, such as moving an object, duplicating an object, or changing the color of an object. They then asked the participant to perform the gesture they believed would create that change in the system. Three of their experimental tasks are summarized below (**Error! Reference source not found.**).

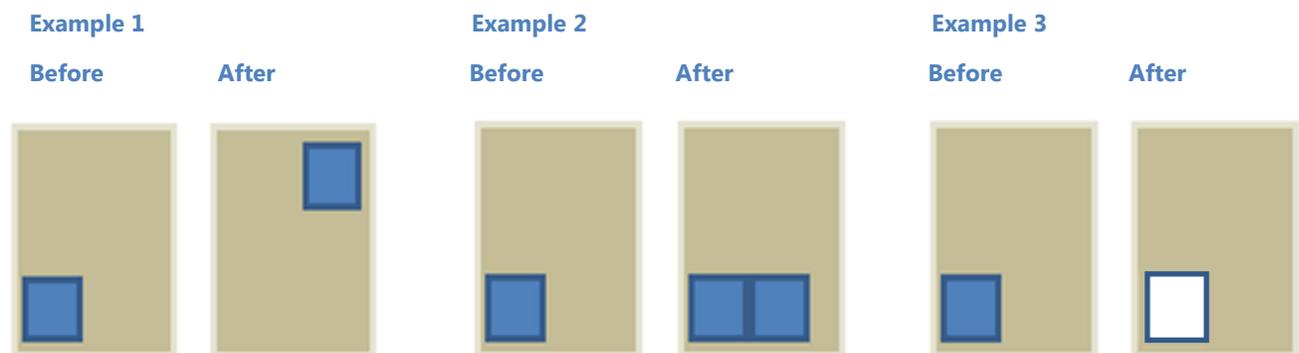


Figure 9.2. *Before-after pairs for three example gestures that a user performs on a hypothetical multi-touch system. What touch gesture would you expect transforms each example from its before state to its after state?*

You can conduct this experiment on yourself. Given a multi-touch screen, what gesture would you expect transforms Example 1 from its *before* state to its *after* state? For Example 1, almost everyone suggests the same action as you likely just envisioned, that of dragging the square with a single finger.

For Example 2, the results are more equivocal. You probably have to think a bit longer before deciding on an appropriate gesture. If we present this example to multiple users, there is some agreement across users, but there are also a number of different gestures that users will choose to "naturally" indicate the duplication of an object.

For Example 3, all bets are off. There is almost no agreement across a set of users as to a natural manipulative gesture to

change the color of an object.

What this exercise shows is that *there is no inherently natural set of gestures* for performing anything beyond a couple of the most commonplace multi-touch manipulations. Looking at the gestures that users suggest for a manipulation can be used as a source of design inspiration, but it does not lead to a set of gestures that is consistent or inherently more natural.

Terminology: Input Device vs. Interaction Technique vs. Conceptual Model vs. User Experience

We take a brief pause here to clarify our terminology. To some extent we can use devices, techniques, models, and experiences as interchangeable terms, but what is more important is for the designer to think about these things at different conceptual levels so as to consider a holistic view of interactive systems.

Input Device. An input device is a transducer that senses physical properties of people, places, or things. What we normally conceive of as an input device (such as a mouse) is often a collection of transducers (Card, Mackinlay, and Robertson 1990, 1991), such as a relative (x,y) motion sensor, physical buttons, and a wheel for scrolling.

Conceptual Model. A conceptual model is a coherent model that users form about the function of a system: what it is, how it works, and how it will respond to their input. Thus the user can determine what input they will likely need to give to the system to achieve a desired result. Conceptual models can be taught, but they are continually refined throughout a user's interaction with a system, given its responses to the user's input. For example, the core conceptual model of the graphical user interface is the point-and-click metaphor of moving the tracking symbol on top of objects on the screen, and then acting on them by clicking or dragging the mouse. Likewise, the desktop metaphor of folders and files is a higher-level conceptual model of how information is stored and retrieved on a computer.

Interaction Technique. An interaction technique is the fusion of input and output, consisting of all hardware and software elements, that provides a way for the user to accomplish a task, given a particular conceptual model. For example, pinch-to-zoom has become a *de facto* standard for touch-screen devices, just as clicking and dragging the mouse as a way to move items is a staple of the desktop metaphor. It is important to remember that the cursor itself is a single element of an interaction technique, and that this is just one of many possible ways to interact using a relative pointing device. Interaction techniques typically vary across input devices, based on the strengths the device's sensing capabilities, its ability to readily incorporate state transitions such as button-presses into the design of the device, and the user's physical abilities and hand comfort when using the device.

User Interface. A user interface is the representation of the system-- the summation all of its input devices, conceptual models, and interaction techniques-- with which the user interacts. It is the responsibility of the user interface to represent and reinforce the user's conceptual model of the system, in concert with the input device and interaction techniques, as well as to present affordances and constraints that make it clear to users how to achieve key tasks. User interfaces, despite being stereotyped as *graphical* user interfaces, also include auditory, tactile, and kinesthetic qualities, even if such secondary feedback results only from the passive mechanical feedback from the physical input devices.

User Experience. The user experience is the broad array of outputs the user perceives and inputs the user gives when interacting with a user interface, as well as the higher level goals, cognitive states, emotions, and social interactions that these experiences support and engender.

Input Device Properties

The variety of pointing devices is bewildering, but a few common properties characterize the important characteristics of most input sensors. These properties help a designer understand a device and anticipate potential problems. We will first consider these device properties in general, and then show how they apply to some common input devices.

Property sensed. Most devices sense linear position, motion, or force; rotary devices sense angle, change in angle, and torque (Buxton 1995; Card, Mackinlay, and Robertson 1991). For example, touch-screens sense the position of one or more fingers, mice sense motion (change in position), and isometric joysticks sense force. The property sensed determines the mapping from input to output, or transfer function, that is most appropriate for the device. Position sensing devices are absolute input devices, whereas motion-sensing devices are relative input devices. A relative device, such as the mouse, requires visual feedback in the form of a cursor to indicate a screen location.

States sensed. Direct input devices, such as touch-screens, touchpads, and pen input devices, are unique in that they not only sense *position*, but also *contact* (i.e., finger-down and finger-up) events that traditional pointing devices lack (Hinckley and Sinclair 1999). These cannot be treated as the equivalent of mouse-click events. This distinction is obvious but its implications can be quite subtle in the design of interaction techniques, even for a designer well versed in such devices. See the section below on Input Device States for further discussion.

Number of dimensions. Devices sense one or more input dimensions. For example, a mouse senses two linear dimensions of motion, a knob senses one angular dimension, and a six degree-of-freedom magnetic tracker measures three position dimensions and three orientation dimensions (Bowman et al. 2004; Hinckley et al. 1994; Zhai 1998). A pair of knobs or a mouse with a scroll wheel sense separate input dimensions and thus form a “1D + 1D” device, or a “2D + 1D” multichannel device, respectively (Zhai, Smith, and Selker 1997).

Device acquisition time. The average time to move one's hand to a device is known as acquisition time. Homing time is the time to return from a device to a “home” position (e.g., return from mouse to keyboard). The effectiveness of a device for pointing tends to dominate acquisition time (Douglas and Mithal 1994). Thus, integration of a pointing device with the keyboard may not improve overall performance, but evaluations still must assess any influence of acquisition times (Dillon, Eday, and Tombaugh 1990; Hinckley et al. 2006) to be certain that a comparison between devices is fair. The elimination of acquisition time is a key advantage of touch interaction for mobile interaction, but of course some homing time is still required bring one's hand into contact with the screen.

Other metrics. System designers must weigh other performance metrics (Card, Mackinlay, and Robertson 1990), including pointing speed and accuracy, error rates, learning time, footprint and gain (Accot and Zhai 2001; MacKenzie 1995; Jellinek and Card 1990), user preference, comfort, and cost. Other important engineering parameters include sampling rate, resolution, accuracy, and linearity (MacKenzie 1995).

DIRECT INPUT DEVICES

A direct input device has a unified input and display surface. A mouse, by contrast, is an indirect input device because the user must move the mouse one surface (a desk) to indicate a point on another (the screen). Direct devices such as touch-screens, or display tablets operated with a pen, are not necessarily easier to use than indirect devices. Direct devices often lack buttons for state transitions. Occlusion is also a major design challenge: the finger or pen covers the area where the user is pointing, so the user may not be able to see all of the options in a pop-up menu, for example. The hand and arm also may occlude visual feedback, dialogs, status indicators, or other controls. Indirect input scales better to large interaction surfaces, since it requires less body movement, and also allows interaction at a distance from the display.

Since direct input represents a major trend in both research and industry, we give a detailed treatment here of a number of design dimensions and considerations for direct devices, with a briefer assay into indirect devices in the section that follows.

In addition to having unified input and display surfaces, with direct input devices the system's feedback for user input is typically localized to the physical point(s) of contact. Some direct input devices can only sense a bare finger. Others such as resistive touchscreens can sense either a plastic stylus or a bare finger, but cannot distinguish one type of contact from the other. Transparent electromagnetic digitizers, such as those found on Tablet PC's, require the use of a special pen, and cannot sense touch unless the system integrates a second touch-sensitive digitizer. Some commercially available digitizers based on capacitive coupling can sense a special pen, and simultaneously differentiate it from multi-touch inputs (Engelhard 2008).

Since there is not yet any "perfect" pen, touch, or combined pen + touch technology, a few of the key questions to ask are:

- Can the digitizer sense touch? If so, is it a "soft-touch" screen, or is a definite contact pressure required to sense touch? If significant pressure is required, it makes touch-based dragging operations more fatiguing and more difficult for the user to perform successfully (without the finger skipping, for example). On the flip side, many "hard-touch" screens can be used while the operator wears gloves, making them well suited to demanding working environments, whereas capacitive touch screens require contact from bare fingers in order to sense touch.
- How many points of contact can the digitizer sense? Many current devices are limited to 2-4 touch points, which limits the multi-touch or hand palm contact signals that one can reliably sense. In some cases, two-point devices have further limitations (such as line-of-sight issues with optical sensing techniques, or only reporting the bounding box of the touch points, rather than the actual touch points, with some touch screen technologies). On some devices, large contact area touches, such as from a thumb or palm, may be reported as two or more points of contact, which introduces further complexities in the software and interaction design. Likewise, closely spaced contacts may merge into a single contact on many touchscreens, producing further potential difficulties and technical limitations. Such complexities are often the outcome of sophisticated filtering techniques, firmware, and drivers, rather than the hardware per se, but nonetheless pervade commercially available touch devices.
- Can the touch contact area or pen pressure be sensed and reported to applications, or is each contact reduced to a single (x,y) point? If contact area is sensed, can applications access the actual image or contour of the contact, or is it only reported as a bounding ellipse (or other simplified representation) of the contact?
- If a stylus is supported at all, is a special pen required, or can contact from any hard object be sensed? If a special stylus is required, keep in mind that it is not a matter of *if*, but rather *when*, the user loses the stylus. Most special pens on the market are powered by inductive coupling, but some are active pens that require batteries.
- Can pen contacts be distinguished from touch contacts? Capacitive styli are available for many touch-screens, but typically these inputs cannot be differentiated from ordinary finger contacts.
- Can pen contacts be sensed while one is also touching the device? Can touch still be sensed while the pen remains in proximity of the screen? Many current devices stop reporting touch when the pen is in range of the screen, so as to reduce false inputs triggered by palm contact. Also, note that most digitizers can only sense a single pen at a time, and typically one pen cannot be distinguished from another (although some high-end Wacom display tablets have a "PenID" feature that can distinguish various types of pens, in order to support facile swapping between custom brush settings, for example).

Standard data-sheets for digitizers often leave these questions unanswered, and it may be unclear what brand or type of digitizer an integrated product uses, making it difficult to ascertain exactly what a particular device offers. Buyer beware, and ask a lot of questions. On the flip side, if you are designing an application without direct knowledge of the underlying hardware that a user will actually be using to experience your software, then you must somehow choose a design path that takes into account all of these possible hardware manifestations. This is akin to successfully guiding a single thread through a gauntlet of needles in the dark. Welcome to the breadth/depth dichotomy.

We stated above that every input modality is best for something and worst for something else. Ultimately it is the designer's job to know what to use when, for whom, for what, and why. From a technology standpoint much of this turns on a nuanced understanding of the properties of an input modality. To offer insight into the main issues for commonplace devices, the following tableau summarizes interaction properties shared by pen and touch (*Table 1*). We do not characterize these properties as "pros" and "cons" to accentuate our belief that almost any property of a device, when used appropriately, can be advantageous in a design. A number of recent systems that explore combined pen and touch input epitomize this approach to design (Hinckley, Pahud, and Buxton 2010; Hinckley et al. 2010; Zeleznik, Bragdon, Ko, et al. 2010; Brandl et al. 2008; Brandl et al. 2009; Frisch, Heydekorn, and Dachsel 2009). Beyond pen and touch, we believe this is a valuable perspective for a broad class of input devices, modalities, and form-factors, as exemplified by other research into multimodal input (Bolt 1980; Cohen et al. 1997; Tse et al. 2008; Levine and Ehrlich 1991).

PROPERTY	PEN	TOUCH
Contacts	1 point A single well-defined point.	1-10+ contact regions Often with shape information (Cao et al. 2008).
Occlusion	Small (pen tip) But hand still occludes screen.	Moderate ("fat finger") to Large (pinch, palm, whole hand gestures)
Precision	High Tripod grip / lever arm affords precision, writing & sketching tasks.	Moderate Nominal target size for rapid acquisition via touch is about 10-18 mm ² (Vogel and Baudisch 2007) (Sears 1993) (Lewis, Potosnak, and Magyar 1997)
Hand	Preferred hand	Either hand / Both hands
Elementary Inputs	Tap, Drag, Draw Path	Tap, Hold, Drag Finger, Pinch
Intermediary	Mechanical Intermediary Takes time to unsheathe the pen. Pen can be forgotten.	None: Bare-Handed Input Nothing to unsheathe, nothing to lose. No lever arm.
Acquisition Time	High (first use: unsheathe the pen) Moderate on subsequent uses: pen tucked between fingers.	Low No mechanical intermediary to acquire.
Buttons	Barrel Button, Eraser (<i>some pens</i>)	None
Activation Force	Non-Zero Tip switch or minimum pressure.	Zero (capacitive touch). Note that resistive touch requires some force.
False Inputs	Palm Rejection: Palm triggers accidental inputs, fingers drag on screen while writing, etc. This is a difficult problem. Designs must accommodate incidental palm contact when it inevitably occurs.	"Midas Touch Problem" Fingers brush screen, finger accidentally rests on screen while holding device, etc. "Chess Player's Syndrome" Device senses touch when none occurred. Common problem on optical touch-screens.

Table 1. Comparison of several key properties shared by touch and pen input devices (Hinckley, Pahud, and Buxton 2010; Hinckley et al. 2010).

This limited survey shows that pen and touch, while sharing common ground as direct input modalities, also exhibit many important differences, and these again differ substantially from the properties of indirect pointing devices such as the mouse. However, also keep in mind that this tableau is just a summary of some common properties of these devices, and the properties of specific pen or touch devices can vary widely, as discussed further below.

Single Touch vs. Multiple Touch. Devices capable of detecting touch are often classified colloquially as either 'touch' or 'multi-touch' devices, but there is a more subtle distinction. Single-touch devices, such as traditional resistive touch screens, are adequate for emulating a mouse, and for detecting most of the gestures employed in commercial 'multi-touch' software today (Potter, Weldon, and Shneiderman 1988). Multi-touch devices can be further classified by the number of finger contacts they are able to detect and track. A multiple contacts are required for a user to perform true multi-touch gestures, such as pinch-to-zoom or multi-fingered grabbing (Krueger, Gionfriddo, and Hinrichsen 1985; Moscovich and Hughes 2006). Still more contacts must be tracked to enable multiple users to preform multi-touch gestures at once, as desired for table-top interfaces, for example.

Many devices reputed to be "multi-touch" in fact can only sense limited information about the touch contacts, such as the bounding box (rather than the actual x,y coordinates) of the touches. This class of devices includes the DiamondTouch Table (Dietz and Leigh 2001) and some optical touch sensing technologies. These are sometimes referred to as "1.5D" devices, since they do not support two fully independent degrees of freedom for each touch point. Nonetheless, such devices can implement a wide array of "multi-touch" interaction techniques (Wu and Balakrishnan 2003; Forlines and Shen 2005).

Pressure and Contact Area Sensing. Pressure is the measure of force that the user exerts on an input device. Pressure sensing is often confused with contact area sensing. True pressure sensing is supported by many pen-operated devices, but typically only contact area can be sensed by touch-screen interfaces. The two are related, and although contact area is a useful proxy of pressure, it cannot be treated as a true equivalent of pressure for the purposes of interface and interaction technique design. For example, a woman with long fingernails who touches a screen will produce a touch with a large contact area, even though the actual force applied may be very light. With contact area, rather than relying on an absolute degree of contact, one should instead emphasize *changes in contact area* as a more controllable parameter that users can modulate, such as by rolling one's finger in contact with the display (Benko, Wilson, and Baudisch 2006). Contact area has been demonstrated to add states to touch devices (Forlines and Shen 2005), allowing a type of hover state when touching lightly, and then committing to a particular action when the user presses more forcefully. Used as a continuous parameter, contact area can enable an continuous control of the thickness of a paint stroke, or set the z-order of onscreen objects (Davidson and Han 2008), for example.

However, some touch devices do offer true pressure sensing capability. For example, some laptop touchpads include sensors that measure the force exerted on the pad. In multi-touch systems, such approaches typically provide only an aggregate of forces across all contact areas, rather than an independent measure of pressure for each touch.

Users are able to perceive up to 7 distinct levels of pressure, provided that there is suitable visual feedback (Ramos, Boulos, and Balakrishnan 2004). Pressure can then be used for a variety of interaction techniques such as mode-switching (Agarawala and Balakrishnan 2006), or continuous adjustment of parameters while drawing a pen stroke. Pressure can then be used for a variety of interaction techniques such as mode-switching (Agarawala and Balakrishnan 2006), or continuous adjustment of parameters while drawing a pen stroke (Ramos and Balakrishnan 2005; Ramos, Boulos, and Balakrishnan 2004; Ramos and Balakrishnan 2006).

Hand Postures & Shape-Based Input. Devices that can sense not only points of contact but also the shape of the contact region can allow for richer input (Krueger 1991; Cao et al. 2008). Researchers have also used the sensed contact region to compute the user's perceived 'point' of contact for high-precision touch interaction. For example, Holz & Baudisch (Holz and Baudisch 2010) use a fingerprint sensor to determine the finger's contact, orientation, and position, and map these parameters to more precisely determine the user's intended touch location. Subtler uses of shape-based input have demonstrated the advantage of using a geometric representation of the contact area, instead of the traditional reduction of all touch information to a single point, such as to enable rapid interaction with multiple widgets (Moscovich 2009), or to support greater expressiveness from each touch (Cao et al. 2008; Zeleznik, Bragdon, Adeptura, et al. 2010). A challenge for the effective and practical use of shape-based touch interaction is the lack of a commonly accepted input event and interaction model for software development and user interface design. It remains difficult to develop practical multi-touch applications that wander too far from the path followed by the traditional point-based hit testing model of interaction. Probabilistic approaches offer one potential solution to this dilemma (Schwarz et al. 2010).

In-Air Hand Postures and "Direct" Sensing Beyond the Display. Recent approaches have also demonstrated the utility of differentiating in-air hand postures, as opposed to those that occur while in contact with a device (Grossman, Wigdor, and Balakrishnan 2004; Hilliges et al. 2009; Wilson and Benko 2010). For example, the SecondLight system can see through the display, and can both project light and sense interactions in the volume above the display itself. Using this system, Hilliges used cameras to detect simple postures both in the air and when in contact with the display, using the moment of contact as a state transition to delimit interactions (Hilliges et al. 2009). This enables both the user and the system to agree on, and differentiate between, hand gestures that occur in contact with the display, as opposed to incidental movements of hands in the air. Thus,

detecting and differentiating system response based on physical contact with the screen delivers an experience with discrete and controllable states, even though the device perceives similar hand movements in either case. Wilson et al further explore the possibilities of this space using multiple depth cameras and projectors (Wilson and Benko 2010).

Furthermore, this demonstrates once again how direct input is about much more than touch alone: the positions of users relative to the display (Ballendat, Marquardt, and Greenberg 2010), the motions of their hands above the display (Tang et al. 2010), and the posture of the hand as it comes into contact with the displays (Holz and Baudisch 2010) are all also forms of direct input that can extend and enrich user interfaces. What is critical to note about these examples is that they do not attempt to use in-air gesturing to replace the mouse, simulate text entry on a virtual keyboard, or replace established touch gestures. Rather they leverage the distinct properties of spatial sensing to provide new capabilities that are well differentiated from direct touch and other inputs.

Finger Differentiation, User Differentiation, and User Identification. Some touch devices are able to determine which of the user's fingers are in contact with the display. For example, the Microsoft Surface images enough of the hand in proximity to the display that it is often possible to determine which fingers of a hand are in contact with the display (Lepinski, Grossman, and Fitzmaurice 2010). Some systems can identify the source of contacts, such as through capacitive coupling techniques (Dietz and Leigh 2001), so that a touch from one user can be distinguished from the touch of another user. User identification is a stronger form of differentiation that persists the distinction between multiple users across sessions. One way to achieve this is through the sensing and identification of the user's fingerprints while touching a display (Holz and Baudisch 2010), although current demonstration systems cannot achieve such sensing in real time. Computer vision technologies have also been demonstrated which can identify users, or distinguish which fingers of the user's hand are touching the display, based on visual features.

Parallax. Parallax error is a mismatch between the sensed input position and the apparent input position due to viewing angle. *Display parallax* is the displacement between the sensing and display surfaces. Of course, zero parallax is ideal, but a rule of thumb used by industry practitioners is that if the display parallax is less than about 2mm, then its practical impact is not that significant. *Transducer parallax* results from any additional parallax error that may result from the offset between the tip of a mechanical intermediary and the actual component that is sensed, such as the coil that sits higher up in the body of Wacom electromagnetic pens. If the tilt angles of the pen are known, this offset can be corrected.

Latency. Latency is a problem for all interactive systems, but its effects are particularly insidious for direct input devices because any discrepancy between the actual position of the fingers or the pen, versus the position currently sensed by the system, becomes immediately obvious to the user. Latency is the end-to-end measure of the time elapsed between the moment a physical action is performed by the user, versus the moment the system responds to it with feedback that the user can perceive. This round-trip time between cause and effect comes from many sources; it is difficult to minimize and impossible to eliminate in practical systems design. Sources of latency include the hardware sampling rate; the time it takes to report samples to the operating system as well as report events to applications; the time required by software to compute the results; the time to refresh the frame buffer; and the physical screen's update rate to make those results visible to the user. Seow offers a detailed treatment of the effects of latency on user perception (Seow 2008). In experiments, latency typically exhibits strong negative effects on user performance starting at about 100ms (Card, Mackinlay, and Robertson 1991; MacKenzie and Ware 1993). In modern direct-input systems with pen or touch input, latencies of far less than 100ms must be achieved (Card, Mackinlay, and Robertson 1991; MacKenzie and Ware 1993) if one wishes users to perceive the interface as crisp and responsive.

INDIRECT INPUT DEVICES

An indirect input device is one which does not provide input in the same physical space as the output. Indirect devices eliminate occlusion of the screen by the user's hand and fingers. However, typically they require more explicit feedback and representation of the input device (such as a cursor), the intended target on the screen (such as highlighting icons when the cursor hovers over them), and the current state of the device (such as whether a button is held or not).

Virtual devices. Most operating systems treat all input devices as virtual devices, which tempts one to believe that input devices are interchangeable; however, the details of what the input device senses, how it is held, the presence or absence of buttons, and many other properties, can significantly impact the interaction techniques—and hence, the end-user tasks—that a device can effectively support. Although the virtual devices abstraction has been a successful strategy to address the breadth/depth dichotomy for indirect devices-- largely because the tracking symbol serves as an intermediary between the type of device and the interaction techniques that it supports-- it is becoming increasingly clear that the virtual devices abstraction is far less satisfactory when extended to direct input devices.

Indirect Digitizing Tablets and Absolute vs. Relative input. An absolute input device senses the position of an input and passes this along to the operating system. Relative devices sense only changes in position. Digitizing tablets can operate in absolute mode, with a fixed control-to-display (C:D) gain between the tablet surface and the display, or in relative mode, in which the tablet responds only to motion of the stylus or finger. If the user touches the stylus or finger to the tablet in relative mode, the cursor resumes motion from its previous position; in absolute mode, it jumps to the new position. Absolute mode is generally preferable for tasks such as drawing, handwriting, tracing, or digitizing, but relative mode may be preferable for traditional desktop graphical user interaction tasks such as selecting icons or navigating through menus. Digitizing tablets thus allow coverage of many tasks (Buxton, Hill, and Rowley 1985), whereas mice can only operate in relative mode.

Mixing Indirect & Direct Input. It is possible to emulate the properties of an indirect input device using a direct one, simply by offsetting the device's apparent input target from its physical location. This leads to an interesting class of hybrid techniques that mix direct and indirect representations within the user experience of a direct input device (Buxton, Hill, and Rowley 1985; Brown, Buxton, and Murtagh 1990; Sears and Shneiderman 1991; McCallum and Irani 2009). For example, HybridPointing provides a cursor that remains directly beneath an input stylus, yet can also decouple to allow pointing at distant targets on very large displays without great physical movement (Forlines, Vogel, and Balakrishnan 2006). Such techniques also support high precision pointing on touchscreens (Vogel and Baudisch 2007).

Brief Tour of Indirect Input Devices

The following tour discusses important properties of several common indirect input devices, such as mice, joysticks, touchpads, trackballs, and stand-alone touch-tablets and pen digitizer tablets.

Touchpads. Touchpads are small, touch-sensitive tablets often found on laptop computers. Touchpads use relative mode for cursor control because they are too small to map to an entire screen, but most touchpads also have an absolute mode to allow interactions such as character entry, or sliding along the edge of the pad to scroll. The small size of touchpads necessitates frequent clutching, and touchpads can be awkward to use while holding down a button, unless the user employs his or her other hand. Traditionally, touchpads have supported "clicking" by recognizing tapping or double-tapping gestures, but accidental contact (or loss of contact) can erroneously trigger such gestures (MacKenzie and Oniszczak 1998). To reduce these types of problems, some modern touchpads often include a microswitch underneath the pad, so that pressing down the pad produces a mechanical click.

Multi-Touch Pads. Modern touchpads are increasingly becoming multi-touch devices, as opposed to the single-touch models commonplace in the past. The multi-touch interaction model for indirect touchpads typically differs from the multi-touch interaction model for direct-touch displays. If both devices are "natural" multi-touch devices, then why are two different interaction models necessary? The indirect touchpad must support relative cursor control, and typically single touch is mapped

to moving the cursor. Thus two fingers are required to scroll or pan documents. With a direct-touch input device, cursor tracking is not necessary, and hence single touch can pan and scroll. This underscores why characterizing a device simply as a "touch" device is not sufficient to understand its properties. Furthermore, it further underscores why assuming that touch inherently produces a natural interface is a fallacy. Finally, this demonstrates why an "inconsistency" between two related devices-- here an indirect vs. a direct touch device-- is often necessary to produce a user experience that feels logically consistent to the user, given the device and form-factor at hand.

Indirect multi-touch pads can also support additional novel techniques, such as targeting many degrees of freedom to a single cursor position, or representing each point of contact with a separate cursor (Moscovich and Hughes 2006). Such models have also been explored in the context of mice augmented with multi-touch input surfaces (Villar et al. 2009; Benko et al. 2010). An alternative to cursors that can be useful on both direct and indirect devices is to present a "shadow" of the whole hand (Krueger 1991; Wigdor, Forlines, et al. 2007; Tang et al. 2010).

Mice. Douglas Englebart and colleagues (English, Englebart, and Berman 1967) invented the mouse at the Stanford Research Institute. More than forty years later, the mouse persists because its properties provide a good match between human performance and the demands of graphical interfaces (Balakrishnan et al. 1997). For typical pointing tasks on a desktop computer, one can point with the mouse about as well as with the hand itself (Card, English, and Burr 1978). Because the mouse stays put when the user releases it (unlike a stylus, for example), it is quick for users to reacquire and allows designers to integrate multiple buttons or other controls on its surface. Users exert force on mouse buttons in a direction orthogonal to the mouse's plane of motion, thus minimizing inadvertent motion. Finally, with mice, all of the muscle groups of the hand, wrist, arm, and shoulder contribute to pointing, allowing high performance for both rapid, coarse movements as well as slow, precise movements (Guiard 1987; Zhai, Buxton, and Milgram 1996). These advantages suggest the mouse is hard to beat; it will remain the pointing device of choice for desktop graphical interfaces.

Trackballs. A trackball senses the relative motion of a partially exposed ball in two degrees of freedom. Trackballs have a small working space (footprint), afford use on an angled surface, and sometimes are weighted to afford spinning the ball with physical inertia. Trackballs may require frequent clutching movements because users must lift and reposition their hand after rolling the ball a short distance. The buttons are located to the side of the ball, which can make them awkward to hold while rolling the ball (MacKenzie, Sellen, and Buxton 1991). A trackball engages different muscle groups than a mouse, offering an alternative for users who experience discomfort when using a mouse. Trackballs are also well-suited to nonpreferred-hand input in combination with a mouse (Bier et al. 1993).

Isometric joysticks. An isometric joystick (e.g., the IBM Trackpoint) is a force-sensing joystick that returns to center when released. Most isometric joysticks are stiff, offering little feedback of the joystick's displacement. The rate of cursor movement is proportional to the force exerted on the stick; as a result, users must practice in order to achieve good cursor control. Isometric joysticks are particularly appealing to consider when space is at a premium (Douglas and Mithal 1994; Rutledge and Selker 1990; Zhai, Smith, and Selker 1997).

Isotonic joysticks. Isotonic joysticks sense angle of deflection. Some hybrid designs blur the distinctions of isometric and isotonic joysticks, but the main questions that characterize the design space are the following:

- Does the joystick sense force or angular deflection?
- Does the stick return to center when released?, and
- Does the stick move from the starting position?

For a more thorough discussion of the complex design space of joysticks, we recommend (Lipscomb and Pique 1993).

Summary & Perspective on Indirect Input Devices

While many of these indirect devices may feel antiquated in the context of modern system design trends, such devices still have their uses and can offer important lessons to the student of interaction. Furthermore, consoles or work-stations for dedicated and highly specialized tasks-- such as driving a car-- often still incorporate many creative combinations of such devices in conjunction with physical buttons, switches, and dials. To anyone who has attempted the dangerous exercise of operating a touch-screen navigation system while driving, it should come as no surprise that rich spatial arrangements of physical input devices with complementary and specialized roles offers a highly effectively solution while placing fewer demands on the operator's visual attention (e.g., (Fitzmaurice, Ishii, and Buxton 1995). Hence, combinations of indirect input

devices are ideally suited to many applications, and remain important to understand for interaction designers worthy of the name.

INPUT DEVICE STATES

Any device that returns a coordinate can be considered a pointing device, yet there remains a critical missing ingredient: that of the *events* that a device generates, such as when a finger comes into contact with a screen, a button is depressed on a mouse, or a pen leaves the proximity of a digitizer. These events trigger *state transitions* which in turn are the building blocks of interaction techniques. All of this seems blatantly obvious until one realizes that there are subtle discrepancies between the events and properties sensed by various devices while in different states. This innocent façade of greenery conceals a nightmarish thicket that can entangle the designer in stinging nettles and poisonous ivies of failed or severely compromised interaction techniques.

There is a fundamental mismatch between the demands of traditional graphical interfaces and the states and events that can be sensed by devices such as touchscreens, touchpads, and pen-operated devices, which makes it difficult to support the full set of desktop graphical user interface primitives, including click, drag, double-click, and right-click, on such devices. There is no easy solution that does not involve design compromises. When considering such devices, to make device limitations and differences concrete, diagramming all of these states and transitions can be an enlightening exercise.

Input devices taken in general support three possible states (*Table 2*): (a) out-of-range, (b) tracking, and (c) dragging; practitioners refer to these as State 0, State 1, and State 2, respectively, of the three-state model (Buxton, 1990). This model is useful for reasoning about the relationship between the events sensed by an input device and the demands of interaction techniques.

State	Description
0	<i>Out Of Range</i> : The device is not in its physical tracking range.
1	<i>Tracking</i> : Device motion moves only the cursor.
2	<i>Dragging</i> : Device motion moves objects on the screen.

Table 2. Comparison of several key properties shared by touch and pen input devices (Hinckley, Pahud, and Buxton 2010; Hinckley et al. 2010).

The three-state model describes the mouse as a two-state device, supporting State 1, the cursor tracking state, as well as State 2, the dragging state (**Error! Reference source not found.**). State 1 provides cursor feedback of the screen position that the device will act upon, while State 2 allows the user to drag an object by holding down the primary mouse button while moving the device. The mouse senses movement in both the tracking and dragging states, as represented by the (dx, dy) in each state (Fig. 9.2, left), which is shorthand to indicate relative motion tracking capability (Hinckley, Czerwinski, and Sinclair 1998).

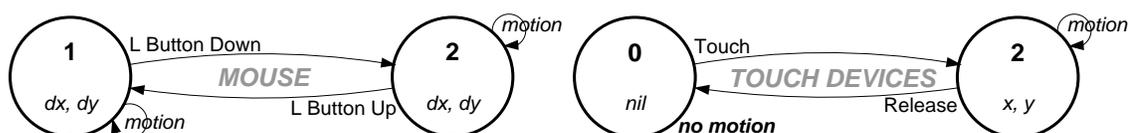


Figure 9.3. *State transition diagram for standard mouse and touch input devices. Each device senses two states, but critically, not the same two states.*

Many touch-activated devices, such as touch-screens and touchpads, are also two-state devices, but *they do not sense the same two states as the mouse* (**Error! Reference source not found.**, right). For example, a mobile phone touch-screen can sense a finger when it is in contact with the display; this is the equivalent of the mouse dragging state (State 2). The touch-screen can also sense when the finger is removed from the screen, but once the finger breaks contact, this enters State 0 (out-of-range), where no motion can be detected (emphasized by the annotation of *nil* in state 0 of **Error! Reference**

source not found., right). Thus, although the mouse and touch-screen both sense two states, the lack of a second motion-sensing state on the touch-screen means that it will be difficult to support the same interaction techniques as a mouse. For example, should sliding one's finger on the device move a cursor, or drag an object? The designer must choose one; a touch-screen or touchpad device cannot support both behaviors at the same time.

If we add pens for tablet computers to the mix, we now have a third state: many pens can sense when they enter or leave the proximity of the screen (**Error! Reference source not found.**). But despite the addition of the third state, the input events sensed by the pen still differ from those of the mouse, as well as those of touch. Hence all three devices are mutually incompatible in subtle ways, and interactive techniques that attempt to support all these devices at the same time will likely encounter various inconsistencies and usability problems as a result.

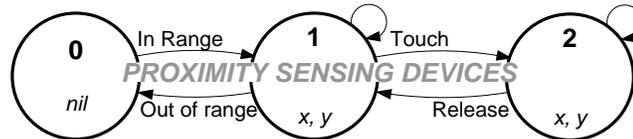


Figure 9.4. State transition diagram for proximity sensing devices, such as a pen on a Tablet PC.

Pen and touch devices often lack a button suitable for right-click, leading to heavy use of awkward interface band-aids such as touch-and-hold to call up context menus. Even if a pen does include a barrel button, it is relatively slow to access as well as prone to inadvertent activation (Li et al. 2005). The press-and-hold solution has become standard on mobile devices, but the timeout introduces an unavoidable delay. For rapid activation, the time-out should be short as possible, but to avoid inadvertent activation, paradoxically the timeout also must be as long as possible. A 500-millisecond timeout offers a reasonable compromise, but most commercial devices assume that context menu invocation occurs only infrequently, so they use delays of about 1000 milliseconds to further tip the design balance towards reducing accidental activations. Techniques designed for pen-operated devices (Apitz et al. 2004); (Kurtenbach and Buxton 1991; Kurtenbach and Buxton 1991; Moran, Chiu, and Melle 1997) should afford rapid and unambiguous activation of one of several possible actions as fundamental building blocks (Hinckley, Baudisch, et al. 2005) to avoid inefficient or highly modal interactions (Hinckley et al. 2006).

Similar issues plague other interaction modalities, such as (a) motion-sensing mobile devices (Hinckley, Pierce, et al. 2005; Hinckley and Song 2010), (b) camera-based tracking of the hands (Wilson and Oliver 2003), and (c) spatial input devices (Hinckley et al. 1994). All of these techniques require a method for users to move the device or their hands without accidentally performing an action, which speaks to the appeal of hybrid solutions such as the combination of spatial sensing with direct-touch input (Wilson and Benko 2010). State transitions thus form fundamental indications of intent that are essential to the construction of rapid, dependable, and habit-forming interaction techniques.

WHAT'S AN INPUT DEVICE FOR? THE COMPOSITION OF USER TASKS

One way of reasoning about input devices and interaction techniques is to view a device or technique in light of the tasks that it can express. But what sort of tasks are there?

Elemental Tasks

While computers can support many activities, at the input level, some subtasks appear repeatedly in graphical user interfaces, such as pointing at a target on the screen or typing a character. Foley and colleagues (Foley, Wallace, and Chan 1984) identified elemental tasks including (a) text (entering symbolic data), (b) select (indicating objects from a set of alternatives), (c) position (pointing to a screen coordinate), and (d) quantify (specifying an exact numeric value). If these are elemental tasks, however, then where do devices such as global positioning system (GPS) readers, cameras, or fingerprint scanners fit in? These offer new “elemental” data types of location, images, and identity, respectively. Elemental tasks are difficult to fully enumerate, because advances in technology will continue to yield data types that enable new tasks and scenarios of use. Furthermore the perspective of what tasks are “elemental” depends in part on the input devices, as well as the interaction design, through which an interface expresses them.

Compound Tasks and Chunking

A fundamental problem with the elemental task approach is that the level of analysis for “elemental” tasks is not well defined. For example, a mouse indicates an (x, y) position on the screen, but an Etch A Sketch[®] separates positioning into two subtasks by providing a single knob for x and a single knob for y (Buxton 1986). If position is an elemental task, why must we subdivide this task for some devices but not others? One way to resolve this puzzle is to view all tasks as hierarchies of subtasks (**Error! Reference source not found.**). Whether or not a task is “elemental” depends on the input device being used: the Etch A Sketch supports separate *QuantifyX* and *QuantifyY* tasks, whereas the mouse supports a compound *2D Position* task (Buxton 1986). Now, if a user wishes to indicate a point on the screen, the integrated (x,y) position of the mouse offers the best solution. But if the user wishes to precisely specify an x coordinate independent of the y coordinate, then the Etch A Sketch[®] more directly expresses this concept.

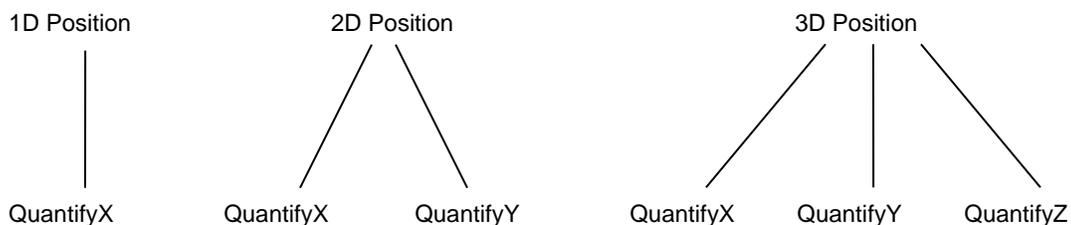


Figure 9.5. *User tasks can be conceived as unitary actions, or as assemblies of subtasks.*

From the user’s perspective, a series of elemental tasks may seem like a single task. For example, scrolling a web page to click on a link could be conceived as an elemental 1D positioning task followed by a 2D selection task, or, it can be viewed as a compound navigation/selection task (Buxton and Myers 1986). An interaction technique can encourage the user to work at the higher level of the compound task, for example, by scrolling with one hand while pointing to the link with the other hand. This is known as chunking.

These examples show that the choice of device influences the level at which the user is required to think about the individual actions that must be performed to achieve a goal. The design of input devices and interaction techniques can help to structure the interface such that there is a more direct match between the user’s tasks and the low-level syntax of the individual actions that must be performed to achieve those tasks. The choice of device and technique thus directly influences the steps required of the user and hence the apparent complexity of an interface design (Buxton 1986).

Phrasing

Most of the above examples of chunking revolved around integration of multiple degrees of freedom to support higher-level tasks. Another implement in the tool-box of the interface designer to afford the design of compound tasks is *phrasing*, which is the use of muscular tension to maintain a temporary state that "glues together" multiple sub-tasks (Buxton 1995). A good example is that of a pull-down menu, where holding down the mouse button integrates that tasks of activating the menu, sliding the pointer to the desired menu item, and then lifting the button to select the menu item. The muscular tension from maintaining contact with a touch-screen, or from holding down a button on a mouse, provides continuous and salient proprioceptive feedback to the user that the system is in a temporary state, or mode, where movement of the input device navigates the menu, instead of moving the cursor. Another key property of phrasing is that closure is inherent in the means used to introduce the phrase: simply releasing the mouse button returns the system to its default state. Obviously, this approach has its limits, as the user cannot maintain muscular tension for very long without discomfort and fatigue, but it has been shown to reduce mode errors for frequent temporary mode switches (Sellen, Kurtenbach, and Buxton 1992). While the preceding example of the pull-down menu requires a mouse button, phrasing can also be achieved on direct input devices by maintaining finger contact with a touch-screen (REF Hinckley UIST 2010), or by holding a pen against a digitizer while articulating a compound gesture (Kurtenbach and Buxton 1991; Apitz et al. 2004).

EVALUATION AND ANALYSIS OF INPUT DEVICES

Beyond standard usability engineering techniques, there are several techniques tailored to the study of input devices and interaction techniques. Representative tasks (Buxton 1995), such as target acquisition, pursuit tracking, freehand drawing, and dragging versus tracking performance (MacKenzie, Sellen, and Buxton 1991), can be used to formally or informally evaluate devices. Here, we focus on formal analysis using Fitts' Law, the Steering Law, and the Keystroke-Level Model, but we begin with a high-level perspective on how to honestly evaluate interaction techniques in general.

Evaluating the True Cost of a Technique in the Context of Surrounding Operations

When evaluating an interaction technique, one must fully consider the true cost to invoke a command (Dillon, Eday, and Tombaugh 1990), as well as the context of surrounding actions that are necessary to set up and recover from a particular action (Appert, Beaudouin-Lafon, and Mackay 2004; Hinckley et al. 2006). Commands that act on a selection are one example: the interaction technique used to select objects can influence how effectively one can act on the selection (and vice versa). Indeed, if one views selection-action phrases as a compound interaction technique, rather than studying selection in isolation from command activation, new approaches that optimize the compound task may become possible (Kurtenbach and Buxton 1991; Hinckley, Baudisch, et al. 2005; Apitz et al. 2004). A discussion of which techniques are most appropriate for a given task can be found in (Mackay 2002).

As another "toy" example, let's say that a user's task is to create a single square. What is the best interaction technique to perform this task? Tapping on a mechanical button is very fast. Perhaps the user could tap a button on a pointing device to create a square centered on the current cursor location. But, what if the user also must be able to produce circles, triangles, and polygons? Creating a square just from a button tap will no longer work because one needs a way to specify which type of shape to create. Perhaps, then, depressing the button to call up a menu where one can stroke in a particular direction to create the desired shape will work well (Kurtenbach and Buxton 1991). What this toy example makes clear is that one cannot optimize the *create square* interaction in isolation. One must also consider the set of surrounding interactions and tasks that together comprise the user interface.

We must bring this same perspective when we evaluate interaction techniques. To extend the example above a bit further, what if the user also wishes to be able to make notations in freehand with the same input device? One then has to manage the *mode* of the device, with a *command mode* to create the shapes, and an *inking mode* to make freehand strokes. The cost of invoking the *create square* command now depends on the total time that it takes to enter the mode, articulate the *create square* command, and then return back to the original mode. We cannot just measure the time it takes to draw the *create square* gesture. Even costs such as repositioning the pointer to the working area of the user interface, or the user's mental preparation time to plan the next step, must be considered. We cannot honestly evaluate the *true cost* of articulating the command without

a full and diligent account of its entire life cycle.

Furthermore, the context of surrounding operations can influence the efficacy of a technique as well. What if the user's task were to create many squares in a row? What if it were to interleave creating single squares with drawing freehand marks? Our view of which technique is most effective in a given context may depend on such usage patterns (Hinckley et al. 2006; Appert, Beaudouin-Lafon, and Mackay 2004; Guimbretiere, Martin, and Winograd 2005).

Fitts' Law

Fitts' Law has been so widely applied to the design and evaluation of input devices that mere mention of it is taken as a dirty word in some quarters. Nonetheless it is a highly sensitive tool for evaluating input devices and techniques when used appropriately.

Fitts' Law is an experimental paradigm (Fitts 1954) that has been widely applied to the comparison and optimization of pointing devices. Fitts' Law is used to measure how effectively a pointing device can acquire targets on the screen. Fitts' Law was first applied to the study of input devices by Card et al. (Card, English, and Burr 1978); it is now a standard for device comparisons (Douglas, Kirkpatrick, and MacKenzie 1999). Fitts' Law can be applied to remarkably diverse task conditions, including (a) rate-controlled devices (MacKenzie 1992), (b) area cursors (Kabbash and Buxton 1995), (c) scrolling (Hinckley et al. 2002), and (d) zooming (Guiard et al. 2001). For further guidance on conducting Fitts' Law studies, see Douglas and colleagues (Douglas, Kirkpatrick, and MacKenzie 1999; MacKenzie 1992), and (Raskin 2000).

The standard Fitts' task paradigm measures the movement time MT between two targets separated by amplitude A , with a width W of error tolerance (**Error! Reference source not found.**).

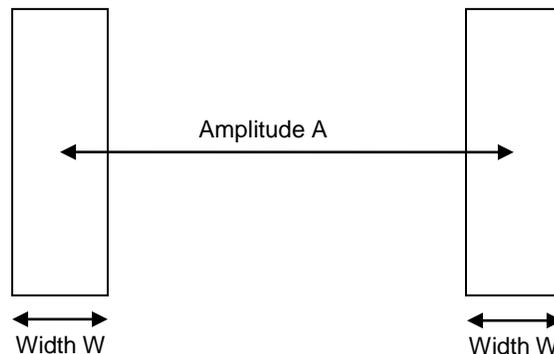


Figure 9.6. *Canonical Fitts' task, with a pointing amplitude A between targets of width W .*

Fitts' Law states that a logarithmic function of the ratio of A to W predicts the average movement time MT . The Fitts' Law formulation typically used for input device studies is:

$$MT = a / b \log_2(A/W + 1) \quad (9.1)$$

Here, the constants a and b are coefficients fit to the average of all observed MT for each combination of A and W in the experiment. One calculates a and b via linear regression using a statistical package or spreadsheet. The constants a and b depend heavily on the exact task setting and input device, so be wary of substituting "typical" values for these constants, or of comparing constants derived from different studies.

Psychomotor interpretations for Fitts' Law have been proposed (Douglas and Mithal 1997). However, since the law does not characterize individual pointing movements, but rather the central tendency of a large number of pointing movements, the law may simply reflect information-theoretic entropy (MacKenzie 1989).

The Steering Law and Minimum Jerk Law

Steering a cursor through a narrow tunnel, as required to navigate a pull-down menu, is not a Fitts' task because steering requires a continuous accuracy constraint: the cursor must stay within the tunnel at all times. For a straight-line tunnel (**Error! Reference source not found.**) of width W and length A , for example, the Steering Law models movement time as a linear

function of A and W :

$$MT = a / b AW \quad (9.2)$$

The Steering Law can alternatively model instantaneous velocity (Accot and Zhai 1997). A limitation of the Steering Law is that it only models successful completion of the task; errors are not considered.

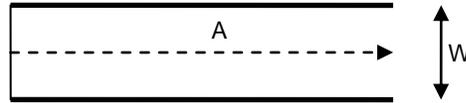


Figure 9.7. *Steering through a tunnel of length A and width W .*

The Minimum Jerk Law (Viviani and Flash 1995) characterizes the dynamics of motions that lack a continuous accuracy constraint. No one has yet formulated a universal law that handles varying accuracy constraints and curvature (Lank and Saund 2005).

The Keystroke-Level Model (KLM) and GOMS Analysis

The KLM is an engineering and analysis tool that can be used to derive a *rough estimate* of the time needed for expert users to complete a routine task (Card, Moran, and Newell 1980). To apply the KLM, count the elemental inputs required to complete a task, including (a) keystrokes, (b) homing times to acquire input devices, (c) pauses for mental preparation, and (d) pointing at targets. For each elemental input, substitute a constant estimate of the average time required using the values from (Card, Moran, and Newell 1980), or by collecting empirical data (Hinckley et al. 2006), and sum them to yield an overall time estimate. The model assumes error-free execution, so it cannot estimate time for the problem-solving behaviors of novices, but it does employ several heuristics to model mental pauses (Raskin 2000).

GOMS (Goals, Objects, Methods, and Selection rules) models extend the keystroke-level model (John and Kieras 1996). Some GOMS models can account for user knowledge and interleaving of tasks, but are more difficult to apply than the KLM. Both GOMS and KLM models are engineering tools that produce estimates for expert completion time of routine tasks. These models do not replace the need for usability testing and evaluation, but they do offer a means to assess a design without implementing software, training end users, and evaluating their performance (Olson and Olson 1990). Physical articulation times derived from KLM or GOMS analyses can also be used to help interpret results of empirical studies (Hinckley et al. 2006).

TRANSFER FUNCTIONS: HOW TO TRANSFORM AN INPUT SIGNAL

A transfer function is a mathematical transformation that scales the data from an input device. Typically, the goal is to provide more stable and more intuitive control, but one can easily design a poor transfer function that hinders performance. Here we discuss some simple transformations that are commonly used for various interaction techniques, but the reader should be aware that a deep understanding of the topic requires expertise in control theory, which is well beyond the scope of this chapter (and indeed beyond the expertise of its authors).

Appropriate Mappings. A transfer function that matches the properties of an input device is known as an appropriate mapping. For force sensing input devices, the transfer function should be a force-to-velocity function; for example, the force one exerts on an isometric joystick controls the speed at which the cursor moves. Other appropriate mappings include position-to-position or velocity-to-velocity functions, used with tablets and mice, respectively.

A common example of an inappropriate mapping is calculating a velocity based on the position of the mouse cursor, such as automatically scrolling a document when selecting a large region of text that extends beyond a single screen. The resulting input is difficult to control, and this inappropriate rate mapping is only necessary because the operating system clips the cursor to the screen edge. A better solution would be to ignore the cursor position and instead use the relative position information reported by the mouse to directly control the change of position within the document.

Self-centering Devices. Rate mappings suit force-sensing devices or other devices that return to center when released (Zhai 1993; Zhai, Smith, and Selker 1997). This property allows the user to stop quickly by releasing the device. The formula for a nonlinear rate mapping is:

$$dx = K x^a \quad (9.4)$$

Where x is the input signal, dx is the resulting rate, K is a gain factor, and a is the nonlinear parameter. The best values for K and a depend on the details of the device and application, and appropriate values must be identified by experimentation or optimal search (Zhai and Milgram 1993). Many commercial devices use more complex mappings (Rutledge and Selker 1990).

Motion-sensing devices. Desktop systems use an exponential transformation of the mouse velocity, known as an acceleration function, to modify the cursor response (Microsoft Corp. 2002). Acceleration functions do not directly improve pointing performance, but do reduce the footprint required by a device (Jellinek and Card 1990), which may lead to greater comfort or less frequent clutching (Hinckley et al. 2002).

Absolute Devices. Absolute devices offer a fixed, 1:1 control to display mapping. This is common in touch-screen and pen input. Relative transfer functions may also offer a 1:1 mapping of movements.

Mixed-mode Absolute/Relative. It is possible to temporarily violate the 1:1 control-to-display mapping of absolute devices such as touch-screens by damping small motions to provide fine adjustments; large motions revert to an absolute 1:1 mapping (Sears and Shneiderman 1991). A drawback of such techniques is that cursor feedback in the tracking state becomes the default behavior of the device, rather than dragging (Buxton 1990), but researchers have demonstrated ways to overcome this by only adjusting cursor position beneath the finger, automatically escalating precise-pointing interactions to indirect input, or providing means for the user to quickly switch between direct and indirect mode (Benko, Wilson, and Baudisch 2006; Vogel and Baudisch 2007; Forlines, Vogel, and Balakrishnan 2006). Other hybrids of direct and indirect devices, with mixed relative and absolute input mappings, have also been explored

FEEDBACK: WHAT HAPPENS IN RESPONSE TO AN INPUT?

From the technology perspective, one can consider feedback as active or passive. Active feedback is under computer control; passive feedback is not, and may result from internal sensations within the user's own body, such as muscle tension from holding down a button, or physical properties of the device, such as the feel of clicking its buttons.

Good industrial design guides the user to the appropriate affordances for a device as soon as a user holds it, and perhaps even suggests its intended uses before the user even touches it (Norman 1990); (Buxton 2007). Mechanical sounds and vibrations produced by a device provide positive feedback for the user's actions (Lewis, Potosnak, and Magyar 1997). The shape of the device and the presence of tactile landmarks can help users acquire and orient a device without having to look at it (Hinckley et al. 1998).

Proprioceptive and Kinesthetic Feedback

Internal sensations of body posture, motion, and muscle tension (Burdea 1996; Gibson 1962) may allow users to feel how they are moving an input device without looking at the device or receiving visual feedback on a display. This is important when the user's attention is divided between multiple tasks and devices ((Balakrishnan and Hinckley 1999; Fitzmaurice and Buxton 1997; Mine, Brooks, and Sequin 1997). Muscular tension can help to phrase together multiple related inputs (Buxton 1995) and may make mode transitions more salient to the user (Hinckley et al. 2006; Raskin 2000; Sellen, Kurtenbach, and Buxton 1992; Hinckley et al. 2010).

Kinesthetic Correspondence

With time users can adapt to any consistent mapping (Cunningham and Welch 1994), so long as the mapping between input and display is a planar reorientation. Despite this ability, graphical feedback on the screen ideally should correspond to the direction that the user moves the input device (Britton, Lipscomb, and Pique 1978). If the user moves a device to the left, then the object on the screen should likewise move left; however, users can easily adapt to certain kinds of non-correspondences: when the user moves a mouse forward and backward, the cursor actually moves up and down on the screen; if the user drags a scrollbar downward, the text on the screen scrolls upwards. Researchers have also found that the dimensions of an input device should match the perceptual structure of a task (Jacob et al. 1994).

To Cursor or not to Cursor (with Direct Input)

The cursor serves as a proxy for the user when engaged in indirect input. When using direct input, the user's finger or a mechanical intermediary functions as its own indicator for a well-calibrated system (Potter, Weldon, and Shneiderman 1988). Despite this, there are uses for a cursor. In devices that can sense a hover state prior to the finger or stylus coming into contact with a digitizer (Buxton 1990), the cursor can show the user the target that will result before they commit to it, and it serves as a precise indicator of the contact location (Sutherland 1964). Further, iconic cursors serve as a useful indicator of state (Tilbrook 1976). Finally, the presence of the cursor and its response to user input gives feedback to the user that the system is active, tracking, and ready to receive commands.

Echo Feedback vs. Semantic Feedback

Input devices, through the lens of interaction techniques, process a stream of sensor values to yield a logical result. A choice can be made of whether the system's feedback will echo unprocessed sensor data back to the user ("here's what the system sees")(Krueger 1991) or instead provide a semantic representation of the user's state ("here's what the system knows") like the cursor (Sutherland 1964). Traditional systems have trended towards the latter. A mouse, for example, senses only movement, but the feedback given to the user is of a *cursor position*, which is a logical state maintained entirely for the purposes of enabling interaction between the computer and the user. In point-based interactions, the alternative (echoing back movement without showing a cursor) makes little sense. Richer input streams, meanwhile, might tempt the designer to skew their feedback more towards the unprocessed data, since it may represent a richer visualization (Buxton 1990; Freeman et al. 2009; Wilson and Benko 2010). While richer, such a representation may not help the user to understand cause and effect. Designs can also include both representations, as illustrated by the LucidTouch system (Wigdor, Forlines, et al. 2007). LucidTouch shows "raw" camera data that makes the device appear transparent, but also includes "touch cursors" to represent system state. The echo feedback of the raw hand images establishes the conceptual model of the device by leading the user to associate each touch cursor to the finger that controls it.

Impoverished Physicality

Modern input devices such as touch-screens and in-air gesture systems lack some of the tactile and kinesthetic feedback inherently present in traditional input devices. Such feedback is an essential element of the user experience, especially when users are attempting to understand why the system response may not be as expected. Many commercial mobile devices use audio to compensate, with phones that beep or click when a virtual key is pressed.

To understand the role that feedback plays, consider the following table (Table 3), which describes various states of a system and the feedback that is provided by either the cursor or the hardware itself. As is immediately evident, most touch-based platforms shift a great deal of the feedback burden onto the application developer.

Cause of Unexpected Behaviour	Feedback Refuting Cause	
	Mouse	Touch
System is non-responsive	OS: Pointer movement	Application
Hardware failed to detect input	HW: Activation of button	Application
Input delivered to wrong location	OS: Visible pointer	Application
Input does not map to expected function	Application	Application
System is in a mode	OS + Application: Pointer icon	Application
Max size reached	OS: Pointer moves past edge	Application
Accidental input (arm brushing)	N/A	Application
Over constrained (too many contacts)	N/A	Application
Stolen capture (2 nd user captures control)	N/A	Application

Table 3. Potential causes of unexpected behavior (left), and the source of feedback that users receive to refute those causes, in representative mouse vs. touch input systems.

However, this difficulty can be addressed by introducing feedback visualizations that represent these specific states, and that replace physical responses with visual ones. Such approaches can help to overcome issues of impoverished physicality and semantic feedback, while making touch devices appear more responsive to users (Wigdor et al. Ripples).

Snapping Behaviors, Active Haptic Feedback, and Tactile Feedback

Software constraints, such as snapping (Baudisch et al. 2005), often suffice to support a user's tasks without resort to exotic haptic or tactile feedback mechanisms. Active force or tactile feedback (Burdea 1996) can provide attractive forces for a target, or additional feedback for the boundaries of a target, but when evaluated for indirect pointing devices, such feedback has typically yielded little or no performance advantage even for isolated target selection (Akamatsu and Mackenzie 1996); (MacKenzie 1995). Such techniques must evaluate selection among multiple targets, because haptic feedback or snapping behavior for one target may interfere with the selection of others ((Grossman and Balakrishnan 2005; Oakley, Brewster, and Gray 2001). Visual dominance refers to the tendency for vision to dominate other modalities (Wickens 1992); haptic feedback typically must closely match visual feedback, which limits its utility as an independent modality (Campbell et al. 1999). One promising use of tactile feedback is to improve state transitions, particularly for direct input devices ((Poupyrev and Maruyama 2003; Snibbe and MacLean 2001; Bau et al. 2010).

KEYBOARDS AND TEXT ENTRY

Typewriters have been in use for over 140 years; the QWERTY key layout dates to 1868 (Yamada 1980). Those who continue to predict the demise of keyboards would be wise to consider the resilience of this design in the face of the nearly unimaginable orders of magnitude of change that have vastly transformed nearly all other aspects of computing hardware.

Despite the antiquity of the design, QWERTY keyboards are extremely well suited to human performance and, at least for heavy text entry, mechanical keyboards are unlikely to be supplanted by new key layouts, speech recognition technologies, or other techniques any time soon.

Many factors can influence typing performance, including key size, shape, activation force, key travel distance, and the tactile and auditory feedback provided by striking the keys (Lewis, Potosnak, and Magyar 1997), but these well-established design details are not our focus here. We also discuss touch-screen "soft" keyboards, which have some commonality with mechanical keyboards, but raise additional design issues, particularly because they are so demanding of visual attention.

Procedural Memory and the Power Law of Practice

Procedural memory facilitates human performance of complex sequences of practiced movements, such as riding a bike, seemingly without any cognitive effort (Anderson 1980). Procedural memory (which is sometimes informally referred to as "muscle memory") is a distinct type of memory that resides below a person's conscious awareness. The encoding of skill in procedural (Anderson 1980). Procedural memory enables touch typing on a keyboard with minimal attention. As a result, users can focus their attention on mental composition and verification of the text appearing on the screen. Dedicated keys or chorded key presses for frequently used commands (hotkeys) likewise allow rapid command invocation (McLoone, Hinckley, and Cutrell 2003). The automation of skills in procedural memory is described by the power law of practice:

$$T = aP^b \quad (9.5)$$

Here, T is the time to perform a task, P is the amount of practice, and the multiplier a and exponent b are fit to the observed data (Anderson 1980). For a good example of applying the power law of practice to text entry research, see (MacKenzie et al. 2001).

Alternative keyboard layouts such as Dvorak offer perhaps a 5% performance gain (Lewis, Potosnak, and Magyar 1997), but the power law of practice suggests this small gain comes at a substantial cost for retraining time. Split-angle ergonomic QWERTY keyboards are close enough to the standard layout that they preserve much of a user's existing skill for typing. They have also been shown to help maintain neutral posture of the wrist and thus avoid ulnar deviation ((Honan et al. 1995; Marklin, Simoneau, and Monroe 1997; Smutz et al. 1994), which has been associated with increased pressure in the carpal tunnel (Putz-Anderson 1988; Rempel et al. 1998).

Two-Thumb Mechanical Keyboards

Many designs for cell phones and other handheld devices, such as the RIM Blackberry, offer two-thumb keyboards with QWERTY key layouts. The principal virtue of QWERTY is that common pairs of letters tend to occur on opposite hands. This alternation is a very efficient movement pattern for both standard and two-thumb keyboards, since one hand completes a key press while the other hand moves to the next key (MacKenzie and Soukoreff 2002). A recent study found that two-thumb keyboards offer text entry rates approaching 60 wpm (Clarkson et al. 2005). This suggests that one-handed text entry rates are fundamentally limited due to the serial nature of character entry, despite novel improvements (MacKenzie et al. 2001; Wigdor and Balakrishnan 2003). Word prediction may help, but also requires overhead for users to monitor and decide whether to use the predictions. Such techniques also require escape mechanisms to allow entry of out-of-vocabulary words.

Touch-Screen Keyboards

Modern multi-touch-screens enable text entry that is adequate for mobile interaction, thus bringing touch-screen keyboards to the mainstream. Yet the user experience with a graphical touch-screen keyboard still faces many challenges and shortcomings that may be addressed by future innovations and optimizations (Gunawardana, Paek, and Meek 2010). There have been many

studies of touchscreen key sizes (Sears 1991), or of optimal target sizes in general (Vogel and Baudisch 2007), but in practice the key size is more-or-less dictated by the screen dimensions. Graphical keyboards demand significant visual attention because the user must look at the screen to press the correct key. It therefore splits the user's visual attention between the workspace (where text is being inserted) and the graphical keyboard itself. This is particularly problematic on larger form-factors, such as slates, because the insertion point may be relatively far from the keyboard; if one brushes the screen by accident while typing (for example), one may not even notice that characters are being mistakenly inserted at a new and unintended location. The quality of the tactile feedback is impoverished as compared to a physical keyboard because the user cannot feel the key boundaries. Many graphical keyboards add audible "clicks" to provide confirmatory feedback, but it is unclear if this actually benefits performance (Sears 1991; Lewis, Potosnak, and Magyar 1997). A graphical keyboard (as well as the user's hand) occludes a significant portion of a device's screen, resulting in less space for the document itself. Furthermore, because the user typically cannot rest their fingers in contact with the display (as one can with mechanical keys), and also because one must carefully keep other fingers pulled back so as to not accidentally touch keys other than the intended one, extended use of touchscreen keyboards can be fatiguing.

Innovative design hybrids that blend touch or stylus-based typing with stroke gestures have been shown to produce high rates of text entry, once the user masters them (Kristensson and Zhai 2004; Zhai and Kristensson 2003; Zhai et al. 2009). It remains unclear if such approaches will achieve widespread adoption.

Handwriting and Character Recognition

Handwriting (even on paper, with no "recognition" involved) proceeds at about 15 wpm. Thus, a pen is a poor replacement for a keyboard; but, of course, a keyboard is an equally poor replacement for a pen. The specter of recognition arises as soon as one contemplates marking a virtual sheet of paper, but it is important to keep in mind that ink has significant value as a natural data type without recognition: it offers an expressive mix of writing, sketches, and diagrams, and when used to annotate a document, concise pen marks or highlights implicitly emphasize the important points in context. Freeform pen input also may help users to generate a breadth of design concepts (Buxton 2007; Oviatt, Cohen, and Mann 2010), as opposed to more rigid input mechanisms such as keyboard-based text entry.

Handwriting recognition technology on the Tablet PC has improved markedly over the past decade. Nonetheless recognizing natural handwriting remains difficult and error prone for computers, and demands error correction input from the user. Handwriting recognition works well for short phrases such as search terms (Hinckley et al. 2007), or for background tasks such as indexing handwritten documents for search, but converting lengthy handwritten passages to error-free text remains a tedious process. Hence, while handwriting recognition is an important enabling technology, in our view pen-operated devices can best avoid the graveyard by emphasizing those user experiences that make minimal demands of recognition, and instead emphasize the virtues of ink as a uniquely expressive data type.

To make performance more predictable for the user, some devices rely on character recognition, often implemented as single-stroke ("unistroke") gestures (Goldberg and Richardson 1993). Unistroke alphabets attempt to strike a design balance such that each letter is easy for a computer to distinguish, yet also straightforward for users to learn (MacKenzie and Zhang 1997). With the widespread adoption of touch-screen keyboards, coupled with the large strides made in handwriting recognition, such approaches have fallen out of favor in most contexts.

MODALITIES OF INTERACTION

In the search for designs that enhance interfaces and enable new usage scenarios, researchers have explored many input modalities and interaction strategies that transcend any specific type of device.

Bimanual Input

People use both hands to accomplish most real-world tasks (Guiard 1987), but computer interfaces make little use of the nonpreferred hand for tasks other than typing. Bimanual input enables compound input tasks such as navigation/selection

tasks, where the user can scroll with the nonpreferred hand while using the mouse in the preferred hand (Buxton and Myers 1986). This assignment of roles to the hands corresponds to Guiard's kinematic chain theory (Guiard 1987): the nonpreferred hand sets a frame of reference (scrolling to a location in the document) for the action of the preferred hand (selecting an item within the page using the mouse).

Interaction with handheld devices often requires two hands for some tasks. For example, users often hold a device with the nonpreferred hand, while the preferred hand performs pinch-to-zoom gestures, taps on small targets, or punches out messages on a soft keyboard. Researchers have explored using the nonpreferred hand for spatial manipulation, by moving or tilting the device (Fitzmaurice and Buxton 1994; Hinckley and Song 2010). This approach leaves the preferred hand free to point or sketch at the content thus revealed.

Some example applications for bimanual input include command selection ((Bier et al. 1993; Kabbash, Buxton, and Sellen 1994), drawing tools (Kurtenbach et al. 1997), and virtual camera control and manipulation (Balakrishnan and Kurtenbach 1999; Hinckley et al. 1998)). Integrating additional buttons and controls with keyboards to encourage bimanual interaction can also improve the efficiency of some common tasks (MacKenzie and Guiard 2001; McLoone, Hinckley, and Cutrell 2003)). Nonpreferred-hand initiated mode switching, such as by holding down a mode button, has also been demonstrated to be a particularly effective means of changing mode in pen-based interfaces (Li et al. 2005).

Gesture recognition vs. physics-based manipulation

As previously discussed, researchers have found that there does not exist a universal set of 'natural' gestures which users will perform without appropriate affordances (Wobbrock, Morris, and Wilson 2009). The sole exception is physical manipulation: moving an object from one place to another or otherwise changing its position or orientation. Physics-based systems extend this by mapping inputs to a virtual world governed by Newtonian physics, leading to user experiences described as 'natural' (Agarwala and Balakrishnan 2006; Wilson et al. 2008; Wilson 2009). This approach has been characterized as "reality-based interaction", which advocates for using naïve physics combined with awareness of the body, surrounding environment, and social interaction as the bases for successful user interfaces (Jacob et al. 2008). This is in contrast to an approach where gestures are specifically recognized to differentiate system responses, such as assigning functions to particular hand postures (Baudel and Beaudouin-Lafon 1993). Physics-based systems have the advantage of 'feeling' like the real world, but have not yet been demonstrated to scale to enable the range of functions expected of an interactive system. A hybrid model has also been demonstrated, where shape-based information is used to manipulate a traditional WIMP GUI (Moscovich 2009). Taxonomies of gestures of touch-based computing are provided by (Wobbrock, Morris, and Wilson 2009) and (Freeman et al. 2009).

Pen and Pen-Based Gesture Input

Pens lend themselves to command gestures analogous to proofreader's marks, such as crossing out a word to delete it. Note that in this example, the gesture integrates the selection of a delete command with the selection of the word to be deleted. Another example is moving a paragraph by circling it and drawing a line to its new location. This integrates the verb, object, and indirect object by specifying the command, the extent of text to move, and the new location for the text (Hinckley, Baudisch, et al. 2005; Kurtenbach and Buxton 1991). Marking menus use straight-line gestures along the primary compass directions for rapid command selection (Kurtenbach, Sellen, and Buxton 1993; Zhao and Balakrishnan 2004).

Pen interfaces must decide whether to treat pen strokes as ink content or as gesture commands. Some applications avoid this recognition problem by treating all strokes as commands (Kurtenbach and Buxton 1991), but for a free-form drawing or note-taking application, users need to interleave ink content and command input. The status-quo solution presents commands in a toolbar or menu at the edge of the screen; however, this necessitates round trips between the work area and the command area (Fitzmaurice, Khan, Piek, et al. 2003), which becomes inconvenient in direct proportion to the display size. Pressing a button with the nonpreferred hand is a fast and robust means to switch between ink and gesture modes (Li et al. 2005).

Techniques to automatically distinguish ink and gestures have been proposed, but only for highly restricted gesture sets (Saund and Lank 2003). Punctuation (tapping) has also been explored as a way to both identify and delimit command phrases (LaViola 2004). A fundamental problem with both of these approaches is that the system cannot classify a set of strokes as a gesture or as ink until after the user has finished drawing the entire command phrase. This makes it difficult to provide interactive feedback or to prompt the user with the available commands before the user commits to an operation.

While moving the pen to toolbars at the edge of the screen seems slow on a tablet computer, in practice, this "round trip

strategy” (Fitzmaurice, Khan, Piekl, et al. 2003) is difficult to improve upon. On a tablet the size of a standard 8.5 x 11 inch sheet of paper, a round trip requires approximately 1.5 seconds; however, the user can mentally prepare for the next step of the interaction while moving the pen. A locally drawn gesture (such as a straight-line marking menu command) may take less time to articulate, but thinking about what command to select requires additional time unless the task is a routine one. Pressing a button for gesture mode also requires some overhead, as does lifting the pen at the end of the gesture. Also note that performing a sequence of gestures (e.g., tagging words in a document as keywords by circling them) requires time to travel between screen locations. The round-trip strategy absorbs this travel time into the round trip itself, but with gestures, this is an extra cost that reduces the benefit of keeping the interaction localized.

Thus, on a tablet-sized device, it is difficult to realize a substantial time savings just by reducing round trips. For our hypothetical task of tagging keywords in a document, **Error! Reference source not found.** illustrates this predicament for average task times drawn from recent studies (Hinckley et al. 2006; Li et al. 2005). The chart shows two successive command selections, and assumes some mental preparation is required before issuing each command. Thus, the potential benefit of pen gestures depends on the -sequence of operations as well as the elimination of multiple round trips, as may be possible with techniques that integrate selection of verb, object, and indirect object ((Hinckley, Baudisch, et al. 2005; Kurtenbach and Buxton 1991). Localized interaction may also offer indirect benefits by reducing physical effort and by keeping the user’s visual attention focused on their work (Grossman et al. 2006; Kabbash, Buxton, and Sellen 1994).

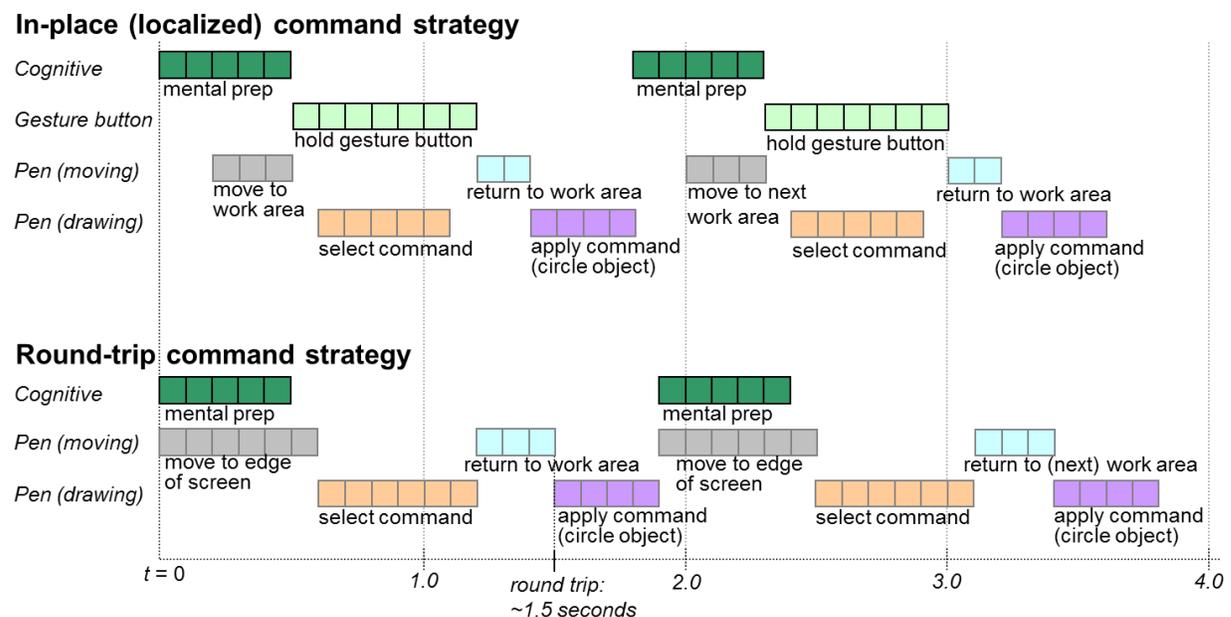


Figure 9.8. Time series contrasting the elements of command selection for in-place command selection, versus a round trip to the edge of the screen (Hinckley et al. 2006). These are approximate times for a tablet-sized device (of about 30cm screen diagonal).

Speech, Multimodal Input, and Situated Interaction

Speech has substantial value without recognition. Computers can augment human-human communication across both time and space by allowing users to record, edit, replay, or transmit digitized speech and sounds (Arons 1993; Stifelman 1996; Buxton 1995). Recording simultaneous speech and handwritten annotations is also a compelling combination for human-human communication and collaboration (Levine and Ehrlich 1991). Systems have used microphone input to detect ambient speech and employ this as a cue to optimize interaction with devices (Horvitz, Jacobs, and Hovel 1999; Sawhney and Schmandt 2000; Schmandt et al. 2000), or to provide dual-purpose speech which serves a communicative purpose while also cueing specific actions for a computer (Lyons et al. 2004).

Speech recognition can succeed for a limited vocabulary, such as speaking the name of a person from one’s contact list to place a cell phone call; however, error rates increase as the vocabulary and complexity of the grammar grows, if the quality of the audio signal from the microphone is not good enough, or if users employ out-of-vocabulary words. It is difficult to use speech

to refer to spatial locations, so it cannot eliminate the need for pointing (Cohen et al. 1989; Oviatt, DeAngeli, and Kuhn 1997); Keyboard-mouse text entry for the English language is about twice as fast as automatic speech recognition (Karat et al. 1999); furthermore, speaking can sometimes interfere with one's ability to compose text and remember words (Karl, Pettey, and Shneiderman 1993). Finally, speech is inherently non-private in public situations. Thus, speech has an important role to play, but claims that speech will soon supplant manual input devices should be considered with skepticism.

Computer understanding of human speech does not enable users to talk to a computer as one would converse with another person. Doing spoken dialog well entails *situated interaction*, which goes far beyond speech recognition itself. For computers to embed themselves naturally within the flow of human activities, they must be able to sense and reason about people and their intentions: in any given dialog, multiple may people come and go, they may interact with system or with each other, and they may interleave their interactions with other activities such that the computational system is not always in the foreground (Bohus, Horvitz, and 2010). Dealing with such challenges pushes existing technologies to their limits, and unfortunately well beyond them at times, as well.

Free-Space Gestures & Whole Body Input

Humans naturally gesture and point using their hands during verbal communication, which has motivated research into freehand gestures, often in combination with speech recognition (Bolt 1980; Hauptmann 1989; Wilson and Oliver 2003; Tse et al. 2008). Cadoz categorizes hand gestures as semiotic, ergotic, or epistemic. Semiotic gestures, such as “thumbs up,” communicate information (Cadoz 1994; Rime and Schiaratura 1991). Ergotic gestures manipulate physical objects. Epistemic gestures are exploratory motions that gather information (Kirsh 1995; Kirsh and Maglio 1994). The interaction literature often focuses on empty-handed semiotic gestures (Freeman and Weissman 1995; Jovic et al. 2000; Maes et al. 1997). A major challenge is to correctly identify when a gesture, as opposed to an incidental hand movement, starts and stops (Baudel and Beaudouin-Lafon 1993; Wilson and Oliver 2003). The lack of deterministic state transitions (Buxton 1990; Vogel and Balakrishnan 2005) can lead to errors of user intent or errors of computer interpretation (Bellotti et al. 2002). Other problems include fatigue from extending one's arms for long periods, and the imprecision of pointing at a distance.

By contrast, tangible interaction techniques (Ishii and Ullmer 1997) and augmented devices (Harrison et al. 1998) sense ergotic gestures via a physical intermediary (Hinckley et al. 1998; Zhai, Milgram, and Buxton 1996). The emergence of cameras, cell phones, and tablets augmented with accelerometers, gyroscopes, and other sensors suggest this is a promising design space.

Whole-body input is also possible, typically utilizing computer vision techniques (Krueger, Gionfriddo, and Hinrichsen 1985). Producing whole-body input by processing the imagery captured by a single camera significantly limits the vocabulary of input to those things that are clearly observable by that single camera, typically to 2-dimensional manipulations within the viewing plane (Krueger, Gionfriddo, and Hinrichsen 1985). More recent technologies have augmented this sensor stream with the distance of objects from the camera, enabling more subtle, 3D interactions. This has been demonstrated by processing two camera images simultaneously and observing binocular disparity. (Ko and Yang 1997; Matsushita and Rekimoto 1997). It has also been demonstrated with techniques such as time-of-flight cameras (Wilson 2007), as well as structured light techniques. Commercial products such as Microsoft Kinect provide depth information for each pixel, which has the potential to enable richer interaction, such as correcting the projected image to allow for consistently-sized projection of objects onto moving surfaces (Wilson 2007).

Background Sensing Techniques

Sensors can enable a mobile device to sense when the user picks up, puts down, looks at, holds, or walks around with the device. These actions give a device information about the context of its use, and represent a hidden vocabulary of naturally occurring gestures that people spontaneously exhibit in day-to-day activity. For example, many current smart phones and slate devices employ a tilt sensor to interactively switch the display between portrait and landscape formats, as well as to automatically save photos in the correct orientation (Hinckley, Pierce, et al. 2005; Hinckley et al. 2000). Here the sensor allows the device to adapt its behavior to the user's needs, rather than requiring the user to take extra steps to control the display format and photo orientation (Buxton 1995).

Sensors can also be embedded in the environment. When one walks into a modern department store, there is no explicit command to open the doors: the doors sense motion and automatically open. Researchers are investigating new ways to leverage such contextual sensing to enrich and simplify interaction with devices and digital environments (Abowd and Mynatt 2000; Schilit, Adams, and Want 1994).

Projector-Based Augmented Reality

Direct interaction can be extended beyond screens by projecting imagery onto real-world physical objects. Such projection is normally coupled with a sensor enabling sensing of the movement of the object on which it is projected, of users' hands within the projection area, or both. Such sensing has been enabled through projection onto input devices, thus transforming indirect multi-touch input devices into direct ones (Dietz and Leigh 2001). More typical is the use of a camera with a viewing area which overlaps the projection area (Ullmer and Ishii 1997; Matsushita and Rekimoto 1997; Klemmer et al. 2001). This has been envisioned as a light bulb capable of high-resolution information display and sensing user interaction with that display (Underkoffler and Ishii 1998). The technique has been used to augment physical objects with digital information, such as overlaying instructions for mechanical operations, and augmenting data ports with network traffic information (Raskar et al. 2005). The technique has also been augmented by sensing manipulations of the projector itself, such as rolling the projector to zoom the UI (Forlines et al. 2005; Cao and Balakrishnan 2006), and by allowing multiple projectors to overlap one another envisioning multi-user interaction scenarios such as sharing calendars (Cao, Forlines, and Balakrishnan 2007). Small projectors have also been mounted on other input devices, such as a stylus and mouse, to augment those interactions (Song et al. 2009; Song et al. 2010).

Direct Muscle-Based Input and Brain-Computer Interfaces

Traditional input devices can be thought of as secondary sensors, in that they sense a physical action that is the consequence of cognition and muscle movements. An alternative approach is to attempt primary sensing by detecting brain activity and muscle movements directly. Muscle sensing is accomplished through electromyography, a technique previously employed for measuring muscular activity or controlling prosthetics. Saponas et al. demonstrated its use to enable sensing of muscle activation as fine-grained as detecting and identifying individual fingers (Saponas et al. 2009), and used in combination with touch-screen input to provide a richer data stream (Benko et al. 2009). Brain-computer interfaces, or BCI, typically employ electroencephalography or functional near-infrared spectroscopy to detect input. Projects have used the technique to detect workload and user engagement (Hirshfield, Chauncey, et al. 2009) in order to conduct usability studies, as well as to explore the possible dynamic adaptation of user interfaces based on such metrics (Hirshfield, Solovey, et al. 2009). Such work remains in its infancy, but appears to hold great promise (particularly as assistive technologies for users suffering from devastating injuries or other significant physical limitations) as sensing and signal processing techniques improve.

FORM-FACTORS FOR DIRECT (AND INDIRECT) INTERACTION

The form factor of a device is a function of both technology and the envisioned context of use of the device. That context extends, among other things, to the tasks the user is likely to want to perform, the physical environment in which the device is likely to be found, and the other devices that populate the surrounding ecosystem. Mobile phones and desktop computers enable some overlapping tasks, and may even be put to complementary uses by the same user at the same time for some tasks, but for each device the focus of the design is on different usage contexts.

Here, we provide some example devices to sketch the terrain and show examples of how hardware form factor, input capabilities, and display size combine to create unique user experiences, and how each of these classes of device, in turn, offers unique interaction properties-- and hence user experiences. In practice, these user experiences may be realized only if the designers focus on the depth of that experience by designing for the context of use of the device, which perhaps necessitates foregoing breadth of solution.

Handhelds & Smartphones

Handheld devices promise quick access to information in many contexts, since they are easily carried in a pocket or purse. Early on, these were dubbed "personal digital assistants" (PDA's), whose purpose was to track contacts, make notes, and keep a calendar. Modern versions are highly valued not only for the information they contain, but also because they are persistently online, promising instant access to real-time data, with e-mail, web browsing, and of course telephony as core functions. Their convenience often overcomes the limitations of these devices, namely their tiny screens and inefficient text entry. A symptom of the success as well as the inadequacies of such devices is the caveat of *sent from my mobile device* that pervades automatic e-mail signatures on smartphones, as if the messages were crafted by a carpenter who blames his tools for the shoddy construction.

One of the first real successes for handheld computing was the Palm Pilot. The Palm Pilot was designed to fit in a man's shirt pocket. The screen was a resistive touch digitizer, enabling interaction with either finger or a plastic stylus. While one-handed interaction was possible in some situations, the form factor was primarily intended to be held in the nonpreferred hand, while the preferred hand interacted with the screen via finger or stylus. The Palm Pilot's designers recognized that they could enable faster text entry and more precise pointing by using the stylus, which was sheathed in the bezel of the device.

While the Palm Pilot's form factor was ostensibly similar to the Apple iPhone, the iPhone replaced the Palm Pilot's resistive screen with a highly sensitive capacitive touch-screen, enabling multi-touch interactions such as the now-ubiquitous pinch gesture (Krueger, Gionfriddo, and Hinrichsen 1985). Perhaps even more important than the multi-touch capability was the soft-touch contact sensing afforded by the capacitive touch screen. Soft-touch sensing enabled more precise pointing as well as pervasive use of dragging for interactive manipulations such as panning, zooming, and scrolling with inertia, in a way that was difficult to support well on the prior generation of resistive touch-screens. Of course, modern processors, graphics hardware, and well-designed animated feedback also made the interactions clear and rewarding for users to experience.

Many handheld devices are designed for use predominantly with a single hand, with the device secured by the palm while the thumb interacts with the screen. Because the thumb is significantly less precise than stylus input, a specialized user interface for the thumb should be considered (Karlson 2007). For example, zooming user interfaces enable interaction with large amounts of data on a small screen while using only the thumb (Karlson 2005). Ironically, however, the iPhone's iconic pinch-to-zoom gesture requires two hands (one hand to hold the device, while the other hand articulates the pinch gesture itself). Touching the screen while tilting a handheld device offers an alternative for one-handed continuous zooming on (Hinckley and Song 2010).

Both the Palm Pilot and the iPhone do away with a physical keyboard in favor of a larger screen. As always, this is a trade-off supporting one use case (viewing a variety of data sources) over another (entering text for documents, e-mails, etc.). A variety of devices make the other choice, opting for a physical keyboard. A now classic form factor is the 'clamshell', which provides a physical keyboard attached to the bottom of a display, which affords use either by two-thumb typing or by placing the device on a table. A noteworthy early device in this space is the Psion Series 5 device (**Error! Reference source not found.**). The Psion included a physical keyboard, along with a resistive touch digitizer. Its hinging mechanism was a clever touch of its industrial design: the display slides forward as the device opens, such that the base of the device cantilevers the display. This enables the user to interact with the angled touch-screen without it flopping around, or toppling the device.



Figure 9.9. *The Psion Series 5 handheld device, circa 1995. The resistive touch screen supports stylus or touch input. Note the counterweight on the bottom part of the clamshell that prevents the device from tipping over when one interacts with the screen.*

As digital devices become physically smaller, the displays and input mechanisms they offer are shrinking.

Considerable effort was once devoted to supporting web browsing in limited screen space (e.g. Buyukkokten, Garcia-Molina, and Paepcke 2001; Trevor et al. 2001), but much of this effort now seems to have been obviated by pinch-to-zoom on touch sensitive displays, plus the emergence of mobile themes for many web sites. Techniques to make small displays virtually larger include (a) peephole displays (Fitzmaurice 1993; Yee 2003), (b) transparent overlays (Harrison et al. 1995; Kamba et al. 1996), and (c) using on-screen visuals to suggest the location of off-screen objects (Baudisch and Rosenholtz 2003). The latter, for example, is one of the principal techniques embraced by the Windows Phone 7 "Metro" UI design, by having typography and UI layout that suggests the presence of additional options beyond the periphery of the current view.

Physical manipulations such as tilting that use the device itself as an interface seem particularly well suited to small devices (Harrison et al. 1998; Hinckley et al. 2000; Rekimoto 1996). Tiny, bright, and inexpensive laser or LED projectors are just around the corner; progress on computer vision techniques suggests that interactive projection may allow small devices to project large displays and sensing surfaces (Raskar et al. 2004; Wilson 2005), but brightness and power consumption remain significant obstacles to widespread adoption of projection displays. Ways to interact with even smaller devices, such as pendants, rings, watches, or even point-like sensor/emitters, have also been considered (Ni and Baudisch 2009).

Slates and Tablets

A number of compelling direct-input devices are starting to populate the long-standing void between handheld devices and larger devices such as drafting boards, tabletop devices, and wall-mounted displays. Of course, laptops have existed in this category for many years, but these have essentially been caricatures of the traditional desktop interface.

Slates, e-readers, dual-screen booklets, and other emerging form factors such as flexible devices support new and specialized uses that stretch the boundaries of traditional computing experiences (Holman and Vertegaal 2008). The new usage contexts afforded by these devices, many of which emphasize direct interaction by touch, pen, or both, are placing new demands on input and interaction techniques, signaling an opportunity for innovation in both industry and academia. As our colleague Bill Buxton (www.billbuxton.com, accessed Sept 1, 2010) writes:

Slate computers and e-readers represent a new class of digital appliance - one targeted for casual use. With the growth of this market will emerge a new and long overdue approach to interaction - one that is in keeping with the casual intent and context of such usage and which will complement, rather than replace, interfaces that support more formal and structured activities.

Even within this emerging category of device, there remains much room for differentiation and specialization. Devices such as the Amazon Kindle focus on recreational reading. Devices such as the Apple iPad support reading as one of many dedicated applications. Both have been successful in the marketplace. From the literature we also know that in many contexts, reading tends to occur in conjunction with writing, such as a student studying a textbook, or an information worker bringing together documents to research a topic (O'Hara and Sellen 1997; Adler and van Doren 1972). Devices such as the iRex iLiad,

which supported pen annotation on top of e-book content, and the Courier project (Microsoft Courier 2010), which explored a dual-screen booklet form-factor supporting both pen and touch input, hint at further developments to come. Digital annotation (Price, Schilit, and Golovchinsky 1998) and dual-screen form factors (Chen et al. 2008; Hinckley et al. 2009) have also been explored in the research literature.

Such mid-size devices place new demands on interaction design, even when they support similar or identical touch-screen interfaces to analogous handheld devices. For example, one reason that the Apple iPad must have such a wide bezel is that it would otherwise be unclear how to grasp the device without accidentally triggering operations on its touch-screen. Our personal experience with touch-screen tablets of this sort suggests that one is nonetheless still far more likely to touch things by accident than on a corresponding hand-held touch interface. This once again illustrates how, even with a subtle shift in the underlying form factor, a desirable property of an input modality-- in this case, that one can just lightly touch the screen to initiate an action-- can also become a liability, because if you just lightly touch the screen *by accident*, you will initiate some (undesired) action. New but related problems also crop up, such as the need to ameliorate accidental palm contact when writing on such a device with a pen, which do not manifest themselves on smaller hand-held form-factors.

Desktops

Rumors of the death of the desktop computer are greatly exaggerated. After 40 years of iterative design, the personal computer, driven by a mouse and keyboard, controlling a WIMP GUI, continues to evolve. Pointing continues to be the primary unit of interaction, where the mouse's fundamental properties make it difficult to beat (Balakrishnan et al. 1997). Another primitive of interaction, scrolling, typically also has a dedicated input device in the form of the scroll wheel. Augmented desktops with additional peripheral displays for direct pen input on horizontal and vertical surfaces have also been explored (Morris, Brush, and Meyers 2008).

Desktops can be expanded with multiple monitors. Research has shown that users have a strong tendency to partition tasks between discrete physical displays; for example, users often dedicate monitors to particular tasks or types of applications (Grudin 2001). Monitor bezels influence the way that users arrange application windows (Hutchings and Stasko 2004) and lead to discontinuities in information displays (Tan and Czerwinski 2003). Thus, the presence of a bezel between multiple monitors can be both a benefit and a hindrance, depending on the user's task, but clearly one large display is not necessarily superior to multiple smaller displays separated by bezels. Augmented desktops with additional peripheral displays for direct pen input on horizontal and vertical surfaces have also been explored (Morris, Brush, and Meyers 2008).

The potential negative consequences of bezels between displays often can be mitigated by careful handling of inputs and appropriate display of graphical feedback. For example, researchers have explored applications that are aware of the seams and can adjust their layout accordingly (Mackinlay and Heer 2004), as well as windowing systems that reposition and duplicate dialog boxes (Hutchings and Stasko 2005). Interaction techniques have also been proposed to account for the physical layout of the monitors. These have include the Mouse Ether technique, where the physical gaps between monitors are echoed in a virtual motor-space that the mouse cursor moves through, leading to better correspondence between input and output in accordance with users' expectations (Baudisch et al. 2004). Researchers have experimented with assigning each monitor its own mouse cursor (Benko and Feiner 2005) as well as the use of head tracking to facilitate jumps of the mouse cursor between distant monitors (Ashdown, Oka, and Sato 2005).

Multitouch Tables and Screens

Screen size is an obvious delimiter between different categories of devices. But so too is screen orientation: research has demonstrated that large touch screens mounted horizontally afford uses distinct from screens mounted vertically.

Extensive study of tabletop interaction has unearthed many issues and techniques unique to the large horizontal form factor. Most relate to multi-user usage scenarios, which a large horizontal screen naturally affords because multiple users can sit while facing one another (Shen, Everitt, and Ryall 2003). While conducive to multi-user interaction, such an arrangement creates issues of content orientation, since an on-screen object oriented correctly for one user is necessarily oriented incorrectly for the other users (Shen et al. 2004). Systems can mitigate this problem by altering the view seen from each side of the table (Matsushita et al. 2004), by using head-worn displays as the sole display mechanism (Agrawala et al. 1997), or in combination with other information displays, such as projection directly onto an input device (Benko, Ishak, and Feiner 2004).

Sharing a single tabletop also leads to issues. Users have been observed to implicitly and fluidly delineate a tabletop display into multiple territories: personal, shared, and storage (Scott et al. 2004), varying by the size of the table (Ryall et al. 2004). It is not clear how much of this separation is due simply to issues of comfort of reach, as described by anthropometricists as the

kinetosphere (Toney and Bruce 2006), versus those dictated by the mores of social distance as studied in the field of *proxemics* (Ballendat, Marquardt, and Greenberg 2010). Viewing information horizontally and from different sides has been shown to lead to perceptual differences (Wigdor, Shen, et al. 2007) and to improve visual search efficiency (Forlines et al. 2006).

In addition to social and psychological factors, multi-user interaction with a single display breaks many existing UI paradigms, such as persistent modes, undo stacks, toolbars, and paint palettes that set a global "mode" in the interface when a user taps on the control. If your system is one of the many that does not offer user differentiation for its contacts, and your interface design includes such elements, then your design is in trouble! Hence, cooperative tools for multiple users often must seek clever design solutions that avoid all global modes in an interface.

This is mitigated somewhat by technologies such as the DiamondTouch, which can identify users and thereby provide a specific mode for each distinct user (Dietz and Leigh 2001). Addressing these and other similar issues, researchers have designed many user interface elements specifically for shared horizontal form factors, such as shared vs. individual controls (Morris et al. 2006), methods for supporting multiple off-axis viewpoints (Hancock and Carpendale 2007), multi-touch content re-orientation methods (Hancock et al. 2006), and text entry (Hinrichs et al. 2007). Researchers have also examined altering the form factor, and have demonstrated differences in user behavior based on screen size (Ryall et al. 2004). Hartmann et al. demonstrate the augmentation of a large (4' x 6') tabletop with physical controls, such as multiple keyboards and mice, as well as other physical objects (Bj et al. 2009).

Vertical Displays & Very Large Displays

Vertical wall-mounted displays enable large form factor screens. Large displays lend themselves to collaboration and sharing of information with groups (Funkhouser and Li 2000; Swaminathan and Sato 1997), as well as giving a substantial physical presence to virtual activities (Buxton et al. 2000; Trimble, Wales, and Gossweiler 2003).

The digital whiteboard is a classic example of a large vertically-oriented display (Elrod et al. 1992; Moran, Chiu, and Melle 1997; Pederson et al. 1993). Many digital whiteboards mix the physical and digital by tracking the position of real markers as the user strokes on the board. Many boards are augmented with projection on the writing area, allowing users to interact with applications while still making digital or physical marker strokes. Many digital whiteboards' reliance on projectors limits their resolution, enabling a large but coarse information display.

Alternative technologies can enable large, high-resolution displays, though such displays typically require the tiling of either projectors or individual displays. Each of these technologies has inherent limitations. Tiled screens currently require a bezel (non-display area) which creates a grid of tiles that interferes with interactions such as direct-touch dragging of items across screens (Ball and North 2005). By contrast, in projected imagery, the lack of physical separation of the display area creates areas of non-uniform brightness and color, even after careful calibration (Surati 1999). Despite these issues, large high-resolution displays have been demonstrated to aid spatial tasks (Tan et al. 2003), improve and reduce gender effects in navigation (Czerwinski, Tan, and Robertson 2002), provide utility in sense making (Andrews, Ender, and North 2010), and influence visual search and steering tasks (Bi, Bae, and Balakrishnan 2010).

Direct input on wall-mounted displays is commonplace, but the constant physical movement required can become burdensome. Hybrid approaches which mix absolute and relative direct-pointing have been proposed (Forlines, Vogel, and Balakrishnan 2006), as have methods for bringing distant targets closer to the user (Baudisch, Bederson, and Zierlinger 2002; Bezerianos and Balakrishnan 2005; Collomb et al. 2005). Interaction-at-a-distance can be afforded via indirect interaction techniques, such as touch input on a nearby tablet device (Nacenta et al. 2005; Malik, Ranjan, and Balakrishnan 2005), eye tracking (Bolt 1981; Shell et al. 2004), or by pointing at the display using a physical device (Bolt 1980; Wilson and Shafer 2003) or the hand itself (Ko and Yang 1997; Nickel and Stiefelhagen 2003; Vogel and Balakrishnan 2005). Even when a user is close to a large display, interacting with portions of the display that are out view or beyond arm's length raises challenges (Bezerianos and Balakrishnan 2005; Khan et al. 2005).

Federations of Form Factors: A Society of Devices

As form factors continue to evolve, and multiple form factors per user become more affordable, each of us is likely to increasingly interact with our digital information and with one another through an ever broadening array of devices, each suited to the contexts we have described. One early prediction of such an ecology of devices was Weiser's vision of ubiquitous computing, comprised of tabs, pads, and boards, which foreshadowed the billion dollar markets for smart phones, e-Readers,

and slate devices (Weiser 1991).

Displays of various sizes support different activities and social conventions; one of the principle challenges of ubiquitous computing is finding techniques that make it easy for users to work within a digital ecology that supports a range of tasks spanning multiple computers, displays, and interactive surfaces. Several projects have probed how to use small displays as adjuncts to large ones (Myers, Lie, and Yang 2000; Myers, Stiel, and Gargiulo 1998);(Rekimoto 1998; Streitz et al. 1999), allowing simultaneous interaction with private information on a personal device, and a shared or public context on a larger display.

What makes such federations of devices fascinating is not so much the individual devices, but rather the virtual bridges that span the spaces in between them. Wireless networking is the technology that will perhaps most disrupt traditional approaches to human-computer interaction and computing in general in the coming years, because it breaks down barriers between devices and enables a new society of devices that fill specific roles, yet can still coordinate their activities (Fitzmaurice, Khan, Buxton, et al. 2003; Want and Borriello 2000). Users need techniques that allow them to access and share information across the boundaries of individual devices, as well as to dynamically bind together multiple devices and displays to accomplish their tasks (Hinckley et al. 2004; Rekimoto et al. 2003). Such interaction techniques inherently involve multiple persons, and thus must also consider how people use physical proximity and relative body orientation (Deasy and Lasswell 1985; Hinckley et al. 2004; Sommer 1965).

The interaction designer thus must consider the full range of scale for display sizes and form-factors that may embody an interaction task, as well as the interactions between different types of devices. How can interaction migrate from watches, cell phones, handheld devices, and tablets, all the way up to desktop monitors, digital whiteboards, and interactive wall-sized displays? There is, as yet, no simple answer to such questions.

TRENDS & OPPORTUNITIES FOR INPUT

The designer of an interactive system should take a broad view of input, and consider not only traditional pointing techniques and graphical user interface widgets, but also issues such as search strategies to access information in the first place, sensor inputs that enable entirely new data types, and synthesis techniques to extract meaningful structure from data. Good search tools may reduce the many inputs needed to manually search and navigate file systems. Knowledge work requires integration of external information from web pages or data-bases (Yee et al. 2003) as well as reuse of personal information from documents, electronic mail messages, and other content authored or viewed by a user (Dumais et al. 2003; Lansdale and Edmonds 1992). Unified full-text indexing allows users to quickly query their personal information across multiple information silos, and can present information in the context of memory landmarks such as the date a message was sent, or the application used to create a document (Cutrell et al. 2006).

Sensor inputs, such as accelerometers, gyroscopes, magnetometers, proximity sensors, and other sensors are now widespread in mobile devices. Cell phones and low-power radios for wireless networking can sense their location or proximity to other devices via analysis of signal strengths (Bahl and Padmanabhan 2000; Krumm and Hinckley 2004). As another example, attempting to type a secure password on a mobile phone keypad quickly convinces one that biometric sensors or some other convenient means for establishing identity is essential to support secure interactions without completely frustrating users. Such sensors could make services such as personalization of interfaces much simpler. Research has already demonstrated how on-screen fingerprint sensing could also enable high precision touch-screen interaction (Holz and Baudisch 2010).

The need to extract models and synthesize structure from large quantities of low-level inputs suggests that data mining and machine learning techniques will become important adjuncts to interaction (Fails and Olsen 2003; Fitzmaurice, Balakrishnan, and Kurtenbach 1999; Horvitz et al. 1998). Whenever a system considers automatic actions on behalf of the user, however, an important design principle in the face of uncertainty is to “do less, but do it well” (Horvitz 1999). A key challenge for input will be representation of these structures: as streams of sensors collect information, how to make this information available to applications so that their responses can be designed and customized is, as yet, unclear, but a probabilistic framework for handling input events may be the right general approach (Schwarz et al. 2010). Applying raw inputs to a physics engine is one extreme in the spectrum of possible approaches (Wilson et al. 2008). But we could argue that it is equally extreme to reduce inputs to a lowest-common denominator, as is common practice in modern toolkits. Somewhere in between, perhaps, lies a more nuanced solution to the breadth/depth dichotomy that carries forth a meaningful representation and framework that leverages the specific and deep aspects of individual platforms, while also enabling broad solutions of very high quality that embrace the wide array of form-factors that comprise the modern society of devices.

We must make substantial progress in all of these areas to advance human interaction with technology. The forms and capabilities of these and other technologies will continue to advance, but human senses and cognitive skills will not. We will continue to interact with computers using our hands and physical intermediaries, not necessarily because our technology requires us to do so, but because touching, holding, and moving physical objects is the foundation of the long evolution of tool use in the human species (Wilson 1998).

Often cited as an economic rule of technology is Moore's law, which states that the amount of processing power available at a given price will double every 18 months. A complement to and expansion of this is that all elements of device construction, design, electronics, and manufacturing, have been and will continue to become less expensive over time. A natural consequence of this trend is the society of devices we have described, as more form factors become realizable at lower costs. Many predicted the end of the digital camera market as mobile phone cameras improved. But instead, the D-SLR market has grown significantly alongside the ever improving camera-equipped phones. Likewise, the e-Reader market, including slates such as the Amazon Kindle and Apple iPad, has seen significant expansion despite the availability of Kindle and iBook software, free of charge, for smart phones and slates. As (bureaucrats? shapers? socialites? framers? founding fathers?) of this society of devices, the designer will be well served to keep a broad perspective, including many of the issues that we have addressed here, and thereby avoid the pitfalls of meaningless consistency-- hopefully in favor of good design.

REFERENCES

- Abowd, G., and E. Mynatt. 2000. Charting Past, Present, and Future Research in Ubiquitous Computing. *ACM Transactions on Computer-Human Interaction* 7 (1):29-58.
- Accot, J., and S. Zhai. 2001. Scale Effects in Steering Law Tasks. In *Proc. CHI'2001 ACM Conference on Human Factors in Computing Systems*.
- Accot, Johnny, and Shumin Zhai. 1997. Beyond Fitts' law: models for trajectory-based HCI tasks. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. Atlanta, Georgia, United States: ACM.
- Adler, M.J., and C. van Doren. 1972. *How to Read a Book*. New York, NY: Simon and Schuster.
- Agarawala, A., and R. Balakrishnan. 2006. Keepin' It Real: Pushing the Desktop Metaphor with Physics, Piles and the Pen. Paper read at CHI 2006.
- Agarawala, Anand, and Ravin Balakrishnan. 2006. Keepin' it real: pushing the desktop metaphor with physics, piles and the pen. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*. Montréal, Québec, Canada: ACM.
- Agrawala, Maneesh, Andrew C. Beers, Ian McDowall, Bernd Frohlich, Mark Bolas, and Pat Hanrahan. 1997. The two-user Responsive Workbench: support for collaboration through individual views of a shared space. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*: ACM Press/Addison-Wesley Publishing Co.
- Akamatsu, M., and I. S. Mackenzie. 1996. Movement Characteristics Using a Mouse with Tactile and Force Feedback. *International Journal of Human-Computer Studies* 45:483-493.
- Anderson, J. R. 1980. Chapter 8: Cognitive Skills. In *Cognitive Psychology and Its Implications*. San Francisco: W. H. Freeman.
- Andrews, Christopher, Alex Endert, and Chris North. 2010. Space to think: large high-resolution displays for sensemaking. In *Proceedings of the 28th international conference on Human factors in computing systems*. Atlanta, Georgia, USA: ACM.
- Apitz, Georg, Fran\, \#231, ois Guimbreti\, \#232, and re. 2004. CrossY: a crossing-based drawing application. In *Proceedings of the 17th annual ACM symposium on User interface software and technology*. Santa Fe, NM, USA: ACM.
- Appert, C., M. Beaudouin-Lafon, and W. Mackay. 2004. Context matters: Evaluating interaction techniques with the CIS model. Paper read at Proc. of HCI 2004, at Leeds, UK.
- Arons, Barry. 1993. SpeechSkimmer: interactively skimming recorded speech. In *Proceedings of the 6th annual ACM symposium on User interface software and technology*. Atlanta, Georgia, United States: ACM.
- Ashdown, Mark, Kenji Oka, and Yoichi Sato. 2005. Combining head tracking and mouse input for a GUI on multiple monitors. In

- CHI '05 extended abstracts on Human factors in computing systems*. Portland, OR, USA: ACM.
- Bahl, P., and V. Padmanabhan. 2000. RADAR: An In-Building RF-Based User Location and Tracking System. Paper read at IEEE 19th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2000).
- Balakrishnan, Ravin, Thomas Baudel, Gordon Kurtenbach, and George Fitzmaurice. 1997. The Rockin'Mouse: Integral 3D manipulation on a plane. Paper read at ACM CHI 1997 Conference on Human Factors in Computing Systems, at New York, NY.
- Balakrishnan, Ravin, and Ken Hinckley. 1999. The role of kinesthetic reference frames in two-handed input performance. In *Proceedings of the 12th annual ACM symposium on User interface software and technology*. Asheville, North Carolina, United States: ACM.
- Balakrishnan, Ravin, and Gordon Kurtenbach. 1999. Exploring bimanual camera control and object manipulation in 3D graphics interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems: the CHI is the limit*. Pittsburgh, Pennsylvania, United States: ACM.
- Ball, Robert, and Chris North. 2005. Effects of tiled high-resolution display on basic visualization and navigation tasks. In *CHI '05 extended abstracts on Human factors in computing systems*. Portland, OR, USA: ACM.
- Ballendat, T., N. Marquardt, and S. Greenberg. 2010. Proxemic Interaction: Designing for a Proximity and Orientation-Aware Environment. Calgary, AB: University of Calgary.
- Bau, Olivier, Ivan Poupyrev, Ali Israr, and Chris Harrison. 2010. TeslaTouch: electrovibration for touch surfaces. In *Proceedings of the 23rd annual ACM symposium on User interface software and technology*. New York, New York, USA: ACM.
- Baudel, T., and M. Beaudouin-Lafon. 1993. Charade: Remote Control of Objects Using Hand Gestures. *Communications of the ACM* 36 (7):28-35.
- Baudisch, P., B. Bederson, and A. Zierlinger. 2002. Drag-and-Pop: Accelerating Drag-and-Drop on Large Screens. Paper read at published?
- Baudisch, P., and R. Rosenholtz. 2003. Halo: a technique for visualizing off-screen objects. Paper read at CHI '03.
- Baudisch, Patrick, Edward Cutrell, Ken Hinckley, and Adam Eversole. 2005. Snap-and-go: helping users align objects without the modality of traditional snapping. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. Portland, Oregon, USA: ACM.
- Baudisch, Patrick, Edward Cutrell, Ken Hinckley, and Robert Gruen. 2004. Mouse ether: accelerating the acquisition of targets across multi-monitor displays. In *CHI '04 extended abstracts on Human factors in computing systems*. Vienna, Austria: ACM.
- Bellotti, Victoria, Maribeth Back, W. Keith Edwards, Rebecca E. Grinter, Austin Henderson, and Cristina Lopes. 2002. Making sense of sensing systems: five questions for designers and researchers. In *Proceedings of the SIGCHI conference on Human factors in computing systems: Changing our world, changing ourselves*. Minneapolis, Minnesota, USA: ACM.
- Benko, H., E. W. Ishak, and S. Feiner. 2004. Collaborative mixed reality visualization of an archaeological excavation. Paper read at Mixed and Augmented Reality, 2004. ISMAR 2004. Third IEEE and ACM International Symposium on, 2-5 Nov. 2004.
- Benko, Hrvoje, and Steven Feiner. 2005. Multi-monitor mouse. In *CHI '05 extended abstracts on Human factors in computing systems*. Portland, OR, USA: ACM.
- Benko, Hrvoje, Shahram Izadi, Andrew D. Wilson, Xiang Cao, Dan Rosenfeld, and Ken Hinckley. 2010. Design and evaluation of interaction models for multi-touch mice. In *Proceedings of Graphics Interface 2010*. Ottawa, Ontario, Canada: Canadian Information Processing Society.
- Benko, Hrvoje, T. Scott Saponas, Dan Morris, and Desney Tan. 2009. Enhancing input on and above the interactive surface with muscle sensing. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*. Banff, Alberta, Canada: ACM.
- Benko, Hrvoje, Andrew D. Wilson, and Patrick Baudisch. 2006. Precise selection techniques for multi-touch screens. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*. Montré, Québec, Canada: ACM.
- Bezerianos, A., and R. Balakrishnan. 2005. The vacuum: facilitating the manipulation of distant objects. Paper read at CHI '05.
- Bi, Xiaojun, Seok-Hyung Bae, and Ravin Balakrishnan. 2010. Effects of interior bezels of tiled-monitor large displays on visual search, tunnel steering, and target selection. In *Proceedings of the 28th international conference on Human factors in computing systems*. Atlanta, Georgia, USA: ACM.
- Bier, E., M. Stone, K. Pier, W. Buxton, and T. DeRose. 1993. Toolglass and Magic Lenses: The See-Through Interface. Paper read at Proceedings of SIGGRAPH 93, Aug. 1-6, at Anaheim, Calif.
- Bier, Eric A., Maureen C. Stone, Ken Pier, William Buxton, and Tony D. DeRose. 1993. Toolglass and magic lenses: the

- see-through interface. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*. Anaheim, CA: ACM.
- Björk, S., Hartmann, Meredith Ringel Morris, Hrvoje Benko, and Andrew D. Wilson. 2009. Augmenting interactive tables with mice & keyboards. In *Proceedings of the 22nd annual ACM symposium on User interface software and technology*. Victoria, BC, Canada: ACM.
- Bohus, Dan, Eric Horvitz, and 2010. On the Challenges and Opportunities of Physically Situated Dialog Paper read at AAAI Fall Symposium on Dialog with Robots, at Arlington, VA.
- Bolt, Richard. 1980. Put-that-there: Voice and gesture at the graphics interface. *ACM SIGGRAPH Computer Graphics* 14 (3):262-270.
- Bolt, Richard A. 1981. Gaze-orchestrated dynamic windows. In *Proceedings of the 8th annual conference on Computer graphics and interactive techniques*. Dallas, Texas, United States: ACM.
- Bowman, D., E. Kruijff, J. LaViola, and I. Poupyrev. 2004. *3D User Interfaces: Theory and Practice*. Addison-Wesley.
- Brandl, P., C. Forlines, D. Wigdor, M. Haller, and C. Shen. 2008. Combining and measuring the benefits of bimanual pen and direct-touch interaction on horizontal interfaces. Paper read at *Proc. AVI '08 Conf. on Advanced Visual interfaces*.
- Brandl, P., J. Leitner, T. Seifried, M. Haller, B. Doray, and P. To. 2009. Occlusion-aware menu design for digital tabletops. Paper read at CHI 2009 Extended Abstracts.
- Britton, E., J. Lipscomb, and M. Pique. 1978. Making Nested Rotations Convenient for the User. *Computer Graphics* 12 (3):222-227.
- Brown, E., W. Buxton, and K. Murtagh. 1990. Windows on tablets as a means of achieving virtual input devices. Paper read at Interact'90, at Amsterdam.
- Burdea, G. 1996. *Force and Touch Feedback for Virtual Reality*. New York, NY: John Wiley & Sons.
- Buxton, B. 2007. *Sketching User Experiences: Getting the Design Right and the Right Design*. San Francisco: Morgan Kaufman.
- Buxton, Bill. 2007. *Sketching User Experiences: Getting the Design Right and the Right Design*
- Buxton, W. 1986. Chunking and Phrasing and the Design of Human-Computer Dialogues. Paper read at Information Processing '86, Proc. of the IFIP 10th World Computer Congress.
- Buxton, W. 1986. There's More to Interaction Than Meets the Eye. In *User Centered System Design: New Perspectives on Human-Computer Interaction*, edited by D. Norman and S. Draper. Hillsdale, N.J.
- Buxton, W. 1995. Integrating the Periphery and Context: A New Taxonomy of Telematics. Paper read at Proceedings of Graphics Interface '95, at Quebec City, Quebec, Canada.
- Buxton, W. 1995. Speech, Language and Audition. In *Readings in Human-Computer Interaction: Toward the Year 2000*, edited by R. Baecker, J. Grudin, W. Buxton and S. Greenberg: Morgan Kaufmann Publishers.
- Buxton, W. 1995. Touch, Gesture, and Marking. In *Readings in Human-Computer Interaction: Toward the Year 2000*, edited by R. Baecker, J. Grudin, W. Buxton and S. Greenberg: Morgan Kaufmann Publishers.
- Buxton, W., R. Hill, and P. Rowley. 1985. Issues and Techniques in Touch-Sensitive Tablet Input. *Computer Graphics* 19 (3):215-224.
- Buxton, W., and B. Myers. 1986. A study in two-handed input. *SIGCHI Bull.* 17 (4):321-326.
- Buxton, William. 1986. Chunking and phrasing and the design of human-computer dialogues. In *Readings in human-computer interaction: Towards the year 2000*, edited by R. Baecker, J. Grudin, W. Buxton and S. Greenberg. San Francisco, CA: Morgan Kaufmann.
- Buxton, William. 1990. Three-state model of graphical input. In *Human-computer interaction - INTERACT'90*, edited by D. Diaper. Amsterdam: Elsevier Science Publishers B. V. (North-Holland).
- Buxton, William. 1990. A three-state model of graphical input. In *Proceedings of the IFIP TC13 Third International Conference on Human-Computer Interaction*: North-Holland Publishing Co.
- Buxton, William A. S. 1995. Chunking and phrasing and the design of human-computer dialogues. In *Human-computer interaction: toward the year 2000*: Morgan Kaufmann Publishers Inc.
- Buxton, William, George Fitzmaurice, Ravin Balakrishnan, and Gordon Kurtenbach. 2000. Large displays in automotive design. *IEEE Computer Graphics and Applications* July/Aug 2000:68-75.
- Buxton, William, and Brad Myers. 1986. A study in two-handed input. In *ACM CHI 1986 Conference on Human Factors in Computing Systems*. New York, NY: ACM.
- Cadoz, C. 1994. *Les realites virtuelles*: Dominos, Flammarion.
- Campbell, C., S. Zhai, K. May, and P. Maglio. 1999. What you feel must be what you see: Adding tactile feedback to the trackpoint. In *Proceedings of INTERACT '99: 7th IFIP conference on Human Computer Interaction*.
- Cao, Xiang, and Ravin Balakrishnan. 2006. Interacting with dynamically defined information spaces using a handheld projector

- and a pen. In *Proceedings of the 19th annual ACM symposium on User interface software and technology*. Montreux, Switzerland: ACM.
- Cao, Xiang, Clifton Forlines, and Ravin Balakrishnan. 2007. Multi-user interaction using handheld projectors. In *Proceedings of the 20th annual ACM symposium on User interface software and technology*. Newport, Rhode Island, USA: ACM.
- Cao, Xiang, A.D. Wilson, R. Balakrishnan, K. Hinckley, and S.E Hudson. 2008. ShapeTouch: Leveraging contact shape on interactive surfaces. In *Horizontal Interactive Human Computer Systems, 2008. TABLETOP 2008. 3rd IEEE International Workshop on*. Amsterdam.
- Cao, Xiang, Andrew D. Wilson, Ravin Balakrishnan, Ken Hinckley, and Scott E. Hudson. 2008. ShapeTouch: Leveraging contact shape on interactive surfaces. Paper read at IEEE TABLETOP 2008 International Workshop on Horizontal Interactive Human Computer Systems
- Card, S., J. Mackinlay, and G. Robertson. 1990. The Design Space of Input Devices. Paper read at Proc. ACM CHI'90 Conference on Human Factors in Computing Systems.
- Card, S., J. Mackinlay, and G. Robertson. 1991. A Morphological Analysis of the Design Space of Input Devices. *ACM Transactions on Information Systems* 9 (2):99-122.
- Card, Stuart, William English, and B. J. Burr. 1978. Evaluation of mouse, rate-controlled isometric joystick, step keys, and text keys for text selection on a CRT. *Ergonomics* 21:601-613.
- Card, Stuart, Thomas Moran, and Allen Newell. 1980. The keystroke-level model for user performance time with interactive systems. *Communications of the ACM* 23 (7):396 - 410.
- Chen, Nicholas, Francois Guimbretiere, Morgan Dixon, Cassandra Lewis, and Maneesh Agrawala. 2008. Navigation techniques for dual-display e-book readers. In *Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*. Florence, Italy: ACM.
- Clarkson, Edward, James Clawson, Kent Lyons, and Thad Starner. 2005. An empirical study of typing rates on mini-QWERTY keyboards. In *CHI '05 extended abstracts on Human factors in computing systems*. Portland, OR, USA: ACM.
- Cohen, P. R., M. Dalrymple, D. B. Moran, F. C. Pereira, and J. W. Sullivan. 1989. Synergistic use of direct manipulation and natural language. In *Proceedings of the SIGCHI conference on Human factors in computing systems: Wings for the mind*. ACM.
- Cohen, Philip, Michael Johnston, David McGee, Sharon Oviatt, Jay Pittman, Ira Smith, Liang Chen, and Josh Clow. 1997. QuickSet: Multimodal Interaction for Distributed Applications. Paper read at ACM Multimedial 97.
- Collomb, Maxime, Mountaz Hasco, \#235, Patrick Baudisch, and Brian Lee. 2005. Improving drag-and-drop on wall-size displays. In *Proceedings of Graphics Interface 2005*. Victoria, British Columbia: Canadian Human-Computer Communications Society.
- Cunningham, H. A., and R. B. Welch. 1994. Multiple concurrent visual-motor mappings: implications for models of adaptation. *Journal of Experimental Psychology: Human Perception and Performance* 20 (5):987-999.
- Cutrell, E., D. Robbins, S. Dumais, and R. Sarin. 2006. Fast, Flexible Filtering with Phlat: Personal Search and Organization Made Easy. Paper read at CHI 2006.
- Czerwinski, Mary, Desney S. Tan, and George G. Robertson. 2002. Women take a wider view. In *Proceedings of the SIGCHI conference on Human factors in computing systems: Changing our world, changing ourselves*. Minneapolis, Minnesota, USA: ACM.
- Davidson, Philip L., and Jefferson Y. Han. 2008. Extending 2D object arrangement with pressure-sensitive layering cues. In *Proceedings of the 21st annual ACM symposium on User interface software and technology*. Monterey, CA, USA: ACM.
- Deasy, C.M., and Thomas E. Lasswell. 1985. *Designing Places for People: A Handbook on Human Behavior for Architects, Designers, and Facility Managers*. New York: Whitney Library of Design an imprint of Watson-Guption Publications.
- Dietz, P., and W. Yerazunis. 2001. Real-Time Audio Buffering for Telephone Applications. Paper read at Proc. ACM UIST 2001 Symp. on User Interface Software & Technology, at Orlando, FL.
- Dietz, Paul, and Darren Leigh. 2001 DiamondTouch: a multi-user touch technology In *Proceedings of the 14th annual ACM symposium on User interface software and technology* Orlando, Florida ACM Press.
- Dillon, R. F., J. D. Eday, and J. W. Tombaugh. 1990. Measuring the True Cost of Command Selection: Techniques and Results. Paper read at Proc. of ACM CHI'90 Conference on Human Factors in Computing Systems.
- Douglas, S., A. Kirkpatrick, and I. S. MacKenzie. 1999. Testing Pointing Device Performance and User Assessment with the ISO 9241, Part 9 Standard. Paper read at Proc. ACM CHI'99 Conf. on Human Factors in Computing Systems.
- Douglas, S., and A. Mithal. 1994. The Effect of Reducing Homing Time on the Speed of a Finger-Controlled Isometric Pointing Device. Paper read at Proc. ACM CHI'94 Conf. on Human Factors in Computing Systems.
- Douglas, S.A., and A.K. Mithal. 1997. *Ergonomics of Computer Pointing Devices, Advanced Perspectives in Applied*

Computing: Springer-Verlag.

- Douglas, Sarah A., Arthur E. Kirkpatrick, and I. Scott MacKenzie. 1999. Testing pointing device performance and user assessment with the ISO 9241, Part 9 standard. In *Proceedings of the SIGCHI conference on Human factors in computing systems: the CHI is the limit*. Pittsburgh, Pennsylvania, United States: ACM.
- Dumais, S., E. Cutrell, J. Cadiz, G. Jancke, R. Sarin, and D. Robbins. 2003. Stuff I've Seen: A system for personal information retrieval and re-use. Paper read at SIGIR 2003.
- Elrod, S., R. Bruce, R. Gold, D. Goldberg, F. Halasz, W. Jannsen, D. Lee, K. McCall, E. Pedersen, K. Pier, J. Tang, and B. Welch. 1992. Liveboard: a large interactive display supporting group meetings, presentations, and remote collaboration. Paper read at ACM CHI 1992 Conference on Human Factors in Computing Systems, at New York, NY.
- Engelhard, Lenny. 2008. Native Dual Mode Digitizers: Supporting Pen, Touch and Multi-Touch Inputs in One Device on any LCD. Paper read at Society for Information Display SID 08 Digest.
- English, W. K., D. C. Englebart, and M. L. Berman. 1967. Display-selection Techniques for Text Manipulation. *Transactions on Human Factors in Electronics* 8 (1):5-15.
- Fails, J. A., and D. R. Olsen. 2003. Interactive machine learning. Paper read at ACM Intelligent User Interfaces (IUI'03).
- Fitts, Paul M. 1954. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology* 47:381-391.
- Fitzmaurice, George, and William Buxton. 1994. The Chameleon: spatially aware palmtop computers. In *Conference companion on Human factors in computing systems*. Boston, Massachusetts, United States: ACM.
- Fitzmaurice, George, Azam Khan, Robert Piek, #233, Bill Buxton, and Gordon Kurtenbach. 2003. Tracking menus. In *Proceedings of the 16th annual ACM symposium on User interface software and technology*. Vancouver, Canada: ACM.
- Fitzmaurice, George W. 1993. Situated information spaces and spatially aware palmtop computers. *Communications of the ACM* 36 (7):38-49.
- Fitzmaurice, George W., Ravin Balakrishnan, and Gordon Kurtenbach. 1999. Sampling, synthesis, and input devices. *Communications of the ACM* 42 (8):54 - 63.
- Fitzmaurice, George W., and William Buxton. 1997. An empirical evaluation of graspable user interfaces: towards specialized, space-multiplexed input. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. Atlanta, Georgia, United States: ACM.
- Fitzmaurice, George W., Hiroshi Ishii, and William A. S. Buxton. 1995. Bricks: laying the foundations for graspable user interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. Denver, Colorado, United States: ACM Press/Addison-Wesley Publishing Co.
- Fitzmaurice, George W., Azam Khan, William Buxton, Gordon Kurtenbach, and Ravin Balakrishnan. 2003. Sentient Data Access via a Diverse Society of Devices. *ACM Queue* 1 (8 (Nov)).
- Foley, J. D., V. L. Wallace, and P. Chan. 1984. The human factors of computer graphics interaction techniques. *IEEE Computer Graphics and Applications* 4 (11):13-48.
- Forlines, Clifton, Ravin Balakrishnan, Paul Beardsley, Jeroen van Baar, and Ramesh Raskar. 2005. Zoom-and-pick: facilitating visual zooming and precision pointing with interactive handheld projectors. In *Proceedings of the 18th annual ACM symposium on User interface software and technology*. Seattle, WA, USA: ACM.
- Forlines, Clifton, and Chia Shen. 2005. DTLens: multi-user tabletop spatial data exploration. In *Proceedings of the 18th annual ACM symposium on User interface software and technology*. Seattle, WA, USA: ACM.
- Forlines, Clifton, Chia Shen, Daniel Wigdor, and Ravin Balakrishnan. 2006. Exploring the effects of group size and display configuration on visual search. In *Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work*. Banff, Alberta, Canada: ACM.
- Forlines, Clifton, Daniel Vogel, and Ravin Balakrishnan. 2006. HybridPointing: fluid switching between absolute and relative pointing with a direct input device. In *Proceedings of the 19th annual ACM symposium on User interface software and technology*. Montreux, Switzerland: ACM.
- Freeman, Dustin, Hrvoje Benko, Meredith Ringel Morris, and Daniel Wigdor. 2009. ShadowGuides: visualizations for in-situ learning of multi-touch and whole-hand gestures. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*. Banff, Alberta, Canada: ACM.
- Freeman, William, and Craig Weissman. 1995. Television control by hand gestures. Paper read at International Workshop on Automatic Face and Gesture Recognition, at Zurich, Switzerland.
- Frisch, Mathias, Jens Heydekorn, and Raimund Dachsel. 2009. Investigating Multi-Touch and Pen Gestures for Diagram Editing on Interactive Surfaces Paper read at ITS '09 Conf. on Interactive Tabletops and Surfaces.
- Funkhouser, Thomas, and K. Li. 2000. Onto the Wall: Large Displays. *Special issue of IEEE Computer Graphics and*

Applications. 20 (4).

- Gibson, J. J. 1962. Observations on active touch. *Psychological Review* 69:477-491.
- Goldberg, David, and Cate Richardson. 1993. Touch-typing with a stylus. In *Proceedings of the INTERACT '93 and CHI '93 conference on Human factors in computing systems*. Amsterdam, The Netherlands: ACM.
- Grossman, Tovi, and Ravin Balakrishnan. 2005. The bubble cursor: enhancing target acquisition by dynamic resizing of the cursor's activation area. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. Portland, Oregon, USA: ACM.
- Grossman, Tovi, Ken Hinckley, Patrick Baudisch, Maneesh Agrawala, and Ravin Balakrishnan. 2006. Hover widgets: using the tracking state to extend the capabilities of pen-operated devices. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*. Montréal, Québec, Canada: ACM.
- Grossman, Tovi, Daniel Wigdor, and Ravin Balakrishnan. 2004. Multi-finger gestural interaction with 3d volumetric displays. In *Proceedings of the 17th annual ACM symposium on User interface software and technology*. Santa Fe, NM, USA: ACM.
- Grudin, J. 2001. Partitioning Digital Worlds: Focal and Peripheral Awareness in Multiple Monitor Use. Paper read at CHI 2001.
- Guiard, Y, F. Buourgeois, D. Mottet, and M. Beaudouin-Lafon. 2001. Beyond the 10-bit Barrier: Fitts' Law in Multi-Scale Electronic Worlds. Paper read at IHM-HCI 2001, Sept., at Lille, France.
- Guiard, Y. 1987. Asymmetric Division of Labor in Human Skilled Bimanual Action: The Kinematic Chain as a Model. *The Journal of Motor Behavior* 19 (4):486-517.
- Guiard, Yves. 1987. Asymmetric division of labor in human skilled bimanual action: The kinematic chain as a model. *Journal of Motor Behavior* 19 (4):486-517.
- Guimbretiere, F., A. Martin, and T. Winograd. 2005. Benefits of merging command selection and direct manipulation. *ACM Trans. Comput.-Hum. Interact.* 12 (3):460-476.
- Gunawardana, Asela, Tim Paek, and Christopher Meek. 2010. Usability guided key-target resizing for soft keyboards. In *Proceeding of the 14th international conference on Intelligent user interfaces*. Hong Kong, China: ACM.
- Hancock, M., and S. Carpendale. 2007. Supporting Multiple Off-Axis Viewpoints at a Tabletop Display. Paper read at Horizontal Interactive Human-Computer Systems, 2007. TABLETOP '07. Second Annual IEEE International Workshop on, 10-12 Oct. 2007.
- Hancock, M. S., F. D. Vernier, D. Wigdor, S. Carpendale, and Shen Chia. 2006. Rotation and translation mechanisms for tabletop interaction. Paper read at Horizontal Interactive Human-Computer Systems, 2006. TableTop 2006. First IEEE International Workshop on, 5-7 Jan. 2006.
- Harrison, B., H. Ishii, K. Vicente, and W. Buxton. 1995. Transparent Layered User Interfaces: An Evaluation of a Display Design to Enhance Focused and Divided Attention. Paper read at Proceedings of CHI'95: ACM Conference on Human Factors in Computing Systems.
- Harrison, Beverly L., Kenneth P. Fishkin, Anuj Gujar, Carlos Mochon, and Roy Want. 1998. Squeeze me, hold me, tilt me! An exploration of manipulative user interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. Los Angeles, California, United States: ACM Press/Addison-Wesley Publishing Co.
- Hauptmann, A. 1989. Speech and Gestures for Graphic Image Manipulation. Paper read at Proceedings of CHI'89: ACM Conference on Human Factors in Computing Systems, Apr. 30 - May 4, at Austin, Texas.
- Hilliges, Otmar, Shahram Izadi, Andrew D. Wilson, Steve Hodges, Armando Garcia-Mendoza, and Andreas Butz. 2009. Interactions in the air: adding further depth to interactive tabletops. In *Proceedings of the 22nd annual ACM symposium on User interface software and technology*. Victoria, BC, Canada: ACM.
- Hinckley, K, K. Yatani, M. Pahud, N. Coddington, J. Rodenhouse, A. Wilson, H. Benko, and B. Buxton. 2010. Pen + Touch = New Tools. Paper read at UIST 2010 Symposium on User Interface Software & Technology, at New York, NY.
- Hinckley, K., M. Pahud, and B. Buxton. 2010. Direct Display Interaction via Simultaneous Pen + Multi-touch Input. Paper read at Society for Information Display SID 2010 Digest.
- Hinckley, K., R. Pausch, J. Goble, and N. Kassell. 1994. A Three-Dimensional User Interface for Neurosurgical Visualization. Paper read at Proc. SPIE Vol. 2164, Medical Imaging 1994: Image Capture, Formatting, and Display.
- Hinckley, K., J. Pierce, E. Horvitz, and M. Sinclair. 2005. Foreground and Background Interaction with Sensor-Enhanced Mobile Devices. *ACM TOCHI* 12 (1 (Special Issue on Sensor-Based Interaction)):31-52.
- Hinckley, K., J. Pierce, M. Sinclair, and E. Horvitz. 2000. Sensing Techniques for Mobile Interaction. Paper read at ACM UIST 2000 Symp. on User Interface Software & Technology, at San Diego, CA.
- Hinckley, K., G. Ramos, F. Guimbretiere, P. Baudisch, and M. Smith. 2004. Stitching: Pen Gestures that Span Multiple Displays. Paper read at ACM 7th International Working Conference on Advanced Visual Interfaces (AVI 2004), May 25-28, at Gallipoli (Leece), Italy.

- Hinckley, K., and M. Sinclair. 1999. Touch-Sensing Input Devices. Paper read at ACM CHI'99 Conf. on Human Factors in Computing Systems.
- Hinckley, K., and H. Song. 2010. Sensor Synaesthesia: Touch in Motion, and Motion In Touch. In *Under review*.
- Hinckley, K., S. Zhao, R. Sarin, P. Baudisch, E. Cutrell, M. Shilman, and D. Tan. 2007. InkSeine: In Situ Search for Active Note Taking. Paper read at CHI 2007.
- Hinckley, Ken, Patrick Baudisch, Gonzalo Ramos, and Francois Guimbretiere. 2005. Design and analysis of delimiters for selection-action pen gesture phrases in scriboli. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. Portland, Oregon, USA: ACM.
- Hinckley, Ken, Edward Cutrell, Steve Bathiche, and Tim Muss. 2002. Quantitative analysis of scrolling techniques. In *Proceedings of the SIGCHI conference on Human factors in computing systems: Changing our world, changing ourselves*. Minneapolis, Minnesota, USA: ACM.
- Hinckley, Ken, Mary Czerwinski, and Michael Sinclair. 1998. Interaction and modeling techniques for desktop two-handed input. Paper read at ACM UIST 1998 Symposium on User Interface Software and Technology, at New York, NY.
- Hinckley, Ken, Morgan Dixon, Raman Sarin, Francois Guimbretiere, and Ravin Balakrishnan. 2009. Codex: a dual screen tablet computer. In *Proceedings of the 27th international conference on Human factors in computing systems*. Boston, MA, USA: ACM.
- Hinckley, Ken, Francois Guimbretiere, Patrick Baudisch, Raman Sarin, Maneesh Agrawala, and Ed Cutrell. 2006. The springboard: multiple modes in one spring-loaded control. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*. Montréal, Québec, Canada: ACM.
- Hinckley, Ken, Randy Pausch, J. C. Goble, and Neal Kassell. 1994. A survey of design issues in spatial input. Paper read at ACM UIST 1994 Symposium on User Interface Software and Technology, at New York, NY.
- Hinckley, Ken, Randy Pausch, Dennis Proffitt, and Neal Kassell. 1998. Two-handed virtual manipulation. *ACM Transactions on Computer-Human Interaction* 5 (3):260-302.
- Hinckley, Ken, Randy Pausch, Dennis Proffitt, and Neal F. Kassell. 1998. Two-handed virtual manipulation. *ACM Trans. Comput.-Hum. Interact.* 5 (3):260-302.
- Hinckley, Ken, Jeff Pierce, Michael Sinclair, and Eric Horvitz. 2000. Sensing techniques for mobile interaction. Paper read at ACM UIST Symposium on User Interface Software and Technology, at New York, NY.
- Hinckley, Ken, Jeff Pierce, Mike Sinclair, and Eric Horvitz. 2000. Sensing techniques for mobile interaction. In *Proceedings of the 13th annual ACM symposium on User interface software and technology*. San Diego, California, United States: ACM.
- Hinckley, Ken, Koji Yatani, Michel Pahud, Nicole Coddington, Jenny Rodenhouse, Andy Wilson, Hrvoje Benko, and Bill Buxton. 2010. Pen + Touch = New Tools. In *Proceedings of the 23rd annual ACM symposium on User interface software and technology*. New York, New York, USA: ACM.
- Hinrichs, U., M. Hancock, C. Collins, and S. Carpendale. 2007. Examination of Text-Entry Methods for Tabletop Displays. Paper read at Horizontal Interactive Human-Computer Systems, 2007. TABLETOP '07. Second Annual IEEE International Workshop on, 10-12 Oct. 2007.
- Hirshfield, Leanne M., Krysta Chauncey, Rebecca Gulotta, Audrey Girouard, Erin T. Solovey, Robert J. Jacob, Angelo Sassaroli, and Sergio Fantini. 2009. Combining Electroencephalograph and Functional Near Infrared Spectroscopy to Explore Users' Mental Workload. In *Proceedings of the 5th International Conference on Foundations of Augmented Cognition. Neuroergonomics and Operational Neuroscience: Held as Part of HCI International 2009*. San Diego, CA: Springer-Verlag.
- Hirshfield, Leanne M., Erin Treacy Solovey, Audrey Girouard, James Kebinger, Robert J.K. Jacob, Angelo Sassaroli, and Sergio Fantini. 2009. Brain measurement for usability testing and adaptive interfaces: an example of uncovering syntactic workload with functional near infrared spectroscopy. In *Proceedings of the 27th international conference on Human factors in computing systems*. Boston, MA, USA: ACM.
- Holman, David, and Roel Vertegaal. 2008. Organic user interfaces: designing computers in any way, shape, or form. *Commun. ACM* 51 (6):48-55.
- Holz, Christian, and Patrick Baudisch. 2010. The generalized perceived input point model and how to double touch accuracy by extracting fingerprints. In *Proceedings of the 28th international conference on Human factors in computing systems*. Atlanta, Georgia, USA: ACM.
- Honan, M., E. Serina, R. Tal, and D. Rempel. 1995. Wrist Postures While Typing on a Standard and Split Keyboard. In *Proc. HFES Human Factors and Ergonomics Society 39th Annual Meeting*.
- Horvitz, E. 1999. Principles of Mixed-Initiative User Interfaces. Paper read at Proc. ACM CHI'99 Conf. on Human Factors in Computing Systems, at Pittsburgh, PA.

- Horvitz, E., A. Jacobs, and D. Hovel. 1999. Attention-Sensitive Alerting. Paper read at Proceedings of UAI '99, Conference on Uncertainty and Artificial Intelligence, July, at Stockholm, Sweden.
- Horvitz, Eric, Jack Breese, David Heckerman, David Hovel, and Koos Rommelse. 1998. The Lumiere Project: Bayesian User Modeling for Inferring the Goals and Needs of Software Users. Paper read at Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, Madison, WI, July 1998, pages 256-265. Morgan Kaufmann: San Francisco.
- Hutchings, Dugald Ralph, and John Stasko. 2004. Shrinking window operations for expanding display space. In *Proceedings of the working conference on Advanced visual interfaces*. Gallipoli, Italy: ACM.
- Hutchings, Dugald Ralph, and John Stasko. 2005. mudibo: multiple dialog boxes for multiple monitors. In *CHI '05 extended abstracts on Human factors in computing systems*. Portland, OR, USA: ACM.
- Ishii, Hiroshi, and Brygg Ullmer. 1997. Tangible bits: towards seamless interfaces between people, bits and atoms. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. Atlanta, Georgia, United States: ACM.
- Jacob, Robert J. K., Linda E. Sibert, Daniel C. McFarlane, and Jr. M. Preston Mullen. 1994. Integrality and separability of input devices. *ACM Trans. Comput.-Hum. Interact.* 1 (1):3-26.
- Jacob, Robert J.K., Audrey Girouard, Leanne M. Hirshfield, Michael S. Horn, Orit Shaer, Erin Treacy Solovey, and Jamie Zigelbaum. 2008. Reality-based interaction: a framework for post-WIMP interfaces. In *Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*. Florence, Italy: ACM.
- Jellinek, H., and S. Card. 1990. Powermice and User Performance. Paper read at Proc. ACM CHI'90 Conf. on Human Factors in Computing Systems.
- Jellinek, Herbert D., and Stuart K. Card. 1990. Powermice and user performance. In *Proceedings of the SIGCHI conference on Human factors in computing systems: Empowering people*. Seattle, Washington, United States: ACM.
- John, B. E., and D. Kieras. 1996. Using GOMS for User Interface Design and Evaluation: Which Technique? *ACM Trans. Comput.-Hum. Interact.* 3 (4):287-319.
- Jojic, N., B. Brumitt, B. Meyers, and S. Harris. 2000. Detecting and estimating pointing gestures in dense disparity maps. Paper read at Proceed. of IEEE Intl. Conf. on Automatic Face and Gesture Recognition.
- Kabbash, Paul, and William A. S. Buxton. 1995. The "prince"; technique: Fitts' law and selection using area cursors. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. Denver, Colorado, United States: ACM Press/Addison-Wesley Publishing Co.
- Kabbash, Paul, William Buxton, and Abigail Sellen. 1994. Two-handed input in a compound task. In *Proceedings of the SIGCHI conference on Human factors in computing systems: celebrating interdependence*. Boston, Massachusetts, United States: ACM.
- Kamba, Tomonari, Shawn A. Elson, Terry Harpold, Tim Stamper, and Piyawadee Sukaviriya. 1996. Using small screen space more efficiently. Paper read at Conference proceedings on Human factors in computing systems.
- Karat, Clare-Marie, Christine Halverson, Daniel Horn, and John Karat. 1999. Patterns of entry and correction in large vocabulary continuous speech recognition systems. In *Proceedings of the SIGCHI conference on Human factors in computing systems: the CHI is the limit*. Pittsburgh, Pennsylvania, United States: ACM.
- Karl, L., M. Pettey, and B. Shneiderman. 1993. Speech-Activated versus Mouse-Activated Commands for Word Processing Applications: An Empirical Evaluation. *International Journal of Man-Machine Studies* 39 (4):667-687.
- Khan, Azam, Justin Matejka, George Fitzmaurice, and Gordon Kurtenbach. 2005. Spotlight: directing users' attention on large displays. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. Portland, Oregon, USA: ACM.
- Kirsh, D., and P. Maglio. 1994. On distinguishing epistemic from pragmatic action. *Cognitive Science* 18 (4):513-549.
- Kirsh, David. 1995. The intelligent use of space. *Artificial Intelligence* 73:p. 31-68.
- Klemmer, Scott R., Mark W. Newman, Ryan Farrell, Mark Bilezikjian, and James A. Landay. 2001. The designers' outpost: a tangible interface for collaborative web site. In *Proceedings of the 14th annual ACM symposium on User interface software and technology*. Orlando, Florida: ACM.
- Ko, Byong K., and Hyun S. Yang. 1997. Finger mouse and gesture recognition system as a new human computer interface. *Comput. Graph.* 21 (5):555-561.
- Kristensson, Per-Ola, and Shumin Zhai. 2004. SHARK 2: a large vocabulary shorthand writing system for pen-based computers. In *Proceedings of the 17th annual ACM symposium on User interface software and technology*. Santa Fe, NM, USA: ACM.
- Krueger, M., T. Gionfriddo, and K. Hinrichsen. 1985. VIDEOPLACE--An Artificial Reality. Paper read at Proceedings of CHI'85: ACM Conference on Human Factors in Computing Systems, Apr. 14-18, at San Francisco, Calif.

- Krueger, Myron. 1991. VIDEOPLACE and the interface of the future. In *The art of human computer interface design*, edited by B. Laurel. Menlo Park, CA: Addison Wesley.
- Krumm, J., and K. Hinckley. 2004. The NearMe Wireless Proximity Server. Paper read at Ubicomp 2004.
- Kurtenbach, G., and W. Buxton. 1991. GEdit: a testbed for editing by contiguous gesture. *SIGCHI Bulletin* 23 (2):22-26.
- Kurtenbach, G., A. Sellen, and W. Buxton. 1993. An empirical evaluation of some articulatory and cognitive aspects of 'marking menus'. *J. Human Computer Interaction* 8 (1).
- Kurtenbach, Gordon, and Bill Buxton. 1991. GEdit: a test bed for editing by contiguous gestures. *SIGCHI Bull.* 23 (2):22-26.
- Kurtenbach, Gordon, and William Buxton. 1991. Issues in combining marking and direct manipulation techniques. In *Proceedings of the 4th annual ACM symposium on User interface software and technology*. Hilton Head, South Carolina, United States: ACM.
- Kurtenbach, Gordon, George Fitzmaurice, Thomas Baudel, and Bill Buxton. 1997. The design of a GUI paradigm based on tablets, two-hands, and transparency. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. Atlanta, Georgia, United States: ACM.
- Lank, E., and E. Saund. 2005. Sloppy Selection: Providing an Accurate Interpretation of Imprecise Selection Gestures. *Computers and Graphics* 29 (4):490 - 500.
- Lansdale, M., and E. Edmonds. 1992. Using memory for events in the design of personal filing systems. *Int. J. Man-Mach. Studies* 36 (1):97-126.
- LaViola, J. and Zeleznik, R. 2004. MathPad2: A System for the Creation and Exploration of Mathematical Sketches. *ACM Transactions on Graphics* 23 (3):432-440.
- Lepinski, G. Julian, Tovi Grossman, and George Fitzmaurice. 2010. The design and evaluation of multitouch marking menus. In *Proceedings of the 28th international conference on Human factors in computing systems*. Atlanta, Georgia, USA: ACM.
- Levine, S., and S. Ehrlich. 1991. The Freestyle system: a design perspective. In *Human-Machine Interactive Systems*, edited by A. Klinger. New York: Plenum Press.
- Lewis, J., K. Potosnak, and R. Magyar. 1997. Keys and Keyboards. In *Handbook of Human-Computer Interaction*, edited by M. Helander, T. Landauer and P. Prabhu. Amsterdam: North-Holland.
- Li, Yang, Ken Hinckley, Zhiwei Guan, and James A. Landay. 2005. Experimental analysis of mode switching techniques in pen-based user interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. Portland, Oregon, USA: ACM.
- Lipscomb, J, and M. Pique. 1993. Analog Input Device Physical Characteristics. *SIGCHI Bulletin* 25 (3):40-45.
- Lyons, Kent, Christopher Skeels, Thad Starner, Cornelis M. Snoeck, Benjamin A. Wong, and Daniel Ashbrook. 2004. Augmenting Conversations Using Dual-Purpose Speech. Paper read at UIST'04 Symposium on User Interface Software & Technology, at Sante Fe, NM.
- Mackay, W. E. 2002. Which Interaction Technique Works When? Floating Palettes, Marking Menus and Toolglasses Support Different Task Strategies. Paper read at Proc. AVI 2002 International Conference on Advanced Visual Interfaces.
- MacKenzie, I. S. 1989. A note on the information-theoretic basis for Fitts' Law. *Journal of Motor Behavior* 21:323-330.
- MacKenzie, I. S. 1992. Fitts' law as a research and design tool in human-computer interaction. *Human-Computer Interaction* 7:91-139.
- MacKenzie, I. S. 1995. Input Devices and Interaction Techniques for Advanced Computing. In *Virtual environments and advanced interface design*, edited by W. Barfield and T. Furness. Oxford, UK: Oxford University Press.
- MacKenzie, I. S., and A. Oniszczak. 1998. A Comparison of Three Selection Techniques for Touchpads. Paper read at Proc. ACM CHI'98 Conf. on Human Factors in Computing Systems.
- MacKenzie, I. S., A. Sellen, and W. Buxton. 1991. A Comparison of Input Devices in Elemental Pointing and Dragging Tasks. Paper read at Proc. ACM CHI '91 Conf. on Human Factors in Computing Systems.
- MacKenzie, I. S., and R. W. Soukoreff. 2002. A model of two-thumb text entry. In *Proceedings of Graphics Interface*. Toronto: Canadian Information Processing Society.
- MacKenzie, I. S., and C. Ware. 1993. Lag as a Determinant of Human Performance in Interactive Systems. Paper read at Proc. ACM INTERCHI'93 Conference on Human Factors in Computing Systems.
- MacKenzie, I. Scott. 1992. Movement time prediction in human-computer interfaces. In *Proceedings of the conference on Graphics interface '92*. Vancouver, British Columbia, Canada: Morgan Kaufmann Publishers Inc.
- MacKenzie, I. Scott, and Yves Guiard. 2001. The two-handed desktop interface: are we there yet? In *CHI '01 extended abstracts on Human factors in computing systems*. Seattle, Washington: ACM.
- MacKenzie, I. Scott, Hedy Kober, Derek Smith, Terry Jones, and Eugene Skepner. 2001. LetterWise: prefix-based disambiguation for mobile text input. In *Proceedings of the 14th annual ACM symposium on User interface software*

- and technology*. Orlando, Florida: ACM.
- MacKenzie, I. Scott, and Shawn X. Zhang. 1997. The immediate usability of graffiti. In *Proceedings of the conference on Graphics interface '97*. Kelowna, British Columbia, Canada: Canadian Information Processing Society.
- Mackinlay, Jock D., and Jeffrey Heer. 2004. Wideband displays: mitigating multiple monitor seams. In *CHI '04 extended abstracts on Human factors in computing systems*. Vienna, Austria: ACM.
- Maes, P., T. Darrell, B. Blumberg, and A. Pentland. 1997. The ALIVE system: wireless, full-body interaction with autonomous agents. *ACM Multimedia Systems* (Special Issue on Multimedia and Multisensory Virtual Worlds).
- Malik, Shahzad, Abhishek Ranjan, and Ravin Balakrishnan. 2005. Interacting with large displays from a distance with vision-tracked multi-finger gestural input. In *Proceedings of the 18th annual ACM symposium on User interface software and technology*. Seattle, WA, USA: ACM.
- Marklin, R., G. Simoneau, and J. Monroe. 1997. The Effect of Split and Vertically-Inclined Computer Keyboards on Wrist and Forearm Posture. In *Proc. HFES Human Factors and Ergonomics Society 41st Annual Meeting*.
- Matsushita, Mitsunori, Makoto Iida, Takeshi Ohguro, Yoshinari Shirai, Yasuaki Kakehi, and Takeshi Naemura. 2004. Lumisight table: a face-to-face collaboration support system that optimizes direction of projected information to each stakeholder. In *Proceedings of the 2004 ACM conference on Computer supported cooperative work*. Chicago, Illinois, USA: ACM Press.
- Matsushita, Nobuyuki, and Jun Rekimoto. 1997. HoloWall: designing a finger, hand, body, and object sensitive wall. In *Proceedings of the 10th annual ACM symposium on User interface software and technology*. Banff, Alberta, Canada: ACM.
- McCallum, David C., and Pourang Irani. 2009. ARC-Pad: absolute+relative cursor positioning for large displays with a mobile touchscreen. In *Proceedings of the 22nd annual ACM symposium on User interface software and technology*. Victoria, BC, Canada: ACM.
- McLoone, H., K. Hinckley, and E. Cutrell. 2003. Bimanual Interaction on the Microsoft Office Keyboard. In *INTERACT 2003*.
- Microsoft Corp. 2002. Windows XP Pointer Ballistics. <http://www.microsoft.com/whdc/device/input/pointer-bal.msp>.
- Microsoft Courier. 2010. In *Wikipedia the Free Encyclopedia*.
- Mine, M., F. Brooks, and C. Sequin. 1997. Moving objects in space: exploiting proprioception in virtual environment interaction. In *ACM SIGGRAPH 1997 Conference on Computer Graphics and Interactive Techniques*. New York, NY: ACM.
- Moran, Thomas P., Patrick Chiu, and William van Melle. 1997. Pen-based interaction techniques for organizing material on an electronic whiteboard. In *Proceedings of the 10th annual ACM symposium on User interface software and technology*. Banff, Alberta, Canada: ACM.
- Morris, M. R., A. J. B. Brush, and B. R. Meyers. 2008. A field study of knowledge workers' use of interactive horizontal displays. Paper read at Horizontal Interactive Human Computer Systems, 2008. TABLETOP 2008. 3rd IEEE International Workshop on, 1-3 Oct. 2008.
- Morris, Meredith Ringel, Andreas Paepcke, Terry Winograd, and Jeannie Stamberger. 2006. TeamTag: exploring centralized versus replicated controls for co-located tabletop groupware. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*. Montréal, Québec, Canada: ACM.
- Moscovich, Tomer. 2009. Contact area interaction with sliding widgets. In *Proceedings of the 22nd annual ACM symposium on User interface software and technology*. Victoria, BC, Canada: ACM.
- Moscovich, Tomer, and John F. Hughes. 2006. Multi-finger cursor techniques. In *Proceedings of Graphics Interface 2006*. Quebec, Canada: Canadian Information Processing Society.
- Myers, B., H. Stiel, and R. Gargiulo. 1998. Collaboration Using Multiple PDAs Connected to a PC. Paper read at Proc. ACM CSCW'98 Conf. on Computer Supported Cooperative Work, Nov. 14-18, at Seattle, WA.
- Myers, Brad A., Kin Pou Lie, and Bo-Chieh Yang. 2000. Two-handed input using a PDA and a mouse. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. The Hague, The Netherlands: ACM.
- Nacenta, Miguel A., Dzmityr Aliakseyeu, Sriram Subramanian, and Carl Gutwin. 2005. A comparison of techniques for multi-display reaching. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. Portland, Oregon, USA: ACM.
- Ni, Tao, and Patrick Baudisch. 2009. Disappearing mobile devices. In *Proceedings of the 22nd annual ACM symposium on User interface software and technology*. Victoria, BC, Canada: ACM.
- Nickel, Kai, and Rainer Stiefelhagen. 2003. Pointing gesture recognition based on 3D-tracking of face, hands and head orientation. In *Proceedings of the 5th international conference on Multimodal interfaces*. Vancouver, British Columbia, Canada: ACM.
- Norman, D. 1990. *The Design of Everyday Things*: Doubleday: New York, NY.
- O'Hara, K., and A. Sellen. 1997. A comparison of reading paper and on-line documents. Paper read at CHI'97.

- Oakley, Ian, Stephen Brewster, and Philip Gray. 2001. Solving multi-target haptic problems in menu interaction. In *CHI '01 extended abstracts on Human factors in computing systems*. Seattle, Washington: ACM.
- Olson, Judith Reitman, and Gary M. Olson. 1990. The growth of cognitive modeling in human-computer interaction since GOMS. *Hum.-Comput. Interact.* 5 (2):221-265.
- Oviatt, S., A. O. Cohen, and A. J. Mann. 2010. Designing Interfaces that Stimulate Ideational Super-fluency in Science (in review).
- Oviatt, Sharon, Antonella DeAngeli, and Karen Kuhn. 1997. Integration and synchronization of input modes during multimodal human-computer interaction. In *Referring Phenomena in a Multimedia Context and their Computational Treatment*. Madrid, Spain: Association for Computational Linguistics.
- Pederson, E., K. McCall, Tom Moran, and F. Halasz. 1993. Tivoli: An electronic whiteboard for informal workgroup meetings. Paper read at ACM CHI Conference on Human Factors in Computing Systems, at New York.
- Potter, R. L., L. J. Weldon, and B. Shneiderman. 1988. Improving the accuracy of touch screens: an experimental evaluation of three strategies. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. Washington, D.C., United States: ACM.
- Poupyrev, Ivan, and Shigeaki Maruyama. 2003. Tactile interfaces for small touch screens. In *Proceedings of the 16th annual ACM symposium on User interface software and technology*. Vancouver, Canada: ACM.
- Price, M. N., B. Schilit, and G. Golovchinsky. 1998. XLibris: The Active Reading Machine. Paper read at CHI'98 Extended Abstracts.
- Putz-Anderson, V. 1988. *Cumulative trauma disorders: A manual for musculoskeletal diseases of the upper limbs*. Bristol, PA: Taylor & Francis.
- Ramos, G., and R. Balakrishnan. 2005. Zliding: Fluid Zooming and Sliding for High Precision Parameter Manipulation. Paper read at UIST 2006.
- Ramos, G., and R. Balakrishnan. 2006. Pressure Marks. Paper read at UNPUBLISHED MANUSCRIPT (under review).
- Ramos, G., M. Boulous, and R. Balakrishnan. 2004. Pressure Widgets. Paper read at CHI 2004.
- Raskar, R., P. Beardsley, J. van Baar, Y. Wang, P. Dietz, J. Lee, D. Leigh, and T. Willwacher. 2004. RFIG lamps: interacting with a self-describing world via photosensing wireless tags and projectors. 23 (3): 406-415.
- Raskar, Ramesh, Jeroen van Baar, Paul Beardsley, Thomas Willwacher, Srinivas Rao, and Clifton Forlines. 2005. iLamps: geometrically aware and self-configuring projectors. In *ACM SIGGRAPH 2005 Courses*. Los Angeles, California: ACM.
- Raskin, J. 2000. *The Humane Interface: New Directions for Designing Interactive Systems*: ACM Press.
- Rekimoto, J., Y. Ayatsuka, M. Kohno, and Haruo Oba. 2003. Proximal Interactions: A Direct Manipulation Technique for Wireless Networking. Paper read at INTERACT 2003.
- Rekimoto, Jun. 1996. Tilting operations for small screen interfaces. Paper read at ACM UIST Symposium on User Interface Software and Technology, at New York.
- Rekimoto, Jun. 1998. A multiple-device approach for supporting whiteboard based interaction. Paper read at ACM CHI Conference on Human Factors in Computing Systems, at New York.
- Rempel, D, J Bach, L Gordon, and R. Tal. 1998. Effects of forearm pronation/supination on carpal tunnel pressure. *J Hand Surgery* 23 (1):38-42.
- Rime, B., and L. Schiaratura. 1991. Gesture and speech. In *Fundamentals of Nonverbal Behaviour*. New York: Press Syndacate of the University of Cambridge.
- Rutledge, J., and T. Selker. 1990. Force-to-Motion Functions for Pointing. Paper read at Proceedings of Interact '90: The IFIP Conf. on Human-Computer Interaction, Aug. 27-31.
- Rutledge, Joseph D., and Ted Selker. 1990. Force-to-motion functions for pointing. In *Proceedings of the IFIP TC13 Third International Conference on Human-Computer Interaction*: North-Holland Publishing Co.
- Ryall, Kathy, Clifton Forlines, Chia Shen, and Meredith Ringel Morris. 2004. Exploring the effects of group size and table size on interactions with tabletop shared-display groupware. In *Proceedings of the 2004 ACM conference on Computer supported cooperative work*. Chicago, Illinois, USA: ACM.
- Saponas, T. Scott, Desney S. Tan, Dan Morris, Ravin Balakrishnan, Jim Turner, and James A. Landay. 2009. Enabling always-available input with muscle-computer interfaces. In *Proceedings of the 22nd annual ACM symposium on User interface software and technology*. Victoria, BC, Canada: ACM.
- Saund, Eric, and Edward Lank. 2003. Stylus input and editing without prior selection of mode. In *Proceedings of the 16th annual ACM symposium on User interface software and technology*. Vancouver, Canada: ACM.
- Sawhney, N., and Chris M. Schmandt. 2000. Nomadic Radio: Speech and Audio Interaction for Contextual Messaging in Nomadic Environments. *ACM Transactions on Computer-Human Interaction* 7 (3):353-383.

- Schilit, Bill N., Norman I. Adams, and Roy Want. 1994. Context-Aware Computing Applications. Paper read at Proc. IEEE Workshop on Mobile Computing Systems and Applications, Dec., at Santa Cruz, CA.
- Schmandt, Chris M., N. Marmasse, S. Marti, N. Sawhney, and S. Wheeler. 2000. Everywhere messaging. *IBM Systems Journal* 39 (3&4).
- Schmidt, Albrecht, Michael Beigl, and Hans-Werner Gellersen. 1999. There is more to context than location. *Computers and Graphics* 23 (6):893-901.
- Schwarz, Julia, Scott Hudson, Jennifer Mankoff, and Andrew D. Wilson. 2010. A framework for robust and flexible handling of inputs with uncertainty. In *Proceedings of the 23rd annual ACM symposium on User interface software and technology*. New York, New York, USA: ACM.
- Scott, Stacey D., M. Sheelagh, T. Carpendale, and Kori M. Inkpen. 2004. Territoriality in collaborative tabletop workspaces. In *Proceedings of the 2004 ACM conference on Computer supported cooperative work*. Chicago, Illinois, USA: ACM.
- Sears, A. 1991. Improving touchscreen keyboards: design issues and a comparison with other devices. *Interacting with Computers* 3 (3):253-269.
- Sears, A. 1993. Investigating touchscreen typing: the effect of keyboard size on typing speed. *Behaviour & Information Technology* 12 (1):17-22.
- Sears, Andrew, and Ben Shneiderman. 1991. High Precision Touchscreens: Design Strategies and Comparisons with a Mouse. *International Journal of Man-Machine Studies* 34 (4):593-613.
- Sellen, A., G. Kurtenbach, and W. Buxton. 1992. The prevention of mode errors through sensory feedback. *Human Computer Interaction* 7 (2):141-164.
- Seow, Steven C. 2008. *Designing and Engineering Time*. 1 ed. Boston: Addison-Wesley.
- Shell, Jeffrey S., Roel Vertegaal, Daniel Cheng, Alexander W. Skaburskis, Changuk Sohn, A. James Stewart, Omar Aoudeh, and Connor Dickie. 2004. ECSSGlasses and EyePliances: using attention to open sociable windows of interaction. In *Proceedings of the 2004 symposium on Eye tracking research & applications*. San Antonio, Texas: ACM.
- Shen, Chia, K. Everitt, and K. Ryall. 2003. UbiTable: Impromptu Face-to-Face Collaboration on Horizontal Interactive Surfaces. Paper read at UbiComp 2003.
- Shen, Chia, Franck D. Vernier, Clifton Forlines, and Meredith Ringel. 2004. DiamondSpin: an extensible toolkit for around-the-table interaction In *Proceedings of the 2004 conference on Human factors in computing systems* Vienna, Austria ACM Press.
- Smutz, W, E Serina, T Bloom, and D Rempel. 1994. A System for Evaluating the Effect of Keyboard Design on Force, Posture, Comfort, and Productivity. *Ergonomics* 37 (10):1649-1660.
- Snibbe, S., and K. MacLean. 2001. Haptic Techniques for Media Control. *CHI Letters (Proc. UIST 2001)* 3 (2):199-208.
- Sommer, R. 1965. Further studies of small group ecology. *Sociometry* 28:337-348.
- Song, Hyunyoung, Tovi Grossman, George Fitzmaurice, Franck D. Vernier, Francois Guimbretiere, Azam Khan, Ramtin Attar, and Gordon Kurtenbach. 2009. PenLight: combining a mobile projector and a digital pen for dynamic visual overlay. In *Proceedings of the 27th international conference on Human factors in computing systems*. Boston, MA, USA: ACM.
- Song, Hyunyoung, Francois Guimbretiere, Tovi Grossman, and George Fitzmaurice. 2010. MouseLight: bimanual interactions on digital paper using a pen and a spatially-aware mobile projector. In *Proceedings of the 28th international conference on Human factors in computing systems*. Atlanta, Georgia, USA: ACM.
- Stifelman, Lisa J. 1996. Augmenting real-world objects: a paper-based audio notebook. In *Conference companion on Human factors in computing systems: common ground*. Vancouver, British Columbia, Canada: ACM.
- Streitz, Norbert A., J. Geuler, Torsten Holmer, Shin'ichi Konomi, Christian Miller-Tomfelde, Wolfgang Reischl, Petra Rexroth, Peter Seitz, and Ralf Steinmetz. 1999. i-LAND: an interactive landscape for creativity and innovation. In *Proceedings of the SIGCHI conference on Human factors in computing systems: the CHI is the limit*. Pittsburgh, Pennsylvania, United States: ACM Press.
- Surati, Rajeev J. 1999. Scalable self-calibrating display technology for seamless large-scale displays, Massachusetts Institute of Technology.
- Sutherland, Ivan E. 1964. Sketch pad a man-machine graphical communication system. In *Proceedings of the SHARE design automation workshop*: ACM.
- Swaminathan, K., and S. Sato. 1997. Interaction Design for Large Displays. *interactions* (Jan-Feb).
- Tan, Desney S, and Mary Czerwinski. 2003. Effects of Visual Separation and Physical Discontinuities when Distributing Information across Multiple Displays In *OZCHI 2003 Conference for the Computer-Human Interaction Special Interest Group of the Ergonomics Society of Australia*.
- Tan, Desney S., Darren Gergle, Peter Scupelli, and Randy Pausch. 2003. With similar visual angles, larger displays improve spatial performance. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. Ft.

- Lauderdale, Florida, USA: ACM.
- Tang, Anthony, Michel Pahud, Kori Inkpen, Hrvoje Benko, John C. Tang, and Bill Buxton. 2010. Three's company: understanding communication channels in three-way distributed collaboration. In *Proceedings of the 2010 ACM conference on Computer supported cooperative work*. Savannah, Georgia, USA: ACM.
- Tilbrook, D.M. 1976. A Newspaper Pagination System, Department of Computer Science, University of Toronto, Toronto.
- Toney, Aaron, and H. Thomas Bruce. 2006. Considering Reach in Tangible and Table Top Design.
- Trimble, Jay, Roxana Wales, and Rich Gossweiler. 2003. NASA's MERBoard. In *Public and Situated Displays: Social and interactional aspects of shared display technologies*, edited by K. O'Hara, M. Perry, E. Churchill and D. Russell: Kluwer.
- Tse, Edward, Saul Greenberg, Chia Shen, Clifton Forlines, and Ryo Kodama. 2008. Exploring true multi-user multimodal interaction over a digital table. In *Proceedings of the 7th ACM conference on Designing interactive systems*. Cape Town, South Africa: ACM.
- Ullmer, Brygg, and Hiroshi Ishii. 1997. The metaDESK: models and prototypes for tangible user interfaces. In *Proceedings of the 10th annual ACM symposium on User interface software and technology*. Banff, Alberta, Canada: ACM.
- Underkoffler, John, and Hiroshi Ishii. 1998. Illuminating light: an optical design tool with a luminous-tangible interface. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. Los Angeles, California, United States: ACM Press/Addison-Wesley Publishing Co.
- Villar, Nicolas, Shahram Izadi, Dan Rosenfeld, Hrvoje Benko, John Helmes, Jonathan Westhues, Steve Hodges, Eyal Ofek, Alex Butler, Xiang Cao, and Billy Chen. 2009. Mouse 2.0: multi-touch meets the mouse. In *Proceedings of the 22nd annual ACM symposium on User interface software and technology*. Victoria, BC, Canada: ACM.
- Viviani, P., and T. Flash. 1995. Minimum-jerk, two-thirds power law and isochrony: converging approaches to the study of movement planning. *Journal of Experimental Psychology: Perception and Performance* 21:32-53.
- Vogel, D., and P. Baudisch. 2007. Shift: A Technique for Operating Pen-Based Interfaces Using Touch. Paper read at CHI 2007.
- Vogel, Daniel, and Ravin Balakrishnan. 2005. Distant freehand pointing and clicking on very large, high resolution displays. In *Proceedings of the 18th annual ACM symposium on User interface software and technology*. Seattle, WA, USA: ACM.
- Vogel, Daniel, and Patrick Baudisch. 2007. Shift: a technique for operating pen-based interfaces using touch. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. San Jose, California, USA: ACM.
- Want, R., and G. Borriello. 2000. Survey on Information Appliances. *IEEE Personal Communications* (May/June):24-31.
- Weiser, M. 1991. The Computer for the 21st Century. *Scientific American* (September):94-104.
- Wickens, Chris. 1992. *Engineering psychology and human performance*. Upper Saddle River, NJ: Prentice-Hall.
- Wigdor, D., C. Forlines, P. Baudisch, J. Barnwell, and C. Shen. 2007. LucidTouch: A See-Through Mobile Device. Paper read at UIST 2007
- Wigdor, Daniel, and Ravin Balakrishnan. 2003. TiltText: using tilt for text input to mobile phones. In *Proceedings of the 16th annual ACM symposium on User interface software and technology*. Vancouver, Canada: ACM.
- Wigdor, Daniel, Clifton Forlines, Patrick Baudisch, John Barnwell, and Chia Shen. 2007. Lucid touch: a see-through mobile device. In *Proceedings of the 20th annual ACM symposium on User interface software and technology*. Newport, Rhode Island, USA: ACM.
- Wigdor, Daniel, Chia Shen, Clifton Forlines, and Ravin Balakrishnan. 2007. Perception of elementary graphical elements in tabletop and multi-surface environments. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. San Jose, California, USA: ACM.
- Wilson, A. 2005. PlayAnywhere: A Compact Interactive Tabletop Projection-Vision System.
- Wilson, Andrew D. 2007. Depth-Sensing Video Cameras for 3D Tangible Tabletop Interaction. In *Second Annual IEEE International Workshop on Horizontal Interactive Human-Computer Systems (TABLETOP'07)*.
- Wilson, Andrew D. 2009. Simulating grasping behavior on an imaging interactive surface. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*. Banff, Alberta, Canada: ACM.
- Wilson, Andrew D., and Hrvoje Benko. 2010. Combining multiple depth cameras and projectors for interactions on, above and between surfaces. In *Proceedings of the 23rd annual ACM symposium on User interface software and technology*. New York, New York, USA: ACM.
- Wilson, Andrew D., Shahram Izadi, Otmar Hilliges, Armando Garcia-Mendoza, and David Kirk. 2008. Bringing physics to the surface. In *Proceedings of the 21st annual ACM symposium on User interface software and technology*. Monterey, CA, USA: ACM.
- Wilson, Andrew, and S. Shafer. 2003. XWand: UI for intelligent spaces. Paper read at ACM CHI 2003 Conference on Human Factors in Computing Systems, at New York.

- Wilson, Andy, and Nuria Oliver. 2003. GWindows: Towards Robust Perception-Based UI. Paper read at Proc. of CVPR 2003 (Workshop on Computer Vision for HCI).
- Wilson, F. R. 1998. *The Hand: How its use shapes the brain, language, and human culture*. New York: Pantheon Books.
- Wobbrock, Jacob O., Meredith Ringel Morris, and Andrew D. Wilson. 2009. User-defined gestures for surface computing. In *Proceedings of the 27th international conference on Human factors in computing systems*. Boston, MA, USA: ACM.
- Wu, Michael, and Ravin Balakrishnan. 2003. Multi-finger and whole hand gestural interaction techniques for multi-user tabletop displays. Paper read at UIST 2003, at Vancouver, BC, Canada.
- Yamada, H. 1980. A historical study of typewriters and typing methods: from the position of planning Japanese parallels. *J. Information Processing* 24 (4):175-202.
- Yee, K-P. 2003. Peephole Displays: Pen Interaction on Spatially Aware Handheld Computers. Paper read at CHI 2003.
- Yee, K-P., K. Swearingen, K. Li, and M. Hearst. 2003. Faceted metadata for image search and browsing. Paper read at CHI 2003.
- Zelevnik, Robert, Andrew Bragdon, Ferdi Adeptura, and Hsu-Sheng Ko. 2010. Hands-on math: a page-based multi-touch and pen desktop for technical work and problem solving. In *Proceedings of the 23rd annual ACM symposium on User interface software and technology*. New York, New York, USA: ACM.
- Zelevnik, Robert, Andrew Bragdon, Hsu-Sheng Ko, and Ferdi Adeptura. 2010. Hands-On Math: A page-based multi-touch and pen desktop for technical work and problem solving. Paper read at UIST 2010 Symposium on User Interface Software & Technology.
- Zhai, S. 1993. Human Performance Evaluation of Manipulation Schemes in Virtual Environments. Paper read at Proc. IEEE Virtual Reality International Symposium (VRAIS'93), at Seattle.
- Zhai, S., W. Buxton, and P. Milgram. 1996. The Partial Occlusion Effect: Utilizing Semi-transparency for Human Computer Interaction. *ACM Transactions on Computer-Human Interaction* 3 (3):254-284.
- Zhai, S., and P. Kristensson. 2003. Shorthand writing on stylus keyboard. Paper read at CHI '03 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems April 05 - 10, at Ft. Lauderdale, Florida, USA.
- Zhai, S., P. Kristensson, P. Gong, M. Greiner, S. A. Peng, L. M. Liu, and A. Dunnigan. 2009. Shapewriter on the iphone: from the laboratory to the real world. Paper read at CHI '09 *Proceedings of the 27th international Conference Extended Abstracts on Human Factors in Computing Systems*, April 04 - 09, at Boston, MA, USA.
- Zhai, S., and P. Milgram. 1993. Human Performance Evaluation of Isometric and Elastic Rate Controllers in a 6DoF Tracking Task. Paper read at Proc. SPIE Telem manipulator Technology.
- Zhai, S., B. A. Smith, and T. Selker. 1997. Improving Browsing Performance: A study of four input devices for scrolling and pointing tasks. Paper read at Proc. INTERACT97: The Sixth IFIP Conf. on Human-Computer Interaction.
- Zhai, Shumin. 1998. User performance in relation to 3D input device design. *SIGGRAPH Comput. Graph.* 32 (4):50-54.
- Zhai, Shumin, Paul Milgram, and William Buxton. 1996. The influence of muscle groups on performance of multiple degree-of-freedom input. Paper read at ACM CHI 1996 Conference on Human Factors in Computing Systems, at New York, NY.
- Zhai, Shumin, Barton A. Smith, and Ted Selker. 1997. Improving Browsing Performance: A study of four input devices for scrolling and pointing tasks. In *Proceedings of the IFIP TC13 Interantional Conference on Human-Computer Interaction*: Chapman & Hall, Ltd.
- Zhao, Shengdong, and Ravin Balakrishnan. 2004. Simple vs. compound mark hierarchical marking menus. In *Proceedings of the 17th annual ACM symposium on User interface software and technology*. Santa Fe, NM, USA: ACM.