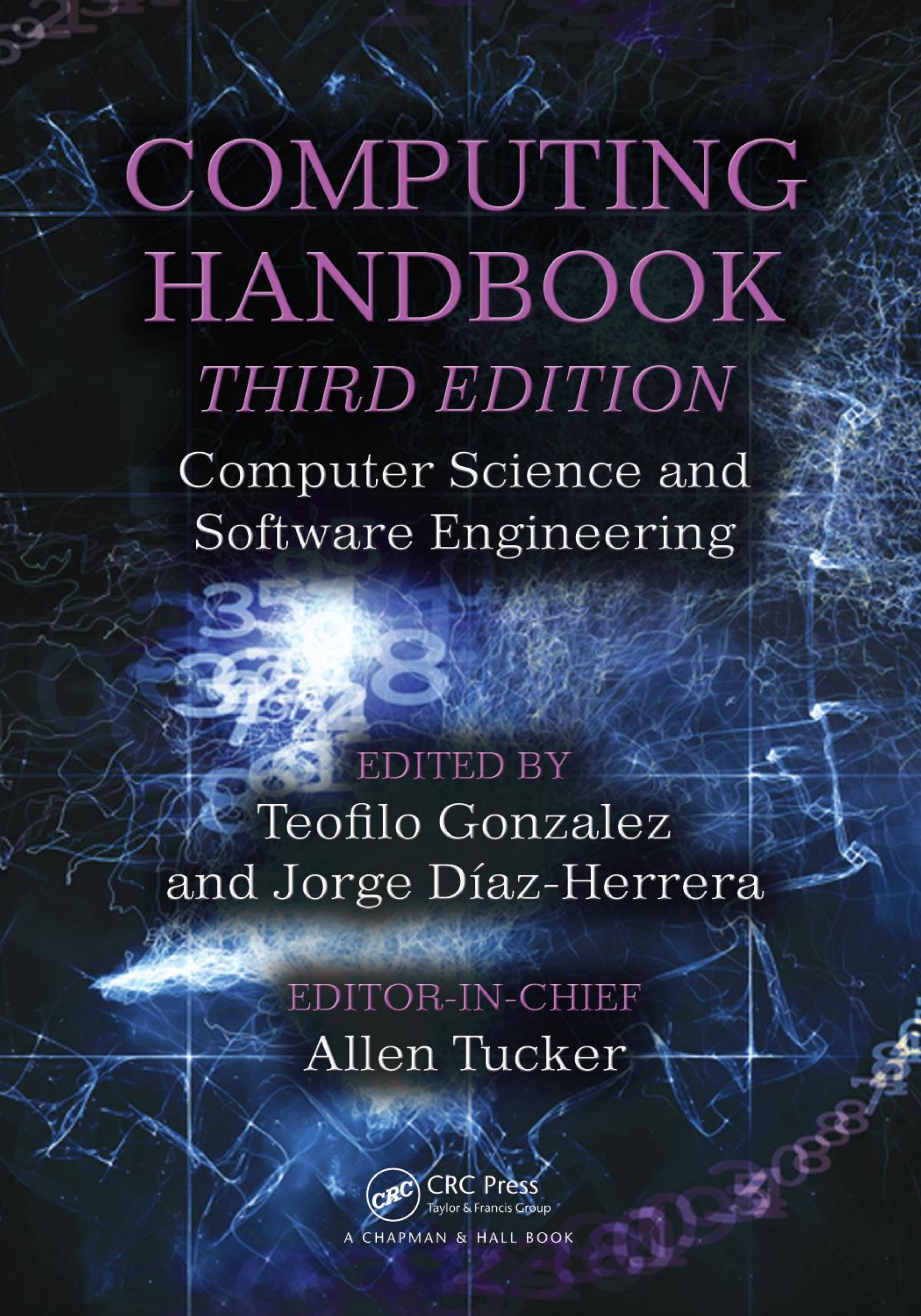

COMPUTING HANDBOOK

THIRD EDITION

Computer Science and Software Engineering

EDITED BY
Teofilo Gonzalez
and Jorge Díaz-Herrera

EDITOR-IN-CHIEF
Allen Tucker

CRC Press
Taylor & Francis Group
A CHAPMAN & HALL BOOK

COMPUTING HANDBOOK

THIRD EDITION

Computer Science and Software Engineering

EDITED BY

Teofilo Gonzalez
University of California
Santa Barbara, California, USA

Jorge Díaz-Herrera
Keuka College
Keuka Park, New York, USA

EDITOR-IN-CHIEF

Allen Tucker
Bowdoin College
Brunswick, Maine, USA

CRC Press
Taylor & Francis Group
Boca Raton London New York

CRC Press is an imprint of the
Taylor & Francis Group, an **informa** business
A CHAPMAN & HALL BOOK

MATLAB® is a trademark of The MathWorks, Inc. and is used with permission. The MathWorks does not warrant the accuracy of the text or exercises in this book. This book's use or discussion of MATLAB® software or related products does not constitute endorsement or sponsorship by The MathWorks of a particular pedagogical approach or particular use of the MATLAB® software.

CRC Press
Taylor & Francis Group
6000 Broken Sound Parkway NW, Suite 300
Boca Raton, FL 33487-2742

© 2014 by Taylor & Francis Group, LLC
CRC Press is an imprint of Taylor & Francis Group, an Informa business

No claim to original U.S. Government works
Version Date: 20140117

International Standard Book Number-13: 978-1-4398-9853-6 (eBook - PDF)

This book contains information obtained from authentic and highly regarded sources. Reasonable efforts have been made to publish reliable data and information, but the author and publisher cannot assume responsibility for the validity of all materials or the consequences of their use. The authors and publishers have attempted to trace the copyright holders of all material reproduced in this publication and apologize to copyright holders if permission to publish in this form has not been obtained. If any copyright material has not been acknowledged please write and let us know so we may rectify in any future reprint.

Except as permitted under U.S. Copyright Law, no part of this book may be reprinted, reproduced, transmitted, or utilized in any form by any electronic, mechanical, or other means, now known or hereafter invented, including photocopying, microfilming, and recording, or in any information storage or retrieval system, without written permission from the publishers.

For permission to photocopy or use material electronically from this work, please access www.copyright.com (http://www.copyright.com/) or contact the Copyright Clearance Center, Inc. (CCC), 222 Rosewood Drive, Danvers, MA 01923, 978-750-8400. CCC is a not-for-profit organization that provides licenses and registration for a variety of users. For organizations that have been granted a photocopy license by the CCC, a separate system of payment has been arranged.

Trademark Notice: Product or corporate names may be trademarks or registered trademarks, and are used only for identification and explanation without intent to infringe.

Visit the Taylor & Francis Web site at
http://www.taylorandfrancis.com

and the CRC Press Web site at
http://www.crcpress.com

21

Input/Output Devices and Interaction Techniques

Ken Hinckley
Microsoft Research

Robert J.K. Jacob
Tufts University

Colin Ware
University of New Hampshire

Jacob O. Wobbrock
University of Washington

Daniel Wigdor
University of Toronto

21.1 Introduction

The computing literature often draws a sharp distinction between input and output; computer scientists are used to regarding a screen as a passive output device and a mouse as a pure input device. However, nearly all examples of human–computer interaction (HCI) require *both* input and output to do anything useful. For example, what good would a mouse be without the corresponding feedback embodied by the cursor on the screen, as well as the sound and feel of the buttons when they are clicked? The distinction between output devices and input devices becomes even more blurred in the real world. A sheet of paper can be used to both record ideas (input) and display them (output). Clay reacts to the sculptor's fingers yet also provides feedback through the curvature and texture of its surface. Indeed, the complete and seamless integration of input and output is becoming a common research theme in advanced computer interfaces such as ubiquitous computing (Weiser, 1991) and tangible interaction (Ishii and Ullmer, 1997). And of course, with the modern commercial success of mobile touch screen devices such as the Apple iPhone and various Android units, people everywhere experience input and output as entirely inseparable and collocated concepts, their fingers acting directly on virtual widgets displayed on a responsive piece of glass. It is no accident that we should address input and output in the same chapter. Further, as the space required for a powerful computer has plummeted, new form factors have emerged. In mobile devices, the desire to provide large screens to maximize media consumption experiences has marginalized dedicated control surfaces, replacing physical keyboards and pointing devices with ad hoc screen-rendered controls manipulated through touch input. Although measurably inferior at nearly all input tasks, touch screens have nonetheless emerged as a dominant form of mobile computing.

Although inextricably linked, for the ease of discussion, we consider separately the issues of input and output. Input and output bridge the chasm between a computer's inner world of bits and the real world perceptible to the human senses. *Input* to computers consists of sensed information about the physical environment. Familiar examples include the mouse, which senses movement across a surface, and the keyboard, which detects a contact closure when the user presses a key. However, any sensed information about physical properties of people, places, or things can also serve as input to computer systems, for example the location of a user carrying a GPS-enabled mobile phone to inform recommendations for nearby restaurants. *Output* from computers can comprise any emission or modification to the physical environment, such as a display (including the cathode ray tube [CRT], flat-panel displays, or even light emitting diodes), speakers, or tactile and force feedback devices (sometimes referred to as *haptic displays*). There have even been olfactory displays such as the Smell Vision machine that debuted at the World's Fair in Stockholm in 1976. Finally, an *interaction technique* is a fusion of input and output, consisting of all hardware and software elements, that provides a way for the user to accomplish a low-level task. For example, in the traditional graphical user interface (GUI), users can scroll through a document by clicking or dragging the mouse (input) within a scrollbar displayed on the screen (output).

A fundamental task of HCI is to shuttle information between the brain of the user and the silicon world of the computer. Progress in this area attempts to increase the useful bandwidth across that interface by seeking faster, more natural, and more convenient means for users to transmit information to computers, as well as efficient, salient, and pleasant mechanisms to provide feedback to the user. On the user's side of the communication channel, interaction is constrained by the nature of human attention, cognition, and perceptual motor skills and abilities; on the computer side, it is constrained only by the technologies and methods that we can invent.

Research in input and output centers around the two ends of this channel: the devices and techniques computers can use for communicating with people and the perceptual abilities, processes, and organs people can use for communicating with computers.

It then attempts to find the common ground through which the two can be related by studying new modes of communication and expression that could be used for HCI and developing devices and techniques to use such modes. Basic research seeks theories and principles that inform us of the parameters of human cognitive and perceptual facilities, as well as models that can predict or interpret user performance in computing tasks (e.g., Wobbrock et al., 2008). Advances can be driven by the need for new modalities to support the unique requirements of specific application domains, by technological breakthroughs that HCI researchers apply to improving or extending the capabilities of interfaces, or by theoretical insights suggested by studies of human abilities and behaviors, or even problems uncovered during careful analyses of existing interfaces. These approaches complement one another, and all have their value and contributions to the field, but the best research seems to have elements of all of these.

21.2 Interaction Tasks, Techniques, and Devices

A designer looks at the *interaction tasks* necessary for a particular application (Foley et al., 1984). Interaction tasks are low-level primitive inputs required from the user, such as entering a text string or choosing a command. For each such task, the designer chooses an appropriate interaction technique. In selecting an interaction device and technique for each task in a human–computer interface, simply making an optimal choice for each task individually may lead to a poor overall design, with too many different or inconsistent types of devices or dialogues. Therefore, it is often desirable to compromise on the individual choices to reach a better overall design. Design is fundamentally about managing trade-offs in complex solution spaces, and it is often said that designs that are best for something are often worst for something else.

There may be several different ways of accomplishing the same task. For example, one could use a mouse to select a command by using a pop-up menu, a fixed menu (a palette or command bar), multiple clicking, circling the desired command, or even writing the name of the command with the mouse. Software might even detect patterns of mouse use in the background, such as repeated "surfing" through menus, to automatically suggest commands or help topics (Horvitz et al., 1998). The latter suggests a shift from the classical view of interaction as direct manipulation where the user is responsible for all actions and decisions to one that uses *background sensing* techniques to allow technology to support the user with semiautomatic or implicit actions and services (Buxton, 1995a).

21.3 Composition of Interaction Tasks

Early efforts in HCI sought to identify elemental tasks that appear repeatedly in human–computer dialogues. Foley et al. (1984) proposed that user interface transactions are composed of the following elemental tasks:

Selection: Choosing objects from a set of alternatives.
Position: Specifying a position within a range. This includes picking a screen coordinate with a pointing device.
Orient: Specifying an angle or 3D orientation.
Path: Specifying a series of positions and/or orientations over time.
Quantify: Specifying an exact numeric value.
Text: Entry of symbolic data.

While these are commonly occurring tasks in many direct-manipulation interfaces, a problem with this approach is that the level of analysis at which one specifies "elemental" tasks is not well defined. For example, for *Position* tasks, a screen coordinate could be selected using a pointing device such as a mouse, but might be entered as a pair of numeric values (*Quantify*) using a pair of knobs (like an Etch-A-Sketch) where precision is paramount. But if these represent elemental tasks, why do we find that we must subdivide *Position* into a pair of *Quantify* subtasks for some devices but not for others?

Treating all tasks as hierarchies of subtasks, known as *compound tasks*, is one way to address this. With appropriate design and by using technologies and interaction techniques that parallel the way the user thinks about a task as closely as possible, the designer can phrase together a series of elemental tasks into a single *cognitive chunk*. For example, if the user's task is to draw a rectangle, a device such as an

Etch-A-Sketch is easier to use. For drawing a circle, a pen is far easier to use. Hence 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. See Buxton (1986) for further discussion of this important concept.

The six elemental tasks previously enumerated may be a complete list of "fundamental" low-level tasks that underlie most interaction with computers, but it could be argued that this list is not complete; for example, which of these six tasks does a fingerprint scanner support? Perhaps, if used for password replacement, it could be viewed as supporting the *Text* task; alternatively, one might add "Establishment of Identity" to the list. This points to a problem with the fundamental task approach. While identifying "elemental tasks" can be useful for thinking about interaction techniques in general, a problem with viewing tasks as assemblies of elemental tasks is that it typically only considers explicit input in the classical direct-manipulation paradigm. Where do devices like cameras, microphones, and the fingerprint scanner previously discussed fit in? These support higher-level data types and concepts (e.g., images, audio, and identity). Advances in technology will continue to yield new "elemental" inputs. However, these new technologies also may make increasing demands on systems to move from individual samples to synthesis of meaningful structure from the resulting data (Fitzmaurice et al., 1999).

21.4 Properties of Pointing Devices

The breadth of input devices and displays on the market today can be bewildering. It is important to understand that, when considering these devices, they are not all created equal. We must remember that the traditional mouse-based user interface, the Windows–Icons–Menus–Pointers (WIMP), has as perhaps its most essential component an abstraction of the logical target of user actions. This abstraction has gone by many names. The inventors of the mouse, Engelbart and English, named it the *bug* and later referred to it as the *telepointer*. In Windows, it is the pointer. In OSX, it is alternately the pointer and the cursor. But by whatever name, it remains a focal point for user-generated events in the system. A funny thing has happened with the pointer: a kind of abstraction has grown up around it, where a plethora of hardware can control it, and it is the movement of the pointer, rather than the hardware, around which software designers create their experiences. This has led to widespread misunderstanding that the design of the GUI itself is abstract. It's not. It has been designed over more than 40 years of iteration to be highly optimized for a particular piece of hardware.

Thus, when considering input devices, one must be careful to avoid thinking of them in the overly simplistic terms of how they will control a cursor. Instead, we consider a number of organizing properties and principles in order to make sense of the design space and performance issues. First, we consider continuous, manually operated pointing devices (as opposed to discrete input mechanisms such as buttons or keyboards, or other devices not operated with the hand, which we will discuss briefly later). For further insight readers may also wish to consult complete taxonomies of devices (Buxton, 1983; Card et al., 1991). As we shall see, however, it is nearly impossible to describe properties of input devices without reference to output—especially the resulting feedback on the screen—since after all input devices are only useful insofar as they support interaction techniques that allow the user to accomplish something.

21.4.1 Physical Property Sensed

Traditional pointing devices typically sense position, motion, or force. A tablet senses position, a mouse measures motion (i.e., change in position), and an isometric joystick senses force. An isometric joystick is a unmoving force sensing joystick such as the IBM TrackPoint ("eraser head") found on many laptops. For a rotary device, the corresponding properties are angle, change in angle, and torque. Position-sensing devices are also known as *absolute input devices*, whereas motion-sensing devices are *relative input devices*. An absolute device can fully support relative motion, since it can calculate changes to position, but a relative device cannot fully support absolute positioning and in fact can only emulate "position" at all by introducing a cursor on the screen. Note that it is difficult to move the mouse cursor to a particular area of the screen (other than the edges) without looking at the screen, but with a tablet one can easily point to a region with the stylus using the kinesthetic sense (Balakrishnan and Hinckley, 1999), informally known as "muscle memory."

21.4.2 Transfer Function

A device, in combination with the host operating system, typically modifies its signals using a mathematical transformation that scales the data to provide smooth, efficient, and intuitive operation. An appropriate mapping is a transfer function that matches the physical properties sensed by the input device. Appropriate mappings include force-to-velocity, position-to-position, and velocity-to-velocity functions. For example, an isometric joystick senses force; a nonlinear rate mapping transforms this into a velocity of cursor movement (Rutledge and Selker, 1990; Zhai and Milgram, 1993; Zhai et al., 1997). Ideally, the device should also be self-centering when using a rate mapping, with a spring return to the zero input value, so that the user can stop quickly by releasing the device. A common inappropriate mapping is calculating a speed of scrolling based on the position of the mouse cursor, such as extending a selected region by dragging the mouse close to the edge of the screen. The user has no feedback of when or to what extent scrolling will accelerate, and the resulting interaction can be hard to learn how to use and difficult to control.

Gain entails a simple multiplicative transfer function known as the device gain, which can also be described as a *control-to-display (C:D) ratio*, the ratio between the movement of the input device and the corresponding movement of the object it controls. For example, if a mouse (the control) must be moved 1 cm on the desk to move a cursor 2 cm on the screen (the display), the device has a 1:2 control-display ratio. However, on commercial pointing devices and operating systems, the gain is rarely constant*; an *acceleration function* is often used to modulate the gain depending on velocity. An acceleration function is a transfer function that exhibits an exponential relationship between velocity and gain. Experts believe the primary benefit of acceleration is to reduce the *footprint*, or the physical movement space, required by an input device (Hinckley et al., 2001; Jellinek and Card, 1990). One must also be very careful when studying the possible influence of gain settings on user performance: experts have criticized gain as a fundamental concept, since it confounds two separate concepts (device size and display size) in one arbitrary metric (Accot and Zhai, 2001). Furthermore, user performance may exhibit speed–accuracy tradeoffs, calling into question the assumption that there exists an "optimal" C:D ratio (MacKenzie, 1995).

21.4.3 Number of Dimensions

Devices can measure one or more linear and angular dimensions. For example, a mouse measures two linear dimensions, a knob measures one angular dimension, and a six degree-of-freedom (6 DOF) magnetic tracker measures three linear dimensions and three angular (for examples of 6 DOF input and design issues, see Green and Liang, 1994; Hinckley et al., 1994b; Serra et al., 1997; Ware and Jessome, 1988). If the number of dimensions required by the user's interaction task does not match the number of dimensions provided by the input device, then special handling (e.g., interaction techniques that may require extra buttons, graphical widgets, and mode switching) will need to be introduced. A lack of control degrees-of-freedom is a particular concern for 3D user interfaces and interaction (Hinckley et al., 1994b; Zhai, 1998). Numerous interaction techniques have been proposed to allow standard 2D pointing devices to control 3D positioning or orientation tasks (e.g., Bukowski and Sequin, 1995; Chen et al., 1988; Conner et al., 1992). Well-designed interaction techniques using specialized multiple degree-of-freedom input devices can sometimes offer superior performance (Hinckley et al., 1997; Ware and Rose, 1999), but may be ineffective for standard desktop tasks, so overall performance must be considered (Balakrishnan et al., 1997; Hinckley et al., 1999).

---

* Direct input devices (see next page) are an exception, since the C:D ratio is typically fixed at 1:1 (but see also Sears and Shneiderman, 1991 and Forlines et al., 2006).

21.4.4 Pointing Speed, Accuracy, and Throughput

The standard way to characterize pointing device performance employs *Fitts' law* paradigm (Fitts, 1954; Soukoreff and MacKenzie, 2004; Wobbrock et al., 2011). Fitts' law relates the *movement time* to point at a target, the *amplitude* of the movement (the distance to the target), and the *width* of the target (i.e., the precision requirement of the pointing movement). The movement time is proportional to the logarithm of the distance divided by the target width, with constant terms that vary from one device to another. While not emphasized in this chapter, Fitts' law is the single most important quantitative analysis, testing, and prediction tool available to input research and device evaluation. For an excellent overview of its application to the problems of HCI, including use of Fitts' law to characterize throughput (a composite measure of both speed and accuracy often informally called bandwidth), see MacKenzie (1992). For discussion of other accuracy metrics, see MacKenzie et al. (2001). Recently Fitts' law testing paradigm has been proposed as an international standard for evaluating pointing devices (Douglas et al., 1999; Keates et al., 2002).

Recent years have seen a number of new insights and new applications for Fitts' law (Guiard, 2009). Fitts' law was originally conceived in the context of rapid, aimed movements, but Fitts' law can also be applied to tasks such as scrolling (Hinckley et al., 2001), multi-scale navigation (Guiard et al., 2001), crossing boundaries (Accot and Zhai, 2002) and predicting pointing error rates instead of movement time (Wobbrock et al., 2008, 2011). Researchers have also recently applied Fitts' law to expanding targets that double in width as the user approaches them. Even if the expansion begins after the user has already covered 90% of the distance from a starting point, the expanding target can be selected as easily as if it had been fully expanded since the movement began (McGuffin and Balakrishnan, 2002); see also Zhai et al. (2003). However, it remains unclear if this can be successfully applied to improve pointing performance for multiple targets that are closely packed together (as typically found in menus and tool palettes). For tasks that exhibit continuous speed–accuracy requirements, such as moving through a hierarchical menu, Fitts' law cannot be applied, but researchers have recently formulated the steering law, which does addresses such tasks. In fact, the steering law was independently discovered outside the computing fields twice before (Accot and Zhai, 1997, 1999, 2001).

21.4.5 Input Device States

To select a single point or region with an input device, users need a way to signal when they are selecting something versus when they are just moving over something to reach a desired target. The need for this fundamental signal of intention is often forgotten by researchers eager to explore new interaction modalities such as empty-handed pointing (e.g., using camera tracking or noncontact proximity sensing of hand position). The *three-state model* of input (Buxton, 1990b) generalizes the states sensed by input devices as *tracking*, which causes the cursor to move; *dragging*, which allows selection of objects by clicking (as well as moving objects by clicking and dragging them); and *out of range*, which occurs when the device moves out of its physical tracking range (e.g., a mouse is lifted from the desk, or a stylus is removed from a tablet). Most pointing devices sense only two of these three states: for example, a mouse senses tracking and dragging, but a touchpad senses tracking and the out-of-range state. Hence, to fully simulate the functionality offered by mice, touchpads need special procedures, such as tapping to click, which are prone to inadvertent activation (e.g., touching the pad by accident causes a click [MacKenzie and Oniszczak, 1998]). For further discussion and examples, see Buxton (1990b) and Hinckley et al. (1998a).

21.4.6 Direct versus Indirect Control

A mouse is an *indirect input device* (one must move the mouse to point to a spot on the screen); a touch screen is a *direct input device* (the display surface is also the input surface). Direct devices raise several unique issues. Designers must consider the possibility of parallax error resulting from a gap between the input and display surfaces, reduced transmissivity of the screen introduced by a sensing layer, or occlusion of the display by the user's hands. Another issue is that touch screens can support a cursor tracking state, or a dragging state, but not both; typically, touch screens move directly from the out-of-range state to the dragging state when the user touches the screen, with no intermediate cursor feedback (Buxton, 1990b). Techniques for touch-screen cursor feedback have been proposed, but typically require that selection occurs on lift-off (Potter et al., 1988; Sears and Shneiderman, 1991; Sears et al., 1992). See Section 21.5.4.

21.4.7 Device Acquisition Time

The average time to pick up or put down an input device is known as **acquisition time** (or sometimes homing time). It is often assumed to be a significant factor for user performance, but Fitts' law throughput of a device tends to dominate human performance time unless switching occurs frequently (Douglas and Mithal, 1994). However, one exception is stylus or pen-based input devices; pens are generally comparable to mice in general pointing performance (Accot and Zhai, 1999), or even superior for some high-precision tasks (Guiard et al., 2001), but these benefits can easily be negated by the much greater time it takes to switch between using a pen and using a keyboard.

21.4.8 Modeling of Repetitive Tasks

The keystroke-level model (KLM) is commonly used to model expert user performance in repetitive tasks such as text editing. The KLM includes standard operators that represent average times required for pointing with an input device, pressing a button, pauses for decision making, and device acquisition time, but the model does not account for errors or nonexpert behaviors such as problem solving (Card et al., 1980). Good examples of research that apply the KLM include Wang et al. (2001) and MacKenzie and Soukoreff (2002). Another approach called GOMS modeling is an extension of the KLM that can handle more complex cases (Olson and Olson, 1990), but many practitioners still use the KLM to evaluate input devices and low-level interaction techniques because of KLM's greater simplicity.

21.4.9 Hardware Criteria

Various other characteristics can distinguish input devices, but are perhaps less important in distinguishing the fundamental types of interaction techniques that can be supported. Engineering parameters of a device's performance such as sampling rate, resolution, accuracy, and linearity can all influence performance. **Latency** is the end-to-end delay between the user's physical movement, sensing this and providing the ultimate system feedback to the user. Latency can be a devious problem as it is impossible to completely eliminate from system performance; latency of more than 75–100 ms significantly impairs user performance for many interactive tasks (MacKenzie and Ware, 1993; Robertson et al., 1989). For vibrotactile or haptic feedback, users may be sensitive to much smaller latencies of just a few milliseconds (Cholewiak and Collins, 1991). The effects of latency can also be magnified in particular interaction techniques; for example, when dragging on a touch screen, users notice latency as a separation between their finger and the dragged object, even for extremely low latencies (Ng et al., 2012).

21.5 Discussion of Common Pointing Devices

Here, we briefly describe commonly available pointing devices and some issues that can arise with them in light of the properties previously discussed.

21.5.1 Mouse

A mouse senses movement relative to a flat surface. Mice exhibit several properties that are well suited to the demands of desktop graphical interfaces (Balakrishnan et al., 1997). The mouse is stable and does not fall over when released (unlike a stylus on a tablet). A mouse can also provide integrated buttons for selection, and since the force required to activate a mouse's buttons is orthogonal to the plane of movement, it helps minimize accidental clicking or interference with motion. Another subtle benefit is the possibility for users to employ a combination of finger, hand, wrist, arm, and even shoulder muscles to span the range of tasks from short precise selections to large, ballistic movements (Balakrishnan and MacKenzie, 1997; Zhai et al., 1996). Finally, Fitts' law studies show that users can point with the mouse about as well as with the hand itself (Card et al., 1978).

21.5.2 Trackball

A trackball is like a roller-ball mouse that has been turned upside down, with a mechanical ball that rolls in place. Trackballs, like mice, are indirect relative pointing devices. The main advantage of trackballs is that they can be used on an inclined surface and they often require a smaller physical footprint than mice. They also employ different muscle groups, which some users find more comfortable, especially users with motor impairments caused by, e.g., spinal cord injuries (Fuhrer and Fridie, 2001; Sperling and Tullis, 1988; Wobbrock and Myers, 2006). However, rolling the trackball while holding down any of its buttons requires significant dexterity, especially when trying to do so with only one hand. Therefore, tasks requiring moving the trackball with a button held down, for example, dragging an icon across the desktop, can be awkward or even impossible for some users (MacKenzie et al., 1991).

21.5.3 Tablets

Touch-sensitive tablets (that are not also touch screens) are indirect pointing devices and may be used in either absolute or relative pointing modes. In absolute mode, each position on the tablet maps directly to a corresponding position on the screen (e.g., the top-right corner). In relative mode, movements on the tablet move a cursor on the screen according to a transfer function, as with a mouse. Most tablets sense the absolute position of a mechanical intermediary such as a stylus or puck on the tablet surface. A puck is a mouse that is used on a tablet; the only difference is that it senses absolute position and it cannot be used on a surface other than the tablet. Absolute mode is generally preferable for tasks such as tracing, digitizing, drawing, freehand inking, and signature capture. Tablets that sense contact of the bare finger are known as touch tablets (Buxton et al., 1985); touchpads are miniature touch tablets, as commonly found on portable computers (MacKenzie and Oniszczak, 1998). A touch screen is a touch-sensitive tablet collocated with an information display, but demands different handling than a tablet (see Section 21.4.6).

21.5.4 Pen Input

Pen-based input for mobile devices is an area of increasing practical concern. Pens effectively support activities such as inking, marking, and gestural input (see Section 21.8.2), but raise a number of problems when supporting graphical interfaces originally designed for mouse input. Pen input raises the concerns of direct input devices as previously described. There is no way to see exactly what position will be selected before selecting it: pen contact with the screen directly enters the dragging state of the three-state model (Buxton, 1990b). There is no true equivalent of a "hover" state for tool tips* nor an extra button for context menus. Pen dwell time on a target can be used to provide one of these two functions. When detecting a double-tap, one must allow for longer interval between the taps (as compared to double-click on a mouse), and one must also accommodate a significant change to the screen position between taps. Finally, users often want to touch the screen of small devices using a bare finger,

---

* Tool tips are small explanatory labels or balloons that appear next to a button, icon, or other interface widget when the user holds the cursor still over that object. The "hover" state is detected when there is little or no cursor movement for a fixed time-out (many systems use a time-out of approximately 1 s).

so applications should be designed to accommodate imprecise selections. Note that some pen-input devices, such as the Tablet PC, use an inductive sensing technology that can only sense contact from a specially instrumented stylus and thus cannot be used as a touch screen. However, this deficiency is made up for by the ability to track the pen when it is close to (but not touching) the screen, allowing support for a tracking state with cursor feedback (and hence ToolTips as well).

21.5.5 Joysticks

There are many varieties of joysticks. As previously mentioned, an isometric joystick senses force and returns to center when released. Because isometric joysticks can have a tiny footprint, they are often used when space is at a premium, allowing integration with a keyboard and hence rapid switching between typing and pointing (Douglas and Mithal, 1994; Rutledge and Selker, 1990). Isotonic joysticks sense the angle of deflection of the stick; they tend to move more than isometric joysticks, offering better feedback to the user. Such joysticks may or may not have a mechanical spring return to center. Some joysticks even include both force and position sensing and other special features. For a helpful organization of the complex design space of joysticks, see Lipscomb and Pique (1993).

21.5.6 Alternative Means of Pointing

Researchers have explored using the feet (Pearson and Weiser, 1988), head tracking, (Bates and Istance, 2003; LoPresti et al., 2000) and eye tracking (Jacob, 1990; Kumar et al., 2007; Lankford, 2000) as alternative approaches to pointing. Head tracking has much lower pointing bandwidth than the hands and may require the neck to be held in an awkward fixed position, but has useful applications for intuitive coupling of head movements to virtual environments (Brooks, 1988; Sutherland, 1968) and interactive 3D graphics (Hix et al., 1995; Ware et al., 1993). Eye movement-based input, properly used, can provide an unusually fast and natural means of communication, because we move our eyes rapidly and almost unconsciously. The human eye fixates visual targets within the fovea, which fundamentally limits the accuracy of eye gaze tracking to 1 degree of the field of view (Zhai et al., 1999). Eye movements are subconscious and must be interpreted carefully to avoid annoying the user with unwanted responses to his actions, known as the Midas touch problem (Jacob, 1991). Current eye-tracking technology is expensive and has numerous technical limitations, confining its use thus far to research labs and disabled persons with few other options.

21.6 Feedback and Perception-Action Coupling

The ecological approach to human perception (Gibson, 1986) asserts that the organism, the environment, and the tasks the organism performs are inseparable and should not be studied in isolation. Hence, perception and action are intimately linked in a single motor-visual feedback loop, and any separation of the two is an artificial one. The lesson for interaction design is that techniques must consider both the motor control (input) and feedback (output) aspects of the design and how they interact with one another.

From the technology perspective, one can consider feedback as passive or active. Active feedback is under computer control. This can be as simple as presenting a window on a display or as sophisticated as simulating haptic contact forces with virtual objects when the user moves an input device. We will return to discuss active feedback techniques later in this chapter, when we discuss display technologies and techniques.

Passive feedback may come from sensations within the user's own body, as influenced by physical properties of the device, such as the shape, color, and feel of buttons when they are depressed. The industrial design of a device suggests the purpose and use of a device even before a user touches it (Norman, 1990). Mechanical sounds and vibrations that result from using the device provide confirming feedback of the user's action. The shape of the device and the presence of landmarks can help users orient a device without having to look at it (Hinckley et al., 1998b). Proprioceptive and kinesthetic feedback are somewhat imprecise terms, often used interchangeably, that refer to sensations of body posture, motion, and muscle tension (MacKenzie and Iberall, 1994). These senses allow users to feel how they are moving an input device without looking at the device and indeed without looking at the screen in some situations (Balakrishnan and Hinckley, 1999; Kane et al., 2008; Mine et al., 1997). This may be important when the user's attention is divided between multiple tasks and devices (Fitzmaurice and Buxton, 1997). Sellen et al. (1992) report that muscular tension from depressing a foot pedal makes modes more salient to the user than purely visual feedback. Although all of these sensations are passive and not under the direct control of the computer, these examples nonetheless demonstrate that they are relevant to the design of devices, and interaction techniques can consider these qualities and attempt to leverage them.

21.6.1 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 21.1), 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.

Some work has addressed this issue by providing haptic sensations using piezoelectric actuators, electrovibration, or deformable fluids (Bau et al., 2010; Jansen et al., 2010; Poupyrev and Maruyama, 2003; Poupyrev et al., 2002). Others have supplied physical cutouts as templates to provide passive haptic sensations to a finger (or stylus) (Buxton et al., 1985; Wobbrock et al., 2003). Still others have placed transparent physical plastics, ranging in feel from "squishy" to "rigid," on touch-screen surfaces to provide for haptic sensations (Bilton, 2011; Jansen et al., 2012; Weiss, 2010; Weiss et al., 2009). If haptic possibilities are not available, well-designed visual feedback indicating when and where touches were perceived by the system is crucial for effective interaction (Wigdor et al., 2009).

User performance may be influenced by correspondences between input and output. Some correspondences are obvious, such as the need to present confirming visual feedback in response to the user's

TABLE 21.1 Potential Causes of Unexpected Behavior (Left) and the Source of Feedback That Users Receive to Refute Those Causes in Representative Mouse versus Touch Input Systems

	Feedback Refuting Cause	
Cause of Unexpected Behavior	Mouse	Touch
System is nonresponsive	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 (second user captures control)	N/A	Application

actions. Ideally, feedback should indicate the results of an operation before the user commits to it (e.g., highlighting a button or menu item when the cursor moves over it). Kinesthetic correspondence and perceptual structure, described later, are less obvious.

21.6.1.1 Kinesthetic Correspondence

Kinesthetic correspondence refers to the principle that graphical feedback on the screen should correspond to the direction that the user moves the input device, particularly when 3D rotation is involved (Britton et al., 1978). Users can easily adapt to certain non-correspondences: when the user moves a mouse forward and back, the cursor actually moves up and down on the screen; if the user drags a scrollbar downward, the text on the screen scrolls upward. With long periods of practice, users can adapt to almost anything (e.g., for over 100 years psychologists have known of the phenomena of prism adaptation, where people can eventually adapt to wearing prisms that cause everything to look upside down [Stratton, 1897]). However, one should not force users to adapt to a poor design.

21.6.1.2 Perceptual Structure

Researchers have also found that the interaction of the input dimensions of a device with the control dimensions of a task can exhibit perceptual structure. Jacob et al. (1994) explored two input devices: a 3D position tracker with integral (x, y, z) input dimensions and a standard 2D mouse, with (x, y) input separated from (z) input by holding down a mouse button. For selecting the position and size of a rectangle, the position tracker is most effective: here, the integral 3D position input matches the integral presentation of the feedback on the screen. But for selecting the position and grayscale color of a rectangle, the mouse is most effective: here, the user perceives the position and grayscale color of the rectangle as separate quantities and can more easily perform the task when the input device provides separate controls. Hence neither a 3D integral input nor a 2D (x, y) plus 1D (z) input is uniformly superior; the better performance results when the task and device are both integral or both separable.

21.7 Pointing Facilitation Techniques

In an effort to improve the performance of pointing devices, particularly devices for indirect relative pointing (e.g., mice), researchers have invented numerous *pointing facilitation techniques*. Such techniques attempt to improve on the "raw" pointing performance one would exhibit with one's hand or finger by utilizing various means for making targets easier to acquire. Such techniques are sometimes said to "beat Fitts' law" (Balakrishnan, 2004), which refers to performing better than Fitts' law, the widespread quantitative model of human movement time, would predict (Fitts, 1954; MacKenzie, 1992).

Pointing facilitation techniques can be divided into two flavors: those that are *target-agnostic* and those that are *target-aware* (Balakrishnan, 2004; Wobbrock et al., 2009a). Target-agnostic techniques require no knowledge of target identities, locations, or dimensions, working instead only with information directly available from the input device and on-screen cursor. Such techniques are relatively easy to deploy in real-world systems. Target-aware techniques, on the other hand, require knowledge of targets, often target locations and dimensions. Such techniques may also manipulate targets in some way, such as by expanding them (McGuffin and Balakrishnan, 2005). As a result, target-aware techniques are exceedingly difficult to deploy across commercial systems and, until recently, have mostly been confined to research laboratories. Recent efforts, however, have shown that target-aware techniques *can* be deployed on real-world systems by using an underlying architecture for target identification and interpretation (Dixon et al., 2012). Nevertheless, an ongoing theoretical challenge with target-aware techniques is that what constitutes a target is not easily defined. In a word processor, for example, a target may be every character or every word or perhaps even the space *between* characters. In a calendar program, every half hour slot may be a target. In a paint program, every pixel on the paint canvas is a target. Even on the open desktop, icons are not the only targets; every pixel is a candidate for the start of a drag-rectangle operation or a right-click menu. What constitutes a target is a more complex question than at first it may seem.

Target-agnostic techniques avoid the aforementioned issues but are relatively few in number, as only information from the input device and on-screen cursor is available to techniques. Target-agnostic techniques include conventional pointer acceleration (Casiez et al., 2008), dynamic gain adjustment based on the spread of movement angles (Wobbrock et al., 2009a), dynamic gain adjustment based on velocity changes (Hourcade et al., 2008), freezing the mouse cursor in place when the mouse button is down (Trewin et al., 2006), and visual-and-motor-space magnification of pixels with a lens (Findlater et al., 2010; Jansen et al., 2011). Click histories in the form of "magnetic dust" have also been used as a means for providing target-agnostic gravity (Hurst et al., 2007). Placing targets along impenetrable screen edges is also a target-agnostic pointing facilitation strategy, as edges constrain, guide, and trap on-screen cursors (e.g., in screen corners) (Appert et al., 2008; Farris et al., 2001; Froehlich et al., 2007; Walker and Smelcer, 1990). Related to but not strictly a pointing facilitation technique in itself, *kinematic endpoint prediction* (Lank et al., 2007) predicts the endpoints of mouse movements en route to targets using motion kinematic formulae like the minimum-jerk law (Flash and Hogan, 1985).

Target-aware techniques are more numerous than their impoverished target-agnostic brethren. With target identities, locations, and dimensions available to them, target-aware techniques contain the highest-performing pointing techniques in the world. One strategy for target-aware pointing is to modify the mouse cursor. For general-purpose pointing, the *bubble cursor*, which operates using a simple rule that "the closest target is always selected," still remains unbeaten (Grossman and Balakrishnan, 2005). It is rivaled by the *DynaSpot*, which is a speed-dependent bubble cursor that also transitions to being a point cursor after stopping, making it capable of selecting specific pixels, not just objects (Chapuis et al., 2009). Both are variations on the static *area cursor*, which is target-aware so that when its large area overlaps multiple targets, it can degrade to a point cursor at its center (Kabbash and Buxton, 1995; Worden et al., 1997). Other explorations of area cursors have incorporated *goal crossing* (Accot and Zhai, 1997, 2002) into the cursors themselves (Findlater et al., 2010).

Other target-aware schemes modify the targets themselves. Mouse gain is dropped inside *sticky icons* (Blanch et al., 2004; Worden et al., 1997). Even more aggressive, *gravity wells* and *force fields* actually "pull" cursors into themselves (Ahlström et al., 2006; Hwang et al., 2003). *Bubble targets* and, more generally, *target expansion* enlarge targets as the cursor approaches (Cockburn and Firth, 2003; McGuffin and Balakrishnan, 2002, 2005; Zhai et al., 2003). A more extreme form of target expansion is to simply bring targets close to the mouse cursor as it begins its movement (Baudisch et al., 2003). More radically, the mouse cursor can be made to "jump over" the open space between targets, bypassing nontarget pixels altogether (Guiard et al., 2004). There are a great many ways to "beat Fitts' law" and it is likely that new pointing facilitation techniques will be invented for years to come.

21.8 Keyboards, Text Entry, and Command Input

For over a century, keyboards, whether on typewriters, desktop computers, laptops, or mobile devices, have endured as the mechanism of choice for text entry. The resiliency of the keyboard, in an era of unprecedented technological change, is the result of how keyboards complement human skills and may make keyboards difficult to supplant with new input devices or technologies. We summarize some general issues surrounding text entry later, with a focus on mechanical keyboards; see also Lewis et al. (1997) and MacKenzie and Tanaka-Ishii (2007).

21.8.1 Skill Acquisition and Skill Transfer

Procedural memory is a specific type of memory that encodes repetitive motor acts. Once an activity is encoded in procedural memory, it requires little conscious effort to perform (Anderson, 1980). Because procedural memory automates the physical act of text entry, touch typists can rapidly type words without interfering with the mental composition of text. The process of encoding an activity in procedural memory can be formalized as the *power law of practice*: $T_n = T_1 \times n^{(-\alpha)}$, where T_n is the time to perform the *n*th task, T_1 is the time to perform the first task, and α reflects the learning rate (Card et al., 1983; De Jong, 1957; Snoddy, 1926). The power law of practice is sometimes recast as $Y = aX^b$, called the power law of learning, as it produces "learning curves" for which Y is an increasing measure of proficiency (e.g., words per minute), X is a measure of the amount of practice (e.g., trial, block, or session), and a and b are regression coefficients. See MacKenzie and Zhang (1999) for an example in text entry. This suggests that changing the keyboard can have a high relearning cost. However, a change to the keyboard can succeed if it does not interfere with existing skills or allows a significant transfer of skill. For example, some ergonomic keyboards preserve the basic key layout, but alter the typing pose to help maintain neutral postures (Honan et al., 1995; Marklin and Simoneau, 1996), whereas the Dvorak key layout may have some small performance advantages, but has not found wide adoption due to high retraining costs (Lewis et al., 1997).

21.8.2 Eyes-Free Operation

With practice, users can memorize the location of commonly used keys relative to the home position of the two hands, allowing typing with little or no visual attention (Lewis et al., 1997). By contrast, soft keyboards (small on-screen virtual keyboards found on many handheld devices) require nearly constant visual monitoring, resulting in a secondary focus-of-attention (FoA), where the primary focus is on the user's work, that is, the text being composed. A third FoA may consist of a source document that the user is transcribing. Furthermore, with stylus-driven soft keyboards, the user can only strike one key at a time. Thus the design issues for soft keyboards differ tremendously from mechanical keyboards (Zhai et al., 2000).

21.8.3 Tactile Feedback

On a mechanical keyboard users can feel the edges and gaps between the keys, and the keys have an activation force profile that provides feedback of the key strike. In the absence of such feedback, as on touch-screen keyboards (Sears, 1993), performance may suffer and users may not be able to achieve eyes-free performance (Lewis et al., 1997).

21.8.4 Combined Text, Command, and Navigation Input

Finally, it is easy to forget that keyboards provide many secondary command and control actions in addition to pure text entry, such as navigation keys (Enter, Home/End, Delete, Backspace, Tab, Esc, Page Up/Down, Arrow Keys, etc.), chord key combinations (such as Ctrl + C for Copy) for frequently used commands, and function keys for miscellaneous functions defined by the current application. Without these keys, frequent interleaving of mouse and keyboard activity may be required to perform these secondary functions.

21.8.5 Ergonomic Issues

Many modern information workers suffer from repetitive strain injury (RSI). Researchers have identified many risk factors for such injuries, such as working under stress or taking inadequate rest breaks. People often casually associate these problems with keyboards, but the potential for RSI is common to many manually operated tools and repetitive activities (Putz-Anderson, 1988). Researchers have advocated themes for ergonomic design of keyboards and other devices (Pekelney and Chu, 1995), including reducing repetition, minimizing force required to hold and move the device or press its buttons, avoiding sharp edges that put pressure on the soft tissues of the hand, and designing for natural and neutral postures of the user's hands and wrists (Honan et al., 1995; Marklin et al., 1997). Communicating a clear orientation for gripping and moving the device through its industrial design also may help to discourage inappropriate, ergonomically unsound grips.

21.8.6 Other Text Entry Mechanisms

One-handed keyboards can be implemented using simultaneous depression of multiple keys; such *chord keyboards* can sometimes allow one to achieve high peak performance (e.g., court stenographers), but take much longer to learn how to use (Buxton, 1990a; Mathias et al., 1996; Noyes, 1983). They are often used in conjunction with wearable computers (Smailagic and Siewiorek, 1996) to keep the hands free as much as possible (but see also Section 21.8.1). With complex written languages, such as Chinese and Japanese, key chording and multiple stages of selection and disambiguation are currently necessary for keyboard-based text entry (Wang et al., 2001). Handwriting and character recognition may ultimately provide a more natural solution, but for Roman languages handwriting (even on paper, with no recognition involved) is much slower than skilled keyboard use. To provide reliable stylus-driven text input, some systems have adopted unistroke (single-stroke) gestural “alphabets” (Goldberg and Richardson, 1993; Wobbrock et al., 2003) that reduce the demands on recognition technology, while remaining relatively easy for users to guess and learn (MacKenzie and Zhang, 1997; Wobbrock et al., 2005). However, small “two-thumb” keyboards (MacKenzie and Soukoreff, 2002) or fold-away peripheral keyboards are becoming increasingly popular for mobile devices. Recently, hybrid stroke-keyboard combinations have emerged that enable users to tap conventionally on keys or, without explicitly switching modes, to make unistroke gestures over the keys that form the desired word (Kristensson and Zhai, 2004; Zhai and Kristensson, 2003). Precise gestures over keys are not required as the stroke is pattern-matched to produce the intended word. Stroke keyboards have even been commercialized (e.g., ShapeWriter and Swype, both acquired in 2012 by Nuance Corporation) and generally have received a welcome response from consumers. Dictation using continuous speech recognition is available on the market today, but the technology still has a long way to go; a recent study found that the corrected words-per-minute rate of text entry using a mouse and keyboard are about twice as fast as dictation input (Karat et al., 1999). We further discuss speech interaction in the next section.

21.9 Modalities of Interaction

Here, we briefly review a number of general strategies and input modalities that have been explored by researchers. These approaches generally transcend a specific type of input device, but rather span a range of devices and applications.

21.9.1 Speech and Voice

Carrying on a full conversation with a computer as one might do with another person is well beyond the state of the art today and, even if possible, may be a naive goal. Yet even without understanding the content of the speech, computers can digitize, store, edit, and replay segments of speech to augment human–human communication (Arons, 1993; Buxton, 1995b; Stifelman, 1996). Conventional voice mail and the availability of MP3 music files on the web are simple examples of this. Computers can also infer information about the user’s activity from ambient audio, such as determining if the user is present or perhaps engaging in a conversation with a colleague, allowing more timely delivery of information or suppression of notifications that may interrupt the user (Horvitz et al., 1999; Sawhney and Schmandt, 2000; Schmandt et al., 2000).

Understanding speech as input has been a long-standing area of research. While progress is being made, it is slower than optimists originally predicted, and daunting unsolved problems remain. For limited vocabulary applications with native English speakers, speech recognition can excel at recognizing words that occur in the vocabulary. Error rates can increase substantially when users employ words that are out-of-vocabulary (i.e., words the computer is not "listening" for), when the complexity of the grammar of possible phrases increases or when the microphone is not a high-quality close-talk headset. Even if the computer could recognize all of the user's words, the problem of understanding natural language is a significant and unsolved one. It can be avoided by using an artificial language of special commands or even a fairly restricted subset of natural language. But, given the current state of the art, the closer the user moves toward full unrestricted natural language, the more difficulties will be encountered.

Speech input is not the only form of voice-based input researchers have explored. Nonspeech voice input has been examined as well, particularly for scenarios in which speech commands are not well suited. Such scenarios may involve continuous control responses, as opposed to discrete responses. Since speech utterances themselves are discrete, they are well mapped to discrete commands (e.g., "save document") but poorly mapped to continuous activities (e.g., painting in a voice-driven paint program, maneuvering in a first-person shooter, or panning or zooming into documents or photographs). In addition, nonspeech voice has numerous qualities often ignored in speech: loudness, pitch, intonation, nasality, duration, timbre, prosody, vowel quality, and so on. Some of these qualities have been utilized, for example, in the *Vocal Joystick*, a voice control engine enabling continuous omnidirectional control by making vowel sounds that form a continuous circular phonetic space (Harada et al., 2006). Explorations utilizing the Vocal Joystick include drawing programs (Harada et al., 2007) and video games (Harada et al., 2011). Others also have explored controlling video games using nonspeech voice (Igarashi and Hughes, 2001). Even humming and whistling have been explored as forms of nonspeech input for text entry and gaming (Sporka et al., 2004, 2006a,b).

21.9.2 Pens

Pen-based input can emulate mouse-based input, for example, on a Tablet PC, or can support gesture-based input (see Section 21.9.4). Pens afford acting on documents, for example, crossing out a word to delete it or circling a paragraph and drawing an arrow to move it. Besides commands, pens can also provide *ink*, where strokes are not interpreted as commands but left as annotations. Pens thus constitute a versatile input modality, not just an isolated input device.

Pens have been employed in numerous specific interaction techniques besides just the emulation of mice or general gestural input. One of the most compelling is the *marking menu*. Marking menus use directional pen motion to provide rapid menu selection (Kurtenbach and Buxton, 1993; Kurtenbach et al., 1993). Marking menus have been incorporated into composite pen-based systems for indicating targets of an action and the action itself (Hinckley et al., 2005). Entire pen-specific interfaces for Tablet PCs have been explored that incorporate marking menus among other features (Hinckley et al., 2007). Marking menus have been improved or extended in numerous ways, for example, by enabling access to hierarchical menus with multiple discrete strokes (Zhao and Balakrishnan, 2004), using relative position and orientation (Zhao et al., 2006) or using two hands on a touch surface (Lepinski et al., 2010) (see also Section 21.9.3).

Of course, pen-specific interaction techniques go beyond marking menus. Multiple Tablet PCs can be "stitched" together using pen strokes that begin on the screen of one device, cross the bezel boundary, and end on the screen of an adjacent device (Hinckley et al., 2004). Pen pressure has been used in different ways, for example, to activate a magnifying lens for easier target acquisition (Ramos et al., 2007). Pen-based target acquisition has also been facilitated by slip-resistant bubble cursors (Moffatt and McGrenere, 2010) and target-directional beams (Yin and Ren, 2007).

The need for frequent mode switching in pen interfaces has inspired numerous interaction techniques. Barrel buttons, pen dwell time, pen pressure, and the "eraser end" of pens have been explored for mode switching (Li et al., 2005). Pens themselves have been made into multi-touch devices for grip sensing (Song et al., 2011). Even the tilt angle and azimuth of pens have been studied and used to control modes (Xin et al., 2011, 2012). Researchers have also explored multimodal pen and voice input; this is a powerful combination because pen and voice have complementary strengths and weaknesses and can disambiguate one another (Cohen and Sullivan, 1989; Cohen et al., 1997; Harada et al., 2007; Kurihara et al., 2006; Oviatt, 1997).

An important benefit of pen-based interfaces is their support for *sketching*. Sketching interfaces often raise the difficult issue of how to disambiguate users' marks as ink versus as commands (Kramer, 1994; Moran et al., 1997; Mynatt et al., 1999). One solution to this is *hover widgets* (Grossman et al., 2006), which uses the pen hover state on Tablet PCs for command and property selection, separate from inking on the screen. Pen input, via sketching, has been used to define 3D objects (Igarashi et al., 1999; Zeleznik et al., 1996). Computational support for pen-based sketching has a rich history. For a comprehensive review, readers are directed to a recent survey (Johnson et al., 2009).

21.9.3 Touch Input

With the widespread adoption of mobile touch-screen devices, particularly mobile phones based on Apple's iOS operating system or the Android operating system, touch interaction has become the mainstream. Although touch and gesture are often thought of together—and rightly so, as many gestures employ touch—they are not inherently the same thing, as one may have touch without gesture and gesture without touch. In the former case, incidental contact with no intended meaning such as brushing up against an object involves touch but not gesture. In the latter case, one may perform gestures in midair using a computer vision system without touching anything. Despite these examples, for many of today's interactions, touch and gesture are meaningfully coupled together. We cover touch in this section and gesture in the next (Section 21.9.4).

The earliest explorations of touch interaction were on touch tablets that were input-sensing surfaces separated from their output graphical displays. Such tablets often employed relative pointing with a cursor on the display. Although some tablets supported the use of pens, pucks, or other objects, touch tablets, as their name suggests, required no such apparatuses, enabling fingers to directly act on their surfaces (Buxton et al., 1985). Although touch tablets still exist, they are far outnumbered today by touch screens whose input and output surfaces are collocated.

Direct-touch interfaces on touch-screen devices or interactive tabletops usually operate without a cursor in absolute positioning mode. Although styli and pens have been used with touch-screen devices for many years (see Section 21.9.2), often on resistive touch screens, today's devices often employ capacitive touch-sensing techniques enabling the human finger to interact directly. Using the finger in direct-touch interfaces raises many challenges. One challenge is the "fat finger problem," in which the user's relatively large fingertip proves to be insufficiently precise when selecting small on-screen targets. A related challenge is the "occlusion problem," in which the user's finger or hand occludes any objects beneath it (Vogel and Casiez, 2012). Numerous explorations of these issues have been conducted with a variety of proposals for their amelioration. Such proposals include offset cursors and lift-off selection (Potter et al., 1988; Sears and Shneiderman, 1991), magnifying lenses (Roudaut et al., 2008; Vogel and Baudisch, 2007), zooming (Olwal et al., 2008), precision handles (Albinsson and Zhai, 2003), picture-in-picture-style "radar views" (Karlson and Bederson, 2007), probabilistic hit testing (Schwarz et al., 2010, 2011), and techniques utilizing finger orientation (Wang et al., 2009). In addition, a stream of work has studied and exploited interaction on the *backsides* of devices (or, in the case of interactive tabletops, the *undersides*), where fingers cannot possibly occlude portions of the graphical display (Baudisch and Chu, 2009; Hiraoka et al., 2003; Sugimoto and Hiroki, 2006; Wigdor et al., 2006, 2007; Wobbrock et al., 2007, 2008).

Finally, while the aforementioned work has innovated to ameliorate the challenges of touch, others have studied the properties of human touch itself (Wang and Ren, 2009). Findings indicate that users touch with consistent offsets from their perceived target location depending on their finger angle (Holz and Baudisch, 2010), and users align visible features on the *tops* of their fingers with the underlying target on the screen (Holz and Baudisch, 2011). Extracting fingerprints for modeling users and their touch styles can double touch accuracy (Holz and Baudisch, 2010), and if cameras can be used (e.g., above a tabletop), features atop users' fingers can be correlated with their perceived touch point for improved accuracy (Holz and Baudisch, 2011). It is also possible to distinguish with which *part* of a user's finger he or she touches the screen from the acoustic signature of touches (e.g., fingertip, finger pad, fingernail, or knuckle) (Harrison et al., 2011). Rather than distinguishing single touch points, it can be useful to support whole-hand touches that create entire touch regions (Cao et al., 2008).

21.9.4 Gestural Input

Gesture is a powerful input modality that takes place in a variety of forms: as strokes on 2D surfaces or as motions made in 3D space. Depending on the sensing technologies employed, gestures may require holding or wearing an object (e.g., a stylus or a Nintendo Wiimote), or gestures may be performed with the bare fingers or hands (e.g., on a capacitive touch screen or in the air before a Microsoft Kinect). Gestures are highly expressive, rapid, capable of symbolic or metaphorical association, and often entertaining for performers and observers alike. Although gestures have been part of HCI for years, in the last decade, gestures have become a mainstream input modality on commercialized platforms.

Gestures have a powerful property in that they can designate both the *object* of an action and the *action itself* in one fluid motion (Buxton et al., 1983). Thus, gestures support cognitive chunking by integrating command selection with specification of the command's scope (Kurtenbach and Buxton, 1991a,b).

Gestures may be defined in the simplest case by a single point, as in a *tap gesture* performed by a finger on a capacitive touch screen. At this level, the concepts of "touch" and "gesture" are trivially the same. It does not take long, however, for a touch unfolding over time to become a complex gesture involving *paths* or ordered sequences of points over time. Paths may be *unistrokes* (Goldberg and Richardson, 1993), meaning they begin upon receiving the first input point (e.g., "touch down"), unfold as that point moves (e.g., "touch move"), and end when the input points cease (e.g., "touch up"). Of course, a "touch" is only exemplary; a unistroke may be made with a stylus or in midair with a wand or entire hand. The defining aspect of a unistroke is that it is segmented (separated) from gestures that precede or succeed it by a clear signal equivalent to "begin" and "end" with no intervening segmentation signals. Gestures that comprise multiple *successive* unistrokes are called *multistrokes* (see, e.g., Hinckley et al., 2006). Gestures that comprise multiple *concurrent* unistrokes using fingers on a touch screen are called *multi-touch gestures*. The famous "pinch" gesture for resizing photographs is an example of a multi-touch gesture, where the user's thumb and forefinger each perform a unistroke concurrently.

Besides paths, gestures may be composed of *poses*. In the case of hand gestures, poses involve certain positions of the hand that convey meaning, such as a fist or flat palm. Whole-body gestures may also entail poses, such as raising one's hands above one's head (Cohn et al., 2012). Poses may be preserved over a path, such as a fist pose held while punching in a boxing game. Or poses themselves may be dynamic, changing either in place or over a path. Of course, as devices or other objects are incorporated into gestures, the possibilities grow according to the properties available. Gestures made with a rigid remote may be different than those available when using a bendable computer (Schwesig et al., 2004).

In an attempt to organize the high variation possible in gestures, numerous gesture taxonomies have been erected both within and outside the computing field. Early taxonomies by sociolinguists included categories such as Efron's *physiographics*, *kinetographics*, *ideographics*, *deictics*, and *batons* (Efron, 1941). Similarly, McNeill's taxonomy contained *iconics*, *metaphorics*, *deictics*, and *beats* (McNeill, 1992). Kendon placed his gestures on a spectrum of formality, identifying *sign languages* as the most formal and *gesticulation* as the least (Cadoz, 1994; Kendon, 1988) broadly categorizes hand gestures as *semiotic*, *ergotic*, and *epistemic*. Semiotic gestures are those used to communicate meaningful information, such as "thumbs up." Ergotic gestures are those used to manipulate physical objects. Epistemic gestures are exploratory movements to acquire haptic or tactile information; see also D. Kirsh (1995a,b).

The aforementioned gesture categories are not entirely foreign to those developed with computer input in mind. Deictic gestures in particular have received much attention, with several efforts using pointing, typically captured using instrumented gloves or camera-based recognition to interact with "intelligent" environments (Baudel and Beaudouin-Lafon, 1993; Freeman and Weissman, 1995; Jojic et al., 2000; Maes et al., 1996). Deictic gestures in combination with speech recognition have also been studied (Bolt, 1980; Hauptmann, 1989; Lucente et al., 1998; Wilson and Shafer, 2003). Explorations of tangible interaction techniques (Ullmer and Ishii, 1997) and efforts to sense movements and handling of sensor-enhanced mobile devices may be considered ergotic gestures (Harrison et al., 1998; Hinckley et al., 2000, 2003). Most research in hand gesture recognition focuses on empty-handed semiotic gestures (Cassell, 2003), which (Rime and Schiaratura, 1991) are further classified as follows:

Symbolic—conventional symbolic gestures such as "OK"
Deictic—pointing to fill in a semantic frame, analogous to deixis in natural language
Iconic—illustrating a spatial relationship
Pantomimic—mimicking an invisible tool, such as, pretending to swing a golf club

Wobbrock et al. studied gestures that users make on an interactive tabletop for accomplishing common actions like "move," "copy," and "delete" (Wobbrock et al., 2009b). Over 1000 gestures were collected from 20 people for 27 common commands. Findings indicated that apart from (Cadoz, 1994) *ergotic* gestures in which simulated physical objects are manipulated (e.g., moving a virtual object from one screen location to another), users employed widely varying gestures and had little gestural agreement. This led Wobbrock et al. to formulate the following taxonomy of surface gestures with four dimensions, each with various properties. Note that the "Nature" dimension represents a similar classification to those seen thus far:

Form—static pose, dynamic pose, static pose and path, dynamic pose and path, one-point touch, one-point path
Nature—symbolic, physical, metaphorical, abstract
Binding—object-centric, world-dependent, world-independent, mixed dependencies
Flow—discrete, continuous

Freeman et al. extended Wobbrock et al.'s "Form" dimension with the following categories that capture how a gesture begins and unfolds over time (Freeman et al., 2009):

Registration pose—single finger, multi-finger, single shape, multi-shape
Continuation pose—static, dynamic
Movement—no path, path

Not surprisingly, the use of gestures for computer input raises numerous challenges. With most forms of gestural input, errors of user intent and errors of computer interpretation seem inevitable (Bellotti et al., 2002). The learnability of gestures is another persistent challenge, as unlike visible buttons, menus, or hyperlinks, gestures are not trivially discoverable by users. Rather like speech commands, gestures must be *articulated* by users. Numerous efforts have explored how to teach gestures to users (Bau and Mackay, 2008; Bragdon et al., 2009; Fothergill et al., 2012; Freeman et al., 2009; Martin and Isokoski, 2008; Plimmer et al., 2011). Other work has attempted to assess and improve the guessability, learnability, and "naturalness" of gestures in the first place (Alexander et al., 2012; Grandhi et al., 2011; Kane et al., 2011; Költringer and Grechenig, 2004; MacKenzie and Zhang, 1997; Wobbrock et al., 2005, 2009b). Yet another challenge with gestures has been for system designers to choose appropriate and distinct gestures that avoid collision in the *feature-space* utilized by gesture recognizers (Long et al., 1999, 2000). Given the popularity of gestures, some researchers have worked to make gesture recognizers easier to incorporate into software prototypes (Anthony and Wobbrock, 2010, 2012; Henry et al., 1990; Landay and Myers, 1993; Li, 2010; Myers et al., 1997; Swigart, 2005; Wobbrock et al., 2007).

21.9.5 Bimanual Input

Aside from touch typing, most of the devices and modes of operation discussed thus far and in use today involve only one hand at a time. But people use both hands in a wide variety of the activities associated with daily life. For example, when writing, a right-hander writes with the pen in the right hand, but the left hand also plays a crucial and distinct role. It holds the paper and orients it to a comfortable angle that suits the right hand. In fact, during many skilled manipulative tasks, Guiard observed that the hands take on asymmetric, complementary roles (Guiard, 1987): for right-handers, the role of the left hand precedes the right (the left hand first positions the paper), the left hand sets the frame of reference for the action of the right hand (the left hand orients the paper), and the left hand performs infrequent, large-scale movements compared to the frequent, small-scale movements of the right hand (writing with the pen). Most applications for bimanual input to computers are characterized by asymmetric roles of the hands, including compound navigation/selection tasks such as scrolling a web page and then clicking on a link (Buxton and Myers, 1986), command selection using the nonpreferred hand (Bier et al., 1993; Kabbash et al., 1994), as well as navigation, virtual camera control, and object manipulation in 3D user interfaces (Balakrishnan and Kurtenbach, 1999; Hinckley et al., 1998b; Kurtenbach et al., 1997). Researchers have also applied this approach to keyboard design (MacKenzie and Guiard, 2001; McLoone et al., 2003). For some tasks, such as banging together a pair of cymbals, the hands may take on symmetric roles; for further discussion of bimanual symmetric tasks, see Guiard (1987) and Balakrishnan and Hinckley (2000).

21.9.6 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 (BCI) typically employ electroencephalography or functional near-infrared spectroscopy to detect input. Projects have used the technique to detect workload and user engagement (Hirshfield et al., 2009a) in order to conduct usability studies, as well as to explore the possible dynamic adaptation of user interfaces based on such metrics (Hirshfield et al., 2009b). 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.

21.9.7 Passive Measurement: Interaction in the Background

Not all interactions with computers need consist of explicit, intentionally communicated commands. Think about walking into a grocery store with automatic doors. You approach the building; the doors sense this motion and open for you. No explicit communication has occurred, yet a computer has used your action of walking toward the store as an "input" to decide when to open the door. Intentional, explicit interaction takes place in the *foreground*, while implicitly sensed interaction takes place in the *background*, behind the fore of the user's attention (Buxton, 1995a). *Background sensing techniques* will be a major emphasis of future research in automation and sensing systems as users become increasingly mobile and become saturated with information from many sources. Researchers are currently exploring ways of providing context awareness through location sensing; ambient sensing of light, temperature, and other environmental qualities; movement and handling of devices (Goel et al., 2012); detecting the identity of the user and physical objects in the environment; and possibly even physiological measures such as heart-rate variability. This type of information potentially can allow technology to interpret the context of a situation and respond more appropriately (Dey et al., 2001; Hinckley et al., 2003; Schilit et al., 1994; Schmidt, 1999). However, like other recognition-based technologies, there is a risk of errors of user intent or computer interpretation: returning to the automatic door, for example, if you walk by (parallel to) the doors, they may sense your motion and open even if you have no intention of entering the building.

Background interaction can also be applied to explicit input streams through passive behavioral measurements, such as observation of typing speed, manner of moving the cursor (Evans and Wobbrock, 2012), sequence and timing of commands activated in a graphical interface (Horvitz et al., 1998), and other patterns of use. For example, a carefully designed user interface could make intelligent use of such information to modify its dialogue with the user, based on inferences about the user's alertness or expertise. These measures do not require additional input devices, but rather gleaning of additional, typically neglected, information from the existing input stream. These are sometimes known as intelligent or adaptive user interfaces, but mundane examples also exist. For example, cursor control using the mouse or scrolling using a wheel can be optimized by modifying the device response depending on the velocity of movement (Hinckley et al., 2001; Jellinek and Card, 1990).

We must acknowledge the potential for misuse or abuse of information collected in the background. Users should always be made aware of what information is or may potentially be observed as part of a human–computer dialogue. Users should have control and the ability to block any information that they want to remain private (Nguyen and Mynatt, 2001).

21.10 Displays and Perception

We now turn our attention to focus on the fundamental properties of displays and techniques for effective use of displays. We focus on visual displays and visual human perception, since these represent the vast majority of displays, but we also discuss feedback through the haptic and audio channels.

21.10.1 Properties of Displays and Human Visual Perception

Display requirements, such as resolution in time and space, derive from the properties of human vision. Thus, we begin with the basic issues relating to display brightness, uniformity, and spatial and temporal resolution.

21.10.1.1 Dynamic Range

The human eye has an enormous dynamic range. The amount of light reflected from surfaces on a bright day at the beach is about five orders of magnitude higher than the amount available under dim lamplights. Yet the shapes, layouts, and colors of objects look nearly identical to the human eye across much of this range. Most displays in common use are self-luminous CRTs or back-lit liquid crystal displays (LCDs). The best of these devices has a dynamic range (the ratio between the maximum and minimum values produced) of a little more than two orders of magnitude. In practice, under typical room lighting conditions, 15%–40% of the light reaching the user's eye is actually ambient room light reflected by the front surface of the phosphors or off of the screen surface. This means that the effective dynamic range of most devices, unless viewed in dark rooms, is no better than three or four to one. Fortunately the human eye can tolerate extreme variation in the overall level of illumination, as well as the amount of contrast produced by the display.

21.10.1.2 Spatial Frequency

The ability of the human visual system to resolve fine targets is known as *visual acuity*. A standard way of measuring visual acuity is to determine how fine a sinusoidal striped pattern can be discriminated from a uniform gray. Humans are capable of perceiving targets as fine as 50–60 cycles/degree of visual angle

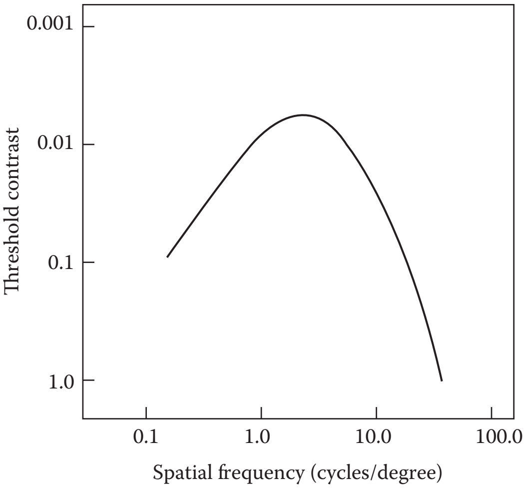

FIGURE 21.1 Spatial contrast sensitivity function of the human visual system. There is a falloff in sensitivity both to detailed patterns (high-spatial frequencies) and to gradually changing gray values (low spatial frequencies).

when the pattern is of very high contrast. Figure 21.1 illustrates the spatial sensitivity of the human eye as a function of spatial frequency. Specifically, it illustrates the degree of contrast required for sinusoidal gratings of different spatial frequencies to be perceived. The function has an inverted U shape with a peak at about two cycles per degree of visual angle. This means that 5 mm stripes at arm's length are optimally visible. The falloff at low spatial frequencies indicates that the human visual system is insensitive to gradual changes in overall screen luminance. Indeed, most CRTs have a brightness falloff toward the edges of as much as 20%, which we barely notice. This nonuniformity is even more pronounced with rear projection systems, due to the construction of screens that project light primarily in a forward direction. This is called the *screen gain*; a gain of 3.0 means that three times as much light is transmitted in the straight through direction compared to a perfect *Lambertian* diffuser. At other angles, less light is transmitted so that at a 45° off-axis viewing angle, only half as much light may be available compared to a perfect diffuser. Screen gain is also available with front projection with similar nonuniformities as a consequence, although the use of curved screens can compensate to some extent.

21.10.1.3 Spatial Resolution

The receptors in the human eye have a visual angle of about 0.8 s of arc. Modern displays provide approximately 40 pixels/cm. A simple calculation reveals that at about a 50 cm viewing distance, pixels will subtend about 1.5 s of arc, about two times the size of cone receptors in the center of vision. Viewed from 100 cm, such a screen has pixels that will be imaged on the retina at about the same size as the receptors. This might suggest that we are in reach of the perfect display in terms of spatial resolution; such a screen would require approximately 80 pixels/cm at normal viewing distances. However, under some conditions the human visual system is capable of producing *superacuities* that imply resolution better than the receptor size. For example, during fusion in stereo vision, disparities smaller than 5 s of arc can be detected (Westheimer, 1979); see also Ware (2000) for a discussion of stereopsis and stereo displays aimed at the practitioner. Another example of superacuity is known as aliasing, resulting from the division of the screen into discrete pixels; for example, a line on the display that is almost (but not quite) horizontal may exhibit a jagged "stairstep" pattern that is very noticeable and unsatisfying. This effect can be diminished by *anti-aliasing*, which computes pixel color values that are averages of all the different objects that contribute to the pixel, weighted by the percentage of the pixel they cover. Similar techniques can be applied to improve the appearance of text, particularly on LCD screens, where individual red, green, and blue display elements can be used for sub-pixel anti-aliasing (Betrisey et al., 2000; Platt, 2000).

21.10.1.4 Temporal Resolution, Refresh, and Update Rates

The *flicker fusion frequency* represents the least rapidly flickering light that the human eye does not perceive as steady. Flicker fusion frequency typically occurs around 50 Hz for a light that turns completely on and off (Wyszecki and Styles, 1982). In discussing the performance of monitors, it is important to differentiate the *refresh rate* and the *update rate*. The refresh rate is the rate at which a screen is redrawn and it is typically constant (values of 60 Hz up to 120 Hz are common). By contrast, the update rate is the rate at which the system software updates the output to be refreshed. Ideally, this should occur at or above the refresh rate, but with increasingly demanding applications and complex data sets, this may not be possible. A rule of thumb states that a 10 Hz update rate is a minimum for smooth animation (Robertson et al., 1989). Motion blur (also known as temporal anti-aliasing) techniques can be applied to reduce the jerky effects resulting from low frame rates (Cook, 1986).

21.11 Color Vision and Color Displays

The single most important fact relating to color displays is that human color vision is trichromatic; our eyes contain three receptors sensitive to different wavelengths. For this reason, it is possible to generate nearly all perceptible colors using only three sets of lights or printing inks. However, it is much more difficult to exactly specify colors using inks than using lights because, whereas lights can be treated as a simple vector space, inks interact in complex nonlinear ways.

21.11.1 Luminance, Color Specification, and Color Gamut

Luminance is the standard term for specifying brightness, that is, how much light is emitted by a self-luminous display. The luminance system in human vision gives us most of our information about the shape and layout of objects in space. The international standard for color measurement is the Commission Internationale de L'Eclairage (CIE) standard. The central function in Figure 21.2 is the CIE $V(\lambda)$ function, which represents the amount that light of different wavelengths contributes to the overall

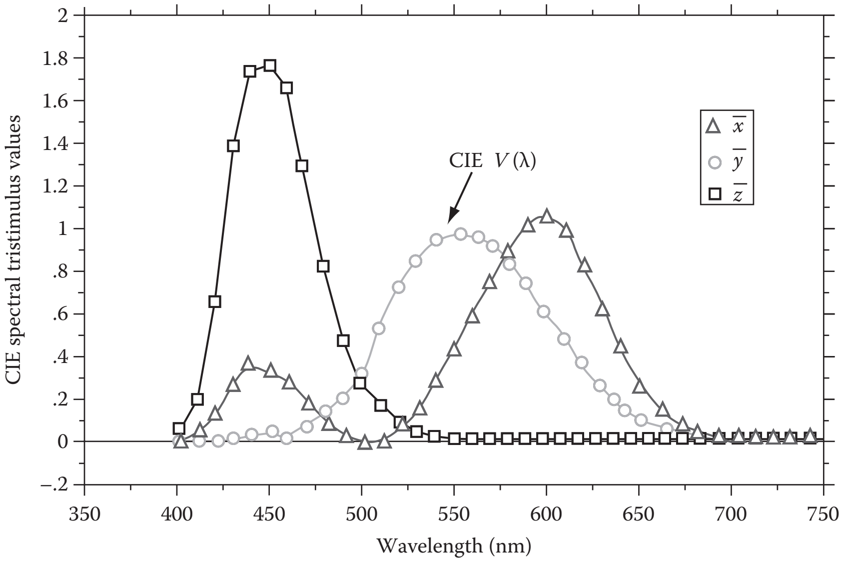

FIGURE 21.2 The CIE tristimulus functions. These are used to represent the standard observer in colorimetry. Short wavelengths at the left-hand side appear blue, in the middle they are green, and to the right they are red. Humans are most sensitive to the green wavelengths around 560 nm.

sensation of brightness. As this curve demonstrates, short wavelengths (blue) and long wavelengths (red) contribute much less than green wavelengths to the sensation of brightness. The CIE tristimulus functions, also shown in Figure 21.2, are a set of color-matching functions that represent the color vision of a typical person. Humans are most sensitive to the green wavelengths around 560 nm. Specifying luminance and specifying a color in CIE tristimulus values are complex technical topics; for further discussion, see Ware (2000) and Wyszecki and Styles (1982).

A chromaticity diagram can be used to map out all possible colors perceptible to the human eye, as illustrated in Figure 21.3. The pure spectral hues are given around the boundary of this diagram in nanometers (10^{-9} m). While the spacing of colors in tristimulus coordinates and on the chromaticity diagram is not perceptually uniform, *uniform color spaces* exist that produce a space in which equal metric distances are closer to matching equal perceptual differences (Wyszecki and Styles, 1982). For example, this can be useful to produce color sequences in map displays (Robertson, 1988).

The gamut of all possible colors is the dark-gray region of the chromaticity diagram, with pure hues at the edge and neutral tones in the center. The triangular region represents the gamut achievable by a particular color monitor, determined by the colors of the phosphors given at the corners of the triangle. Every color within this triangular region is achievable, and every color outside of the triangle is not. This diagram nicely illustrates the trade-off faced by the designer of color displays. A phosphor that produces a very narrow wavelength band will have chromaticity coordinates close to the pure spectral colors, and this will produce more saturated colors (thus enlarging the triangle). However, this narrow band also means that little light is produced.

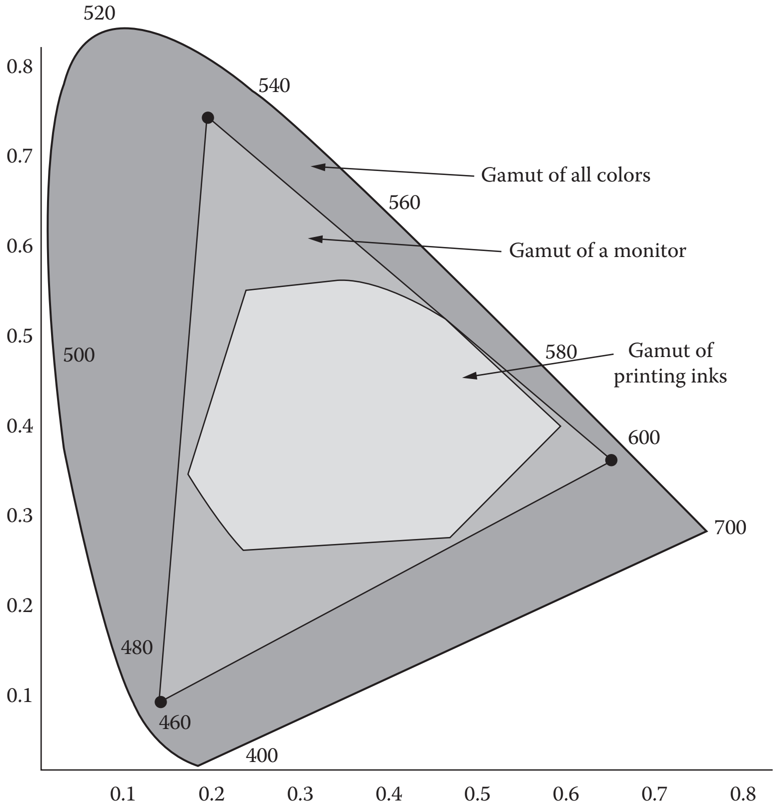

FIGURE 21.3 A CIE chromaticity diagram with a monitor gamut and a printing ink gamut superimposed. The range of available colors with color printing is smaller than that available with a monitor, and both fall short of providing the full range of color that can be seen.

The irregular shape inside the triangle illustrates the gamut of colors obtainable using printing inks. Notice that this set of colors is still smaller, causing difficulties when we try to obtain a hard copy reproduction of the colors on the monitor. Because the eye is relatively insensitive to overall color shifts and overall contrast changes, we can take the gamut from one device and map it into the gamut of the printing inks (or some other device) by compressing and translating it. This is known as gamut mapping, a process designed to preserve the overall color relationships while effectively using the range of a device (Stone et al., 1988). However, it should be noted that the original colors will be lost in this process and that after a succession of gamut mappings, colors may become distorted from their original values.

A process known as chromatic adaptation occurs in the human eye receptors and in the early stages of visual processing: for example, we hardly notice that daylight is much bluer than the yellow cast of tungsten light produced from ordinary light bulbs. The CIE standard does not account for chromatic adaptation nor does it account for color contrast (colors appear differently to the human eye depending on the surrounding visual field). The practical implication is that we can get by with color monitors and printers that are grossly out of calibration. However, accurate color is essential in some applications. It is possible to precisely calibrate a color device so that the particular inputs required to produce a color may be specified in CIE tristimulus values. For monitor calibration, see Cowan (1983); for calibrating print devices, see Stone et al. (1988). It is also possible to correct for the nonlinear response of CRT displays, a process known as *gamma correction*, but keep in mind that CRT designers intentionally insert this nonlinearity to match the human eye's sensitivity to relative changes in light intensity. If one desires a set of perceptually equal gray steps, it is usually best to omit gamma correction. See Ware (2000) for further discussion.

21.12 Information Visualization

Researchers and practitioners have become increasingly interested in communicating large quantities of information quickly and clearly by leveraging the tremendous capabilities of the human visual system, a field known as information visualization. Thanks to advances in computer graphics hardware and algorithms, virtually all new desktop machines available today have sophisticated full-color displays with transparency and texture mapping for complex 2D or 3D scenes, and it now seems inevitable these capabilities will become commonplace on laptop computers, and ultimately even on handheld devices.

21.12.1 General Issues in Information Coding

The greatest challenge in developing guidelines for information coding is that there are usually effective alternatives, such as color, shape, size, texture, blinking, orientation, and gray value. Although a number of studies compare one or more coding methods separately, or in combination, there are so many interactions between the task and the complexity of the display that guidelines based on science are not generally practical. However, Tufte provides excellent guidelines for information coding from an aesthetic perspective (Tufte, 1983, 1990, 1997). For further discussion, examples, and case studies, see also Ware (2000) and Card et al. (1999).

A theoretical concept known as *preattentive processing* has interesting implications for whether or not the coding used can be processed in parallel by the visual system. The fact that certain coding schemes are processed faster than others is called the pop-out phenomenon, and this is thought to be due to early preattentive processing by the visual system. Thus, for example, the shape of the word "bold" is not processed preattentively, and it will be necessary to scan this entire page to determine how many times the word appears. However, if all of the instances of the word *bold* are emphasized, they pop out at the viewer. This is true as long as there are not too many other emphasized words on the same page: if there are less than seven or so instances, they can be processed at a single glance. Preattentive processing is done for color, brightness, certain aspects of texture, stereo disparities, and object orientation and size. Codes that are preattentively discriminable are very useful if rapid search for information is desired (Triesman, 1985). The following visual attributes are known to be preattentive codes and, therefore, useful in differentiating information belonging to different classes:

- *Color*: Use no more than 10 different colors for labeling purposes.
- *Orientation*: Use no more than 10 orientations.
- *Blink coding*: Use no more than 2 blink rates.
- *Texture granularity*: Use no more than 5 grain sizes.
- *Stereo depth*: The number of depths that can be effectively coded is not known.
- *Motion*: Objects moving out of phase with one another are perceptually grouped. The number of usable phases is not known.

However, coding multiple dimensions by combining different pop-out cues is not necessarily effective (Ware, 2000).

21.12.2 Color Information Coding

When considering information display, one of the most important distinctions is between chromatic and luminance information, because these are treated quite differently in human perception. Gray scales are not perceived in the same way as rainbow-colored scales. A purely chromatic difference is one where two colors of identical luminance, such as red and green, are placed adjacent to one another. Research has shown that we are insensitive to a variety of information if it is presented through purely chromatic changes. This includes shape perception, stereo depth information, shape from shading, and motion. However, chromatic information helps us classify the material properties of objects. A number of practical implications arise from the differences in the way luminance and chromatic information is processed in human vision:

- Our spatial sensitivity is lower for chromatic information, allowing image compression techniques to transmit less information about hue relative to luminance.
- To make text visible, it is important to make sure that there is a luminance difference between the color of the text and the color of the background. If the background may vary, it is a good idea to put a contrasting border around the letters (e.g., Harrison and Vicente, 1996).
- When spatial layout is shown either through a stereo display or through motion cues, ensure adequate luminance contrast.
- When fine detail must be shown, for example, with fine lines in a diagram, ensure that there is adequate luminance contrast with the background.
- Chromatic codes are useful for labeling objects belonging to similar classes.
- Color (both chromatic and gray scale) can be used as a quantitative code, such as on maps, where it commonly encodes height and depth. However, simultaneous contrast effects can change the appearance of a patch of color depending on the surrounding colors; careful selection of colors can minimize this (Ware, 1988).

A number of empirical studies have shown color coding to be an effective way of identifying information. It is also effective if used in combination with other cues such as shape. For example, users may respond to targets faster if they can be identified by both shape and color differences (for useful reviews, see Christ [1975]; Stokes et al. [1990]; and Silverstein [1977]). Color codes are also useful in the perceptual grouping of objects. Thus, the relationship between a set of different screen objects can be made more apparent by giving them all the same color. However, it is also the case that only a limited number of color codes can be used effectively. The use of more than about 10 will cause the color categories to become blurred. In general, there are complex relationships between the type of symbols displayed (e.g., point, line, area, or text), the luminance of the display, the luminance and color of the background, and the luminance and color of the symbol (Spiker et al., 1985).

21.12.3 Integrated Control-to-Display Objects

When the purpose of a display is to allow a user to integrate diverse pieces of information, it may make sense to integrate the information into a single visual object or glyph (Wickens, 1992). For example, if the purpose is to represent a pump, the liquid temperature could be shown by changing the color of the pump, the capacity could be shown by the overall size of the pump, and the output pressure might be represented by the changing height of a bar attached to the output pipe, rather than a set of individual dials showing these attributes separately. However, perceptual distortions can result from an ill-chosen display mapping, and the object display may introduce visual clutter: if there are 50 pumps to control, then the outlines of all the pumps may interfere with the data of interest (Tufte, 1983). In object displays, input and output can be integrated in a manner analogous to widgets such as the scrollbar, or even more directly by having input devices that resemble the physical object being handled, known as a prop (Hinckley et al., 1994a). For some good examples of the linking of output and input, see Ahlberg and Shneiderman (1994) as well as Ishii and Ullmer (1997).

This style of presentation and interaction can be especially relevant for telepresence or augmented reality applications, where the user needs to interact with actual physical objects that have attributes that must be viewed and controlled (Feiner et al., 1993; Tani et al., 1992). For more abstract data representation tasks, choosing the color, size, orientation, or texture to represent a particular data attribute may be difficult, and there seem to be practical limits on the number of attributes that one can encode simultaneously. Thus, object displays must usually be custom designed for each different display problem. In general, this means that the display and controls should somehow match the user's cognitive model of the task (Cole, 1986; Norman, 1990).

21.12.4 3D Graphics and Virtual Reality

Much research in 3D information visualization and **virtual reality** is motivated by the observation that humans naturally operate in physical space and can intuitively move about and remember where things are (an ability known as spatial memory). However, translating these potential benefits to artificially generated graphical environments is difficult because of limitations in display and interaction technologies. Virtual environment research pushed this to the limit by totally immersing the user in an artificial world of graphics, but this comes at the cost of visibility and awareness of colleagues and objects in the real world. This has led to research in so-called *fish tank virtual reality* displays by using a head tracking system in conjunction with a stereo display (Deering, 1992; Ware et al., 1993) or a mirrored setup, which allows superimposition of graphics onto the volume where the user's hands are located (Schmandt, 1983; Serra et al., 1997). However, much of our ability to navigate without becoming lost depends upon the vestibular system and spatial updating as we physically turn our bodies, neither of which is engaged with stationary displays (Chance et al., 1998; Loomis et al., 1999). For further discussion of navigation in virtual environments, see Darken and Sibert (1993, 1995); for application of spatial memory to 3D environments, see Robertson et al. (1998, 1999).

Outside of virtual reality, *volumetric displays* present imagery in true 3D space, by illuminating "voxels" (volumetric pixels) in midair. Favalora provides a thorough survey of the various technological implementations of volumetric displays (Favalora, 2005). The true 3D imagery in volumetric displays has been shown to improve depth perception (Grossman and Balakrishnan, 2006b) and shape recognition (Rosen et al., 2004). Besides providing true 3D images, the main difference from the other 3D tabletop displays is that volumetric display is generally enclosed by a surface. This means that users cannot directly interact with the 3D imagery. Balakrishnan et al. explored the implications of this unique difference to interaction design by using physical mock-ups (Balakrishnan et al., 2001). More recent working implementations have allowed users to interact with the display by using hand and finger gestures on and above the display surface (Grossman et al., 2004) and by using a handheld six degree-of-freedom input device (Grossman and Balakrishnan, 2006a). Another key differentiator is that volumetric displays better support collocated multiuser interaction, given a shared view of 3D imagery, without the need to wear glasses or head-mounted displays that can interfere with natural collaboration. Grossman and Balakrishnan describe various techniques that can overcome the limitations of multiuser interaction with volumetric displays (Grossman and Balakrishnan, 2008).

21.12.5 Augmented Reality

Augmented reality superimposes information on the surrounding environment rather than blocking it out. For example, the user may wear a semitransparent display that has the effect of projecting labels and diagrams onto objects in the real world. It has been suggested that this may be useful for training people to use complex systems or for fault diagnosis. For example, when repairing an aircraft engine, the names and functions of parts could be made to appear superimposed on the parts seen through the display together with a maintenance record if desired (Caudell and Mizell, 1992; Feiner et al., 1993). The computer must obtain a detailed model of the environment; otherwise it is not possible to match the synthetic objects with the real ones. Even with this information, correct registration of computer graphics with the physical environment is an extremely difficult technical problem due to measurement error and system latency. This technology has been applied to heads-up displays for fighter aircraft, with semitransparent information about flight paths and various threats in the environment projected on the screen in front of the pilot (Stokes et al., 1990), as well as digitally augmented desk surfaces (Wellner, 1993).

21.13 Scale in Displays

It is important to consider the full range of scale for display devices and form factors that may embody an interaction task. Computer displays increasingly span orders of magnitude in size and available computational resources, from watches, handheld personal data assistants (PDAs), tablet computers, and desktop computers, all the way up to multiple-monitor and wall-size displays. A technique that works well on a desktop computer, such as a pull-down menu, may be awkward on a small handheld device or even unusable on a wall-size display (where the top of the display may not even be within the user's reach). Each class of device seems to raise unique challenges, and the best approach may ultimately be to design special-purpose, appliance-like devices (see Want and Borriello [2000] for a survey) that suit specific purposes.

21.13.1 Small Displays

Users increasingly want to do more and more on handheld devices, mobile phones, pagers, and watches that offer less and less screen real estate. Researchers have investigated various strategies for conserving screen real estate. Transparent overlays allow divided attention between foreground and background layers (Harrison and Vicente, 1996; Harrison et al., 1995a,b; Kamba et al., 1996), but some degree of interference seems inevitable. This can be combined with sensing which elements of an interface are being used, such as presenting widgets on the screen only when the user is touching a pointing device (Hinckley and Sinclair, 1999). Researchers have also experimented with replacing graphical interfaces with *graspable* interfaces that respond to tilting, movement, and physical gestures that do not need constant on-screen representations (Fitzmaurice et al., 1995; Harrison et al., 1998; Hinckley et al., 2000; Rekimoto, 1996). Much research in focus plus context techniques, including fish-eye magnification (Bederson, 2000) and zooming metaphors (Bederson et al., 1996; Perlin and Fox, 1993; Smith and Taivalsaari, 1999), has also been motivated by providing more space than the boundaries of the physical screen can provide. Researchers have started to identify principles and quantitative models to analyze the trade-offs between multiple views and zooming techniques (Baudisch et al., 2002; Plumlee and Ware, 2002). There has been considerable effort devoted to supporting web browsing in extremely limited screen space (Baudisch et al., 2004; Buyukkokten et al., 2001; Jones et al., 1999; Trevor et al., 2001; Wobbrock et al., 2002).

21.13.2 Multiple Displays

Researchers have recently recognized that some very interesting design issues arise when multiple displays are considered, rather than the traditional single display of desktop computers. Having multiple monitors for a single computer is not like having one large display (Grudin, 2001). Users employ the boundary between displays to partition their tasks, with one monitor being reserved for a primary task and other monitors being used for secondary tasks. Secondary tasks may support the primary task (e.g., reference material, help files, or floating tool palettes), may provide peripheral awareness of ongoing events (such as an e-mail client), or may provide other background information (to-do lists, calendars, etc.). Switching between applications has a small time penalty (incurred once to switch and again to return), and perhaps more importantly, it may distract the user or force the user to remember information while switching between applications. Having additional screen space "with a dedicated purpose, always accessible with a glance" (Grudin, 2001), reduces these burdens (Czerwinski et al., 2003), and studies suggest that providing multiple, distinct foci for interaction may aid users' memory and recall (Tan et al., 2001, 2002). Finally, small displays can be used in conjunction with larger displays (Myers et al., 1998, 2000; Rekimoto, 1998), with controls and private information on the small device and shared public information on the larger display. This shows how displays of different dimensions support completely different user activities and social conventions. It is also possible to dynamically join multiple displays for collaboration or to create a larger but temporary tiled display (Hinckley, 2003a,b; Tandler et al., 2001).

21.13.3 Large-Format Displays

Trends in display technology suggest that large-format displays will become increasingly affordable and common. A recent journal special issue includes numerous articles on implementing large-format displays using projection, application design for large displays, and specific application domains such as automotive design (Funkhouser and Li, 2000). Large displays often implicitly suggest multiple simultaneous users, with many applications revolving around collaboration (Funkhouser and Li, 2000; Swaminathan and Sato, 1997) and giving a large-scale physical presence to virtual activities (Buxton et al., 2000). To support input directly on whiteboard-size displays, researchers have explored gestural interaction techniques for pens or touch screens (Guimbretiere et al., 2001; Moran et al., 1997). Some technologies cannot handle more than one point of contact, so system developers must check this carefully if simultaneous use by multiple persons is desired. Large displays also seem to lend themselves to interaction at a distance, although using laser pointers to support such interaction (Myers et al., 2002; Olsen and Nielsen, 2001) has met with mixed success due to the lack of separation between tracking versus dragging states (Buxton, 1990b); using small handheld devices to interact with the full area of a large display also is problematic as the ratio of the display size to the control surface size may be very large (Myers et al., 1998). Environmentally situated ambient displays share some properties of large displays, but emphasize subtle presentation of information in the periphery of attention (Ishii and Ullmer, 1997; Wisneski et al., 1998). Large-format displays and virtual realities also share some design issues; see the taxonomy of Buxton and Fitzmaurice for further discussion (Buxton and Fitzmaurice, 1998).

Unless life-size viewing of large objects is necessary (Buxton et al., 2000), in general it is not yet clear what performance benefits a single large display may offer as compared to multiple monitors with the same screen area partitioned by bezels (Czerwinski et al., 2003). One recent study suggests that the increased field of view afforded by large-format displays can lead to improved 3D navigation performance, especially for women (Czerwinski et al., 2002).

21.14 Force and Tactile Displays

Haptic feedback research has sought to provide an additional channel of sensory feedback by synthesizing forces on the skin of the operator. The touch sensation is extraordinarily complex.

In fact, the sense of "touch" is a very imprecise term: it includes an amalgamation of multiple sensory systems, including sensitivity to pressure, small shear forces in the skin, heat and cold, pain, kinesthesis and proprioception, and the vestibular system (Burdea, 1996; MacKenzie and Iberall, 1994).

There appears to be no physical means by which a complex tactile stimulus can be delivered except in a very localized way. As a result, most haptic feedback devices are limited to simulation of a single point of contact, analogous to feeling the world with the tip of a pencil, although a few examples of whole-hand force feedback devices exist (Burdea, 1996; Iwata, 1990). Efforts in haptic feedback include force feedback (active presentation of forces to the user) and tactile feedback (active presentation of vibrotactile stimuli to the user). Haptic feedback is popular for gaming devices, such as force feedback steering wheels and joysticks, but general-purpose pointing devices with force or tactile feedback remain uncommon. For a comprehensive discussion of force and tactile feedback technologies and techniques, as well as perceptual properties of the skin and joints, see Burdea (1996).

Adding force feedback to a mouse or stylus may impose constraints on the mechanical design, since a physical linkage is typically needed to reflect true forces. This may prevent a force feedback mouse from functioning like a traditional mouse, as it may limit range of motion or preclude clutching by lifting the device. Some devices instead increase resistance between the mouse and the pad, but this prevents simulation of hard contact forces. One can also use a vibrotactile stimulus, such as a vibrating pin under the mouse button or a vibrating shaft of an isometric joystick (Campbell et al., 1999). Combination devices have also been explored (Akamatsu and MacKenzie, 1996). Vibrotactile feedback seems especially promising for small mobile devices, for example, to provide the user with feedback of command recognition when the user's attention may not be focused on the screen (Poupyrev et al., 2002). Applications for remote controls and augmented handles also look promising (MacLean et al., 2000; Snibbe and MacLean, 2001).

Using force feedback to provide attractive forces that pull the user toward a target, or tactile feedback to provide additional feedback for the boundaries of the target, has been found to yield modest speed improvements in some target acquisition experiments, although error rates may also increase (Akamatsu and MacKenzie, 1996; MacKenzie, 1995). However, there have been almost no published studies for tasks where multiple targets are present, as on a computer screen with many icons and menus. Haptic feedback for one target may interfere with the selection of another, unless one uses techniques such as reducing the haptic forces during rapid motion (Oakley et al., 2001). Finally, one should also consider whether software constraints, such as snap to grids, are sufficient to support the user's tasks.

The construction of force output devices is extremely technically demanding. They must be stiff in order to be able to create the sensation of solid contact, yet light so that they have little inertia themselves, and there must be a tight loop between input (position) and output (force). Sigoma (1993) has suggested that having this loop iterated at 5 kHz may be necessary for optimal fine motor control. It has been shown that force feedback improves performance in certain telerobotic applications when, for example, inserting a peg into a hole (Sheridan, 1992). The most promising applications of force output seem to appear in domains where simulation of force is essential, such as surgical simulation and telerobotics (Burdea, 1996).

Another fundamental challenge for haptic feedback techniques results from the interaction between the haptic and visual channels. Visual dominance deals with phenomena resulting from the tendency for vision to dominate other modalities (Wickens, 1992). Campbell et al. (1999) show that tactile feedback improves steering through a narrow tunnel, but only if the visual texture matches the tactile texture; otherwise, tactile feedback harms performance.

21.15 Auditory Displays

Here, we consider computer-generated auditory feedback. Speech audio can consist of synthesized or recorded speech. All other audio feedback is known as nonspeech audio. With stereo speakers or a stereo headset, either type of audio can be presented such that it seems to come from a specific 3D location around the user, known as spatialized audio. For speech input and technology-mediated human–human communication applications that treat stored voice as data, see Section 21.8.1.

21.15.1 Nonspeech Audio

Nonspeech auditory feedback is prevalent in video games but largely absent from other interaction with computing devices. Providing an auditory echo of the visual interface has little or no practical utility and may annoy users. Audio should be reserved to communicate simple, short messages that complement visual feedback (if any). Furthermore, one or more of the following conditions should hold: the message should (1) deal with events in time, (2) call for immediate action, or (3) take place when the user's visual attention may be overburdened or directed elsewhere (Buxton, 1995b; Deatherage, 1972). For example, researchers have attempted to enhance scrollbars using audio feedback (Brewster et al., 1994). However, the meaning of such sounds may not be clear. Gaver advocates ecological sounds that resemble real-world events with an analogous meaning. For example, an empty disc drive might sound like a hollow metal container (Gaver, 1989). If a long or complex message must be delivered using audio, it will likely be quicker and clearer to deliver it using speech output. Audio feedback may be crucial to support tasks or functionality on mobile devices that must take place when the user is not looking at the display (for some examples, see Hinckley et al., 2000).

Nonspeech sounds can be especially useful for attracting the attention of the user. Auditory alerting cues have been shown to work well, but only in environments where there is low auditory clutter. However, the number of simple nonspeech alerting signals is limited, and this can easily result in misidentification or cause signals to mask one another. An analysis of sound signals in fighter aircraft (Doll and Folds, 1985) found that the ground proximity warning and the angle-of-attack warning on an F16 were both an 800 Hz tone, a dangerous confound since these conditions require opposite responses from the pilot. It can also be difficult to devise nonspeech audio events that convey information without provoking an alerting response that unnecessarily interrupts the user. For example, this design tension arises when considering nonspeech audio cues that convey various properties of an incoming e-mail message (Hudson and Smith, 1996; Sawhney and Schmandt, 2000).

21.15.2 Speech Output

Speech auditory output is generally delivered through either recorded speech segments or completely synthetic speech (also known as text-to-speech technology). There has been considerable interest, especially for military applications, in the use of speech in providing warnings to the operators of complex systems. Speech can provide information to direct the operator's attention in a way that alarms cannot (since an unfamiliar alarm simply indicates a problem, without telling the user the nature or context of the problem). Synthetic speech is most useful where visual information is not available, for example, in touch-tone phone menu systems or in screen reader software for blind or low-vision users. Although progress is being made, synthetic voices still sound somewhat unnatural and may be more difficult for users to understand. Recorded speech is often used to give applications, particularly games, a more personal feel, but can only be used for a limited number of responses known in advance.

The rate at which words must be produced to sound natural is a narrow range. For warning messages, 178 words per minute is intelligible but hurried, 123 words per minute is distracting and irritatingly slow, and a more natural rate of 156 words per minute is preferred (Simpson and Marchionda-Frost, 1984). The playback rate of speech can be increased by overlapping samples in time such that one sample is presented to one ear and another sample to the other ear. Technologies to correct for pitch distortions and remove pauses have also been developed (Arons, 1993; Sawhney and Schmandt, 2000; Stifelman, 1996). It is recommended by the US Air Force that synthetic speech be 10 dB above ambient noise levels (Stokes et al., 1990).

21.15.3 Spatialized Audio Displays

It is possible to synthesize spatially localized sounds with a quality such that spatial localization in the virtual space is almost as good as localization of sounds in the natural environment (Wenzel, 1992). Auditory localization appears to be primarily a 2D phenomenon, that is, observers can localize in horizontal position (azimuth) and elevation angle to some degree of accuracy. Azimuth and elevation accuracies are of the order of 15°. As a practical consequence this means that sound localization is of little use in identifying sources in conventional screen displays. Where localized sounds are really useful is in providing an orienting cue or warning about events occurring behind the user, outside of the field of vision.

There is also a well-known phenomenon called visual capture of sound. Given a sound and an apparent visual source for the sound, for example, a talking face on a cinema screen, the sound is perceived to come from the source despite the fact that the actual source may be off to one side. Thus, visual localization tends to dominate auditory localization when both kinds of cues are present.

21.16 Future Directions

The future of interaction with computers will both be very different and very much like it is today. Some of our current tools, such as mice and keyboards, have evolved to suit interaction with desktop GUIs and rapid text entry. As long as users' work continues to involve tasks such as calculating budgets, writing reports, looking up citations, exchanging memos, and other knowledge worker tasks that seem to lend themselves to solution using desktop computers, we will continue to see mice and keyboards in use, not only because they are familiar but also because they closely match human skills and the requirements of the tasks. Devising new techniques that provide more efficient pointing at a desktop display than a mouse, for example, is difficult to achieve (Card et al., 1978). Speech recognition will allow new types of interaction and may enable interaction where it previously has been difficult or infeasible. However, even as technical limitations are removed, speech interaction will not replace all forms of interaction: 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).

But our computers and the tasks they serve are rapidly evolving. Current handheld devices have the display and computational capabilities of common desktop machines from several years ago. What is lacking are new methods of interacting with such devices that uniquely suit mobile interaction, rather than derivatives of the desktop interface. Researchers are still actively exploring and debating the best ways to achieve this. Meanwhile, technology advances and economic trends continue to drive the cost of commodity displays lower and lower, while the limits of the technology continue to increase. Thus, we will continue to see new innovations in both very small and very large displays, and as these become commonplace, new forms of interaction will become prevalent. Very small displays invariably seem to be incorporated into input/output appliances such as watches, pagers, and handheld devices, so interaction techniques for very small form factors will become increasingly important.

The Internet and wireless networking seem to be the main disruptive technologies of the current era. Indeed, it seems likely that 100 years from now the phrase "wireless network" will seem every bit as antiquated as the phrase "horseless carriage" does today. Nobody really understands yet what it will mean for everything and everyone to be connected, but many researchers are working to explore the vision of ubiquitous computing originally laid out by Weiser (1991). Techniques that allow users to communicate and share information will become increasingly important. Biometric sensors or other convenient means for establishing identity will make services such as personalization of the interface and sharing data much simpler (Rekimoto, 1997; Sugiura and Koseki, 1998; Westeyn et al., 2005). Techniques that combine dissimilar input devices and displays in interesting ways also will be important to realize the full potential of these technologies (e.g., Myers et al., 2001; Streitz et al., 1999). Electronic tagging techniques for identifying objects (Want et al., 1999) may also become commonplace. Such a diversity of locations, users, and task contexts points to the increasing importance of sensors to acquire contextual information, as well as machine learning techniques to interpret them and infer meaningful actions (Bellotti et al., 2002; Buxton, 1995a; Hinckley et al., 2003). This may well lead to an age of ubiquitous sensors (Saffo, 1997) with devices that can see, feel, and hear through digital perceptual mechanisms.

Key Terms

Absolute input device: An input device that reports its actual position, rather than relative movement. A tablet or touch screen typically operates this way (see also relative input device).

Acquisition time: The average time to pick up or put down an input device. It is sometimes known as homing time.

Anti-aliasing: The specification of pixel color values so that they reflect the correct proportions of the colored regions that contribute to that pixel. In temporal anti-aliasing the amount of time a region of a simulated scene contributes to a pixel is also taken into account.

Augmented reality: The superimposition of artificially generated graphical elements on objects in the environment. It is achieved with a see-through head-mounted display.

Background sensing techniques: Implicitly sensed interaction takes place in the background, behind the fore of the user's attention. Background sensing techniques use sensor technology or intelligent algorithms to glean additional, typically neglected, information from the existing input stream, with the goal of supporting the user with semiautomatic or implicit actions and services.

Cognitive chunk: A series of elemental tasks that seems like a single concept to the user. For example, users think of pointing at something as a single chunk, but from a technical perspective it may consist of selecting an (X, Y, Z) coordinate in a 3D environment. By using technologies and interaction metaphors that parallel the way the user thinks about a task as closely as possible, the designer can phrase together a series of elemental tasks into a single cognitive chunk.

Compound tasks: A compound task is a hierarchy of elemental subtasks. For example, the navigate/select compound task consists of scrolling to view an item in a list and then clicking on it to select it. When interacting with a graphical scrollbar, scrolling itself may be a compound task with multiple selection or positioning tasks.

C:D ratio: The ratio between the movement a user must make with an input device and the resulting movement obtained on the display. With a large C:D ratio, a large movement is required to effect a small change on the display, affording greater precision. A low ratio allows more rapid operation and takes less desk space. The C:D ratio is sometimes expressed as a single number, in which case it is referred to as the device gain. Note that many experts have criticized gain as a fundamental concept; one must take great care when manipulating gain in experiments, since it confounds display size and control size in one arbitrary metric.

Direct input device: A device that the user operates directly on the screen or other display to be controlled, such as a touch screen (see also indirect input device).

Fish tank virtual reality: A form of virtual reality display that confines the virtual scene to the vicinity of a monitor screen.

Fitts' law: A model that relates the movement time to point at a target, the amplitude of the movement (the distance to the target), and the width of the target (i.e., the precision requirement of the pointing movement). The movement time is proportional to the logarithm of the distance divided by the target width, with constant terms that vary from one device to another. Fitts' law has found wide application in HCI to evaluating and comparing input devices and transfer functions for pointing at targets.

Flicker fusion frequency: The frequency at which a flickering light is perceived as a steady illumination. It is useful in determining the requirements for a visual display.

Footprint: The physical movement space (area) required to operate an input device.

Fovea: The central part of the retina at which vision is the sharpest, about 2° of visual angle in diameter.

Gamma correction: The correction of nonlinearities of a monitor so that it is possible to specify a color in linear coordinates.

Indirect input device: A device that the user operates by moving a control that is located away from the screen or other display to be controlled, such as a mouse or trackball (see also direct input device).

Input device: A hardware computer peripheral through which the user interacts with the computer.

Interaction task: A low-level primitive input to be obtained from the user, such as entering a text string or choosing a command.

Interaction technique: The fusion of input and output, consisting of all hardware and software elements, that provides a particular way for the user to accomplish a low-level task with a physical input device. For example, the pop-up menu is an interaction technique for choosing a command or other item from a small set, using a mouse and a graphical display.

Lambertian diffuser: A diffuser that spreads incoming light equally in all directions.

Latency: The end-to-end delay between the user's physical movement and the system's ultimate feedback to the user. Latency of more than 75–100 ms significantly impairs user performance for many interactive tasks.

Luminance: The standard way of defining an amount of light. This measure takes into account the relative sensitivities of the human eye to light of different wavelengths.

Preattentive processing: Visual stimuli that are processed at an early stage in the visual system in parallel. This processing is done prior to processing by the mechanisms of visual attention.

Refresh rate: The rate at which a computer monitor is redrawn. It is sometimes different from the update rate.

Relative input device: An input device that reports its distance and direction of movement each time it is moved, but cannot report its absolute position. A mouse operates this way (see absolute input device).

Screen gain: A measure of the amount by which a projection video screen reflects light in a preferred direction. The purpose is to give brighter images if viewed from certain positions. There is a corresponding loss in brightness from other viewing positions.

Superacuities: The ability to perceive visual effects with a resolution that is finer than can be predicted from the spacing of receptors in the human eye.

Three-state model: A model for the discrete states of input devices that models transitions between three states: tracking, dragging, and out of range. Most input devices only sense two of these three states (e.g., a mouse senses tracking and dragging, whereas a touchpad senses tracking and the out-of-range state).

Transfer function. A mathematical transformation that scales the data from an input device to ideally provide smooth, efficient, and intuitive operation. Appropriate mappings are transfer functions that match the physical properties sensed by the input device and include force-to-velocity, position-to-position, and velocity-to-velocity functions.

Uniform color space: A transformation of a color specification such that equal metric differences between colors more closely correspond to equal perceptual differences.

Update rate: The rate at which the image on a computer monitor is changed.

Virtual reality: A method of monitoring a user's head position and creating a perceptive view of an artificial world that changes as the user moves, in such a way as to simulate an illusory 3D scene.

Visual acuity: The ability of the human visual system to resolve fine targets.

References

Accot, J. and S. Zhai (1997). Beyond Fitts' law: Models for trajectory-based HCI tasks. *Proceedings of CHI'97: ACM Conference on Human Factors in Computing Systems*, Atlanta, GA. pp. 295–302. http://portal.acm.org/citation.cfm?id=258760

Accot, J. and S. Zhai (1999). Performance evaluation of input devices in trajectory-based tasks: An application of the Steering law. *Proceedings of CHI'99*, Pittsburgh, PA. pp. 466–472.

Accot, J. and S. Zhai (2001). Scale effects in Steering law tasks. *Proceedings of CHI'2001: ACM Conference on Human Factors in Computing Systems*, Seattle, WA. pp. 1–8.

Accot, J. and S. Zhai (2002). More than dotting the i's—Foundations for crossing-based interfaces. *ACM CHI 2002 Conference on Human Factors in Computing Systems*, Minneapolis, MN. pp. 73–80. http://portal.acm.org/citation.cfm?id=503390

Ahlberg, C. and B. Shneiderman (1994). The alphaslider: A compact and rapid selector. *Proceedings of CHI'94*, Boston, MA. pp. 365–371.

Ahlström, D., M. Hitz, and G. Leitner (2006). An evaluation of sticky and force enhanced targets in multi target situations. Paper presented at the *Proceedings of the Nordic Conference on Human–Computer Interaction (NordiCHI'06)*, Oslo, Norway. http://portal.acm.org/citation.cfm?id=1182482

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.

Albinsson, P.-A. and S. Zhai (2003). High precision touch screen interaction. Paper presented at the *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'03)*, Ft. Lauderdale, FL. http://dl.acm.org/citation.cfm?id=642631

Alexander, J., T. Han, W. Judd, P. Irani, and S. Subramanian (2012). Putting your best foot forward: Investigating real-world mappings for foot-based gestures. Paper presented at the *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'12)*, Austin, TX. http://dl.acm.org/citation.cfm?id=2208575

Anderson, J. R. (1980). Cognitive skills, Chapter 8. In *Cognitive Psychology and Its Implications*. San Francisco, CA: W. H. Freeman. pp. 222–254.

Anthony, L. and J. O. Wobbrock (2010). A lightweight multistroke recognizer for user interface prototypes. Paper presented at the *Proceedings of Graphics Interface (GI'10)*, Ottawa, Ontario, Canada. http://portal.acm.org/citation.cfm?id=1839214.1839258

Anthony, L. and J. O. Wobbrock (2012). $N-Protractor: A fast and accurate multistroke recognizer. Paper presented at the *Proceedings of Graphics Interface (GI'12)*, Toronto, Ontario, Canada.

Appert, C., O. Chapuis, and M. Beaudouin-Lafon (2008). Evaluation of pointing performance on screen edges. Paper presented at the *Proceedings of the ACM Working Conference on Advanced Visual Interfaces (AVI'08)*, Napoli, Italy. http://portal.acm.org/citation.cfm?id=1385569.1385590

Arons, B. (1993). SpeechSkimmer: Interactively skimming recorded speech. *UIST'93 Symposium on User Interface Software & Technology*, Atlanta, GA. pp. 187–195.

Balakrishnan, R. (2004). "Beating" Fitts' law: Virtual enhancements for pointing facilitation. *International Journal of Human–Computer Studies* **61**(6): 857–874.

Balakrishnan, R., T. Baudel, G. Kurtenbach, and G. Fitzmaurice (1997). The Rockin'Mouse: Integral 3D manipulation on a plane. *CHI'97 Conference on Human Factors in Computing Systems*, Atlanta, GA. pp. 311–318.

Balakrishnan, R., G. Fitzmaurice, and G. Kurtenbach (March 2001). User interfaces for volumetric displays. *IEEE Computer* **34**(3): 37–45.

Balakrishnan, R. and K. Hinckley (1999). The role of kinesthetic reference frames in two-handed input performance. *Proceedings of the ACM UIST'99 Symposium on User Interface Software and Technology*, Asheville, NC. pp. 171–178.

Balakrishnan, R. and K. Hinckley (2000). Symmetric bimanual interaction. *Proceedings of the CHI 2000*, The Hague, the Netherlands. pp. 33–40.

Balakrishnan, R. and G. Kurtenbach (1999). Exploring bimanual camera control and object manipulation in 3D graphics interfaces. *Proceedings of the CHI'99 ACM Conference on Human Factors in Computing Systems*, Pittsburgh, PA. pp. 56–63.

Balakrishnan, R. and I. S. MacKenzie (1997). Performance differences in the fingers, wrist, and forearm in computer input control. *Proceedings of the CHI'97 ACM Conference on Human Factors in Computing Systems*, Atlanta, GA. pp. 303–310.

Bates, R. and H. O. Istance (2003). Why are eye mice unpopular? A detailed comparison of head and eye controlled assistive technology pointing devices. *Universal Access in the Information Society* **2**(3): 280–290.

Bau, O. and W. E. Mackay (2008). OctoPocus: A dynamic guide for learning gesture-based command sets. Paper presented at the *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST'08)*, Monterey, CA. http://portal.acm.org/citation.cfm?id=1449715.1449724

Bau, O., I. Poupyrev, A. Israr, and C. Harrison (2010). TeslaTouch: Electrovibration for touch surfaces. Paper presented at the *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST'10)*, New York. http://dl.acm.org/citation.cfm?id=1866074

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. and G. Chu (2009). Back-of-device interaction allows creating very small touch devices. Paper presented at the *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'09)*, Boston, MA. http://portal.acm.org/citation.cfm?id=1518701.1518995

Baudisch, P., E. Cutrell, D. Robbins, M. Czerwinski, P. Tandler, B. Bederson, and A. Zierlinger (2003). Drag-and-pop and drag-and-pick: Techniques for accessing remote screen content on touch- and pen-operated systems. Paper presented at the *Proceedings of the 9th IFIP TC13 International Conference on Human–Computer Interaction (INTERACT'03)*, Zurich, Switzerland. http://www.patrickbaudisch.com/publications/2003-Baudisch-Interact03-DragAndPop.pdf

Baudisch, P., N. Good, V. Bellotti and P. Schraedley (2002). Keeping things in context: A comparative evaluation of focus plus context screens, overviews, and zooming. *CHI 2002 Conference on Human Factors in Computing Systems*, Minneapolis, MN. pp. 259–266.

Baudisch, P., X. Xie, C. Wang, and W.-Y. Ma (2004). Collapse-to-zoom: Viewing web pages on small screen devices by interactively removing irrelevant content. *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST'04)*, Santa Fe, NM, October 24–27, 2004. ACM Press, New York. pp. 91–94.

Bederson, B. (2000). Fisheye menus. *Proceedings of the UIST 2000*, San Diego, CA. pp. 217–226.

Bederson, B., J. Hollan, K. Perlin, J. Meyer, D. Bacon, and G. Furnas (1996). Pad++: A zoomable graphical sketchpad for exploring alternate interface physics. *Journal of Visual Languages and Computing* **7**: 3–31.

Bellotti, V., M. Back, W. K. Edwards, R. Grinter, C. Lopes, and A. Henderson (2002). Making sense of sensing systems: Five questions for designers and researchers. *Proceedings of the ACM CHI 2002 Conference on Human Factors in Computing Systems*, Minneapolis, MN. pp. 415–422. http://portal.acm.org/citation.cfm?id=503376.503450

Benko, H., T. S. Saponas, D. Morris, and D. Tan (2009). Enhancing input on and above the interactive surface with muscle sensing. Paper presented at the *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, Banff, Alberta, Canada.

Betrisey, C., J. Blinn, B. Dresevic, B. Hill, G. Hitchcock, B. Keely, D. Mitchell, J. Platt, and T. Whitted (2000). Displaced filtering for patterned displays. *Proceedings of the Society for Information Display Symposium*, Redmond, WA. pp. 296–299.

Bier, E., M. Stone, K. Pier, W. Buxton, and T. DeRose (1993). Toolglass and magic lenses: The see-through interface. *Proceedings of SIGGRAPH'93*, Anaheim, CA. pp. 73–80.

Bilton, N. (December 6, 2011). TouchFire adds a physical keyboard to iPad screen. *The New York Times*. Retrieved from http://gadgetwise.blogs.nytimes.com/2011/12/06/touchfire-adds-a-physical-keyboard-to-ipad-screen/

Blanch, R., Y. Guiard, and M. Beaudouin-Lafon (2004). Semantic pointing: Improving target acquisition with control-display ratio adaptation. Paper presented at the *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'04)*, Vienna, Austria. http://portal.acm.org/citation.cfm?id=985692.985758

Bolt, R. (1980). Put-that-there: Voice and gesture at the graphics interface. *ACM SIGGRAPH Computer Graphics* **14**(3): 262–270.

Bragdon, A., R. Zeleznik, B. Williamson, T. Miller, and J. J. LaViola (2009). GestureBar: Improving the approachability of gesture-based interfaces. Paper presented at the *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'09)*, Boston, MA. http://portal.acm.org/citation.cfm?id=1518701.1519050

Brewster, S. A., P. C. Wright, and A. D. N. Edwards (1994). The design and evaluation of an auditory-enhanced scrollbar. *Conference Proceedings on Human Factors in Computing Systems*, Boston, MA. pp. 173–179.

Britton, E., J. Lipscomb, and M. Pique (1978). Making nested rotations convenient for the user. *Computer Graphics* **12**(3): 222–227.

Brooks, Jr., F. P. (1988). Grasping reality through illusion: Interactive graphics serving science. *Proceedings of CHI'88: ACM Conference on Human Factors in Computing Systems*, Washington, DC. ACM Press, New York. pp. 1–11.

Bukowski, R. and C. Sequin (1995). Object associations: A simple and practical approach to virtual 3D manipulation. *ACM 1995 Symposium on Interactive 3D Graphics*, Monterey, CA. pp. 131–138.

Burdea, G. (1996). *Force and Touch Feedback for Virtual Reality*. New York: John Wiley & Sons.

Buxton, W. (1983). Lexical and pragmatic considerations of input structure. *Computer Graphics* **17**(1): 31–37.

Buxton, W. (1986). Chunking and phrasing and the design of human–computer dialogues. *Information Processing '86, Proceedings of the IFIP 10th World Computer Congress*, Dublin, Ireland. North Holland Publishers, Amsterdam, the Netherlands. pp. 475–480.

Buxton, W. (1990a). The pragmatics of haptic input. *Proceedings of CHI'90: ACM Conference on Human Factors in Computing Systems: Tutorial 26 Notes*, Seattle, WA. ACM Press, New York.

Buxton, W. (1990b). A three-state model of graphical input. *Proceedings of the INTERACT'90*, Cambridge, U.K. Elsevier Science, Amsterdam, the Netherlands. pp. 449–456.

Buxton, W. (1995a). Integrating the periphery and context: A new taxonomy of telematics. *Proceedings of Graphics Interface '95*, Quebec City, Quebec, Canada. pp. 239–246.

Buxton, W. (1995b). Speech, language and audition. In R. Baecker, J. Grudin, W. Buxton, and S. Greenberg (eds.), *Readings in Human–Computer Interaction: Toward the Year 2000*. Los Altos, CA: Morgan Kaufmann Publishers. pp. 525–537.

Buxton, W. and G. Fitzmaurice (1998). HMDs, caves & chameleon: A human-centric analysis of interaction in virtual space. *Computer Graphics* **32**(8): 69–74.

Buxton, W., G. Fitzmaurice, R. Balakrishnan, and G. Kurtenbach (July/August 2000). Large displays in automotive design. *IEEE Computer Graphics and Applications* **20**(4): pp. 68–75.

Buxton, W., E. Fiume, R. Hill, A. Lee, and C. Woo (1983). Continuous hand-gesture driven input. *Proceedings of Graphics Interface (GI'83)*, Edmonton, Alberta, Canada. pp. 191–195. http://www.billbuxton.com/gesture83.html

Buxton, W., R. Hill, and P. Rowley (1985). Issues and techniques in touch-sensitive tablet input. *Computer Graphics* **19**(3): 215–224; *Proceedings of the ACM Conference on Computer Graphics and Interactive Techniques (SIGGRAPH'85)*, San Francisco, CA. http://portal.acm.org/citation.cfm?id=325239

Buxton, W. and B. Myers (1986). A study in two-handed input. *Proceedings of CHI'86: ACM Conference on Human Factors in Computing Systems*, Boston, MA. ACM Press, New York. pp. 321–326.

Buyukkokten, O., H. Garcia-Molina, and A. Paepcke (2001). Accordion summarization for end-game browsing on PDAs and cellular phones. *ACM CHI 2001 Conference on Human Factors in Computing Systems*, Seattle, WA.

Cadoz, C. (1994). *Les realites virtuelles*. Paris, France: 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. *Proceedings of INTERACT'99: 7th IFIP Conference on Human Computer Interaction*, Edinburgh, Scotland. pp. 383–390.

Cao, X., A. D. Wilson, R. Balakrishnan, K. Hinckley, and S. E. Hudson (2008). ShapeTouch: Leveraging contact shape on interactive surfaces. Paper presented at the *Proceedings of the 3rd IEEE Workshop on Horizontal Interactive Human–Computer Systems (Tabletop'08)*, Amsterdam, the Netherlands. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4660195

Card, S., W. English, and B. 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, 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, S., J. Mackinlay, and B. Shneiderman (1999). *Readings in Information Visualization: Using Vision to Think*. San Francisco, CA: Morgan Kaufmann.

Card, S., T. Moran, and A. Newell (1980). The keystroke-level model for user performance time with interactive systems. *Communications of the ACM* **23**(7): 396–410.

Card, S. K., T. P. Moran, and A. Newell (1983). *The Psychology of Human–Computer Interaction*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Casiez, G., D. Vogel, R. Balakrishnan, and A. Cockburn (2008). The impact of control-display gain on user performance in pointing tasks. *Human–Computer Interaction* **23**(3): 215–250. doi: 10.1080/07370020802278163.

Cassell, J. (2003). A framework for gesture generation and interpretation. In R. Cipolla and A. Pentland (eds.), *Computer Vision in Human–Machine Interaction*. Cambridge, U.K.: Cambridge University Press.

Caudell, T. P. and D. W. Mizell (1992). Augmented reality: An application of heads-up display technology to manual manufacturing processes. *Proceedings of the HICCS'92*, Honolulu, HI.

Chance, S., F. Gaunet, A. Beall, and J. Loomis (1998). Locomotion mode affects the updating of objects encountered during travel: The contribution of vestibular and proprioceptive inputs to path integration. *Presence* **7**(2): 168–178.

Chapuis, O., J.-B. Labrune, and E. Pietriga (2009). DynaSpot: Speed-dependent area cursor. Paper presented at the *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'09)*, Boston, MA. http://portal.acm.org/citation.cfm?doid=1518701.1518911

Chen, M., S. J. Mountford, and A. Sellen (1988). A study in interactive 3-D rotation using 2-D control devices. *Computer Graphics* **22**(4): 121–129.

Cholewiak, R. and A. Collins (1991). Sensory and physiological bases of touch. In M. Heller and W. Schiff (eds.), *The Psychology of Touch*. Hillsdale, NJ: Lawrence Erlbaum Associates. pp. 23–60.

Christ, R. E. (1975). Review and analysis of color coding research for visual displays. *Human Factors* **25**: 71–84.

Cockburn, A. and A. Firth (2003). Improving the acquisition of small targets. Paper presented at the *Proceedings of the British HCI Group Annual Conference (HCI'03)*, Bath, England. http://citeseer.ist.psu.edu/574125.html

Cohen, P. R., M. Johnston, D. McGee, S. Oviatt, J. Pittman, I. Smith, L. Chen, and J. Clow (1997). QuickSet: Multimodal interaction for distributed applications. *ACM Multimedial 97*, Seattle, WA.

Cohen, P. R. and J. W. Sullivan (1989). Synergistic use of direct manipulation and natural language. *Proceedings of the ACM CHI'89 Conference on Human Factors in Computing Systems*, Austin, TX. pp. 227–233.

Cohn, G., D. Morris, S. N. Patel, and D. S. Tan (2012). Humantenna: Using the body as an antenna for real-time whole-body interaction. Paper presented at the *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'12)*, Austin, TX. http://dl.acm.org/citation.cfm?id=2208330

Cole, W. G. (1986). Medical cognitive graphics. *ACM CHI'86 Conference on Human factors in Computing Systems*, Boston, MA. pp. 91–95.

Conner, D., S. Snibbe, K. Herndon, D. Robbins, R. Zeleznik, and A. van Dam (1992). Three-dimensional widgets. *Computer Graphics (Proceedings of the 1992 Symposium on Interactive 3D Graphics)*, Redwood City, CA. pp. 183–188, 230–231.

Cook, R. L. (1986). Stochastic sampling in computer graphics. *ACM Transactions on Graphics* **5**(1): 51–72.

Cowan, W. B. (1983). An inexpensive calibration scheme for calibrations of a color monitor in terms of CIE standard coordinates. *Computer Graphics* **17**(3): 315–321.

Czerwinski, M., G. Smith, T. Regan, B. Meyers, G. Robertson, and G. Starkweather (2003). Toward characterizing the productivity benefits of very large displays. *INTERACT 2003*, Zurich, Switzerland.

Czerwinski, M., D. S. Tan, and G. G. Robertson (2002). Women take a wider view. *Proceedings of the ACM CHI 2002 Conference on Human Factors in Computing Systems*, Minneapolis, MN. pp. 195–202.

Darken, R. P. and J. L. Sibert (1993). A toolset for navigation in virtual environments. *Proceedings of the 6th annual ACM symposium on User interface software and technology*, pp. 157–165. ACM, New York.

Darken, R. P. and J. L. Sibert (October 1995). Navigating large virtual spaces. *International Journal of Human–Computer Interaction* **8**(1): 49–72.

Deatherage, B. H. (1972). Auditory and other sensory forms of information presentation. In H. Van Cott and R. Kinkade (eds.), *Human Engineering Guide to Equipment Design*. Washington, DC: U.S. Government Printing Office.

De Jong, J. R. (1957). The effects of increasing skill on cycle time and its consequences for time standards. *Ergonomics* **1**: 51–60.

Deering, M. (1992). High resolution virtual reality. *Computer Graphics* **26**(2): 195–202.

Dey, A., G. Abowd, and D. Salber (2001). A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. *Journal of Human–Computer Interaction* **16**(2–4): 97–166.

Dixon, M., J. Fogarty, and J. O. Wobbrock (2012). A general-purpose target-aware pointing enhancement using pixel-level analysis of graphical interfaces. Paper presented at the *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'12)*, Austin, TX. http://dl.acm.org/citation.cfm?id=2208734

Doll, T. J. and D. J. Folds (1985). Auditory signals in military aircraft: Ergonomic principles versus practice. *Proceedings of the 3rd Symposium on Aviation Psychology*, Ohio State University, Department of Aviation, Colombus, OH. pp. 111–125.

Douglas, S., A. Kirkpatrick, and I. S. MacKenzie (1999). Testing pointing device performance and user assessment with the ISO 9241, Part 9 Standard. *Proceedings of the ACM CHI'99 Conference on Human Factors in Computing Systems*, Pittsburgh, PA. pp. 215–222.

Douglas, S. and A. Mithal (1994). The effect of reducing homing time on the speed of a finger-controlled isometric pointing device. *Proceedings of the ACM CHI'94 Conference on Human Factors in Computing Systems*, Boston, MA. pp. 411–416.

Efron, D. (1941). *Gesture and Environment*. New York: King's Crown Press.

Evans, A. C. and J. O. Wobbrock (2012). Taming wild behavior: The Input Observer for obtaining text entry and mouse pointing measures from everyday computer use. *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'12)*, Austin, TX, May 5–10, 2012. ACM Press, New York. pp. 1947–1956.

Farris, J. S., K. S. Jones, and B. A. Anders (2001). Acquisition speed with targets on the edge of the screen: An application of Fitts' Law to commonly used web browser controls. Paper presented at the *Proceedings of the Human Factors and Ergonomics Society 45th Annual Meeting (HFES'01)*, Minneapolis, MN.

Favalora, G. E. (2005). Volumetric 3D displays and application infrastructure. *IEEE Computer* **38**(8): 37–44.

Feiner, S., B. Macintyre, and D. Seligmann (1993). Knowledge-based augmented reality. *Communications of the ACM* **36**(7): 53–61.

Findlater, L., A. Jansen, K. Shinohara, M. Dixon, P. Kamb, J. Rakita, and J. O. Wobbrock (2010). Enhanced area cursors: Reducing fine-pointing demands for people with motor impairments. Paper presented at the *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST'10)*, New York.

Fitts, P. (1954). The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology* **47**(6): 381–391.

Fitzmaurice, G. W., R. Balakrisnan, and G. Kurtenbach (1999). Sampling, synthesis, and input devices. *Communications of the ACM* **42**(8): 54–63.

Fitzmaurice, G. and W. Buxton (1997). An empirical evaluation of graspable user interfaces: Towards specialized, space-multiplexed input. *Proceedings of CHI'97: ACM Conference on Human Factors in Computing Systems*, Atlanta, GA. ACM Press, New York. pp. 43–50.

Fitzmaurice, G., H. Ishii, and W. Buxton (1995). Bricks: Laying the foundations for graspable user interfaces. *Proceedings of CHI'95: ACM Conference on Human Factors in Computing Systems*, Denver, CO. ACM Press, New York. pp. 442–449.

Flash, T. and N. Hogan (1985). The coordination of arm movements: An experimentally confirmed mathematical model. *The Journal of Neuroscience* **5**(7): 1688–1703.

Foley, J. D., V. Wallace, and P. Chan (November 1984). The human factors of computer graphics interaction techniques. *IEEE Computer Graphics and Applications* **4**(11): 13–48.

Forlines, C., D. Vogel, and R. Balakrishnan (2006). HybridPointing: Fluid switching between absolute and relative pointing with a direct input device. Paper presented at the *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology*, Montreux, Switzerland.

Fothergill, S., H. Mentis, P. Kohli, and S. Nowozin (2012). Instructing people for training gestural interactive systems. Paper presented at the *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'12)*, Austin, TX. http://dl.acm.org/citation.cfm?id=2208303

Freeman, D., H. Benko, M. R. Morris, and D. Wigdor (2009). ShadowGuides: Visualizations for in-situ learning of multi-touch and whole-hand gestures. Paper presented at the *Proceedings of the ACM Conference on Interactive Tabletops and Surfaces (ITS'09)*, Calgary, Alberta, Canada. http://dl.acm.org/citation.cfm?id=1731903.1731935

Freeman, W. T. and C. Weissman (1995). Television control by hand gestures. *International Workshop on Automatic Face and Gesture Recognition*, Zurich, Switzerland. pp. 179–183.

Froehlich, J., J. O. Wobbrock, and S. K. Kane (2007). Barrier pointing: Using physical edges to assist target acquisition on mobile device touch screens. Paper presented at the *Proceedings of the ACM SIGACCESS Conference on Computers and Accessibility (ASSETS'07)*, Tempe, AZ. http://portal.acm.org/citation.cfm?id=1296843.1296849

Funkhouser, T. and K. Li (July–August 2000). Large format displays. *IEEE Computer Graphics and Applications* **20**(4, special issue): 20–75.

Gaver, W. (1989). The SonicFinder: An interface that uses auditory icons. *Human–Computer Interaction* **4**(1): 67–94.

Gibson, J. (1986). *The Ecological Approach to Visual Perception*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Goel, M., J. O. Wobbrock, and S. N. Patel (2012). GripSense: Using built-in sensors to detect hand posture and pressure on commodity mobile phones. *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST'12)*, Cambridge, MA, October 7–10, 2012. ACM Press, New York.

Goldberg, D. and C. Richardson (1993). Touch-typing with a stylus. *Proceedings of the INTERCHI'93 Conference on Human Factors in Computing Systems*, Amsterdam, the Netherlands. pp. 80–87. http://portal.acm.org/citation.cfm?id=169093

Grandhi, S. A., G. Joue, and I. Mittelberg (2011). Understanding naturalness and intuitiveness in gesture production: Insights for touchless gestural interfaces. Paper presented at the *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'11)*, Vancouver, British Columbia, Canada. http://dl.acm.org/citation.cfm?id=1979061

Grossman, T. and R. Balakrishnan (2005). The Bubble Cursor: Enhancing target acquisition by dynamic resizing of the cursor's activation area. Paper presented at the *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'05)*, Portland, OR. http://portal.acm.org/citation.cfm?id=1054972.1055012

Grossman, T. and R. Balakrishnan (2006a). The design and evaluation of selection techniques for 3D volumetric displays. Paper presented at the *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology*, Montreux, Switzerland.

Grossman, T. and R. Balakrishnan (2006b). An evaluation of depth perception on volumetric displays. Paper presented at the *Proceedings of the Working Conference on Advanced Visual Interfaces*, Venezia, Italy.

Grossman, T. and R. Balakrishnan (2008). Collaborative interaction with volumetric displays. Paper presented at the *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Florence, Italy.

Grossman, T., K. Hinckley, P. Baudisch, M. Agrawala, and R. Balakrishnan (2006). Hover Widgets: Using the tracking state to extend the capabilities of pen-operated devices. Paper presented at the *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'06)*, Montréal, Québec, Canada. http://portal.acm.org/citation.cfm?id=1124772.1124898

Grossman, T., D. Wigdor, and R. Balakrishnan (2004). Multi-finger gestural interaction with 3d volumetric displays. Paper presented at the *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology*, Santa Fe, NM.

Green, M. and J. Liang (1994). JDCAD: A highly interactive 3D modeling system. *Computers and Graphics* **18**(4): 499–506.

Grudin, J. (2001). Partitioning digital worlds: Focal and peripheral awareness in multiple monitor use. *Proceedings of the CHI 2001*, Seattle, WA. pp. 458–465.

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, Y. (2009). The problem of consistency in the design of Fitts' law experiments: Consider either target distance and width or movement form and scale. *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'09)*, Boston, MA, April 4–9, 2009. ACM Press, New York. pp. 1809–1818.

Guiard, Y., R. Blanch, and M. Beaudouin-Lafon (2004). Object pointing: A complement to bitmap pointing in GUIs. Paper presented at the *Proceedings of Graphics Interface (GI'04)*, London, Ontario, Canada. http://portal.acm.org/citation.cfm?id=1006060

Guiard, Y., F. Buourgeois, D. Mottet, and M. Beaudouin-Lafon (2001). Beyond the 10-bit barrier: Fitts' Law in multi-scale electronic worlds. *IHM-HCI 2001*, Lille, France.

Guimbretiere, F., M. C. Stone, and T. Winograd (2001). Fluid interaction with high-resolution wall-size displays. *Proceedings of the UIST 2001 Symposium on User Interface Software and Technology*, Orlando, FL. pp. 21–30.

Harada, S., J. A. Landay, J. Malkin, X. Li, and J. A. Bilmes (2006). The vocal joystick: Evaluation of voice-based cursor control techniques. Paper presented at the *Proceedings of the ACM SIGACCESS Conference on Computers and Accessibility (ASSETS'06)*, Portland, OR. http://portal.acm.org/citation.cfm?id=1169021

Harada, S., T. S. Saponas, and J. A. Landay (2007). VoicePen: Augmenting pen input with simultaneous non-linguistic vocalization. Paper presented at the *Proceedings of the ACM International Conference on Multimodal Interfaces (ICMI'07)*, Nagoya, Aichi, Japan. http://dl.acm.org/citation.cfm?id=1322192.1322225

Harada, S., J. O. Wobbrock, and J. A. Landay (2007). VoiceDraw: A hands-free voice-driven drawing application for people with motor impairments. Paper presented at the *Proceedings of the ACM SIGACCESS Conference on Computers and Accessibility (ASSETS'07)*, Tempe, AZ. http://portal.acm.org/citation.cfm?id=1296843.1296850

Harada, S., J. O. Wobbrock, and J. A. Landay (2011). VoiceGames: Investigation into the use of non-speech voice input for making computer games more accessible. Paper presented at the *Proceedings of the 13th IFIP TC13 International Conference on Human–Computer Interaction (INTERACT'11)*, Lisbon, Portugal. http://www.springerlink.com/content/p7557762w547064p/?MUD=MP

Harrison, B., K. Fishkin, A. Gujar, C. Mochon, and R. Want (1998). Squeeze me, hold me, tilt me! An exploration of manipulative user interfaces. *Proceedings of the ACM CHI'98 Conference on Human Factors in Computing Systems*, Los Angeles, CA. pp. 17–24.

Harrison, B., H. Ishii, K. Vicente, and W. Buxton (1995a). Transparent layered user interfaces: An evaluation of a display design to enhance focused and divided attention. *Proceedings of CHI'95: ACM Conference on Human Factors in Computing Systems*, Denver, CO. pp. 317–324.

Harrison, B., G. Kurtenbach, and K. Vicente (1995b). An experimental evaluation of transparent user interface tools and information content. *UIST'95*, Pittsburgh, PA. pp. 81–90.

Harrison, C., Schwarz, J., and Hudson, S. E. (2011). TapSense: Enhancing finger interaction on touch surfaces. Paper presented at the *Proceedings of the ACM Symposium on User Interface Software and Technology* (UIST '11), Santa Barbara, CA. http://dl.acm.org/citation.cfm?id=2047279

Harrison, B. and K. Vicente (1996). An experimental evaluation of transparent menu usage. *Proceedings of CHI'96: ACM Conference on Human Factors in Computing Systems*, Vancouver, British Columbia, Canada. pp. 391–398.

Hauptmann, A. (1989). Speech and gestures for graphic image manipulation. *Proceedings of CHI'89: ACM Conference on Human Factors in Computing Systems*, Austin, TX. ACM Press, New York. pp. 241–245.

Henry, T. R., S. E. Hudson, and G. L. Newell (1990). Integrating gesture and snapping into a user interface toolkit. Paper presented at the *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST'90)*, Snowbird, UT. http://portal.acm.org/citation.cfm?id=97938

Hinckley, K. (2003a). Distributed sensing techniques for face-to-face collaboration. *ICMI-PUI'03 Fifth International Conference on Multimodal Interfaces*, Vancouver, British Columbia, Canada.

Hinckley, K. (2003b). Synchronous gestures for multiple users and computers. *UIST'03 Symposium on User Interface Software & Technology*, Vancouver, British Columbia, Canada. 10pp. (full paper).

Hinckley, K., P. Baudisch, G. Ramos, and F. Guimbretière (2005). Design and analysis of delimiters for selection-action pen gesture phrases in Scriboli. Paper presented at the *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'05)*, Portland, OR. http://portal.acm.org/citation.cfm?id=1055035

Hinckley, K., E. Cutrell, S. Bathiche, and T. Muss (2001). Quantitative analysis of scrolling techniques. *CHI 2002*, Minneapolis, MN.

Hinckley, K., M. Czerwinski, and M. Sinclair (1998a). Interaction and modeling techniques for desktop two-handed input. *Proceedings of the ACM UIST'98 Symposium on User Interface Software and Technology*, San Francisco, CA. ACM, New York. pp. 49–58.

Hinckley, K., F. Guimbretière, M. Agrawala, G. Apitz, and N. Chen (2006). Phrasing techniques for multi-stroke selection gestures. Paper presented at the *Proceedings of Graphics Interface (GI '06)*, Quebec City, Quebec, Canada. http://dl.acm.org/citation.cfm?id=1143104

Hinckley, K., R. Pausch, J. C. Goble, and N. F. Kassell (1994a). Passive real-world interface props for neurosurgical visualization. *Proceedings of CHI'94: ACM Conference on Human Factors in Computing Systems*, Boston, MA. ACM Press, New York. pp. 452–458.

Hinckley, K., R. Pausch, J. C. Goble, and N. F. Kassell (1994b). A survey of design issues in spatial input. *Proceedings of the ACM UIST'94 Symposium on User Interface Software and Technology*, Marina del Rey, CA. ACM Press, New York. pp. 213–222.

Hinckley, K., R. Pausch, D. Proffitt, and N. Kassell (1998b). Two-handed virtual manipulation. *ACM Transactions on Computer–Human Interaction* **5**(3): 260–302.

Hinckley, K., J. Pierce, E. Horvitz, and M. Sinclair (2003). Foreground and background interaction with sensor-enhanced mobile devices. *ACM TOCHI* (submitted for review) (Special Issue on Sensor-Based Interaction).

Hinckley, K., J. Pierce, M. Sinclair, and E. Horvitz (2000). Sensing techniques for mobile interaction. *ACM UIST 2000 Symposium on User Interface Software & Technology*, San Diego, CA. pp. 91–100.

Hinckley, K., G. Ramos, F. Guimbretiere, P. Baudisch, and M. Smith (2004). Stitching: Pen gestures that span multiple displays. Paper presented at the *Proceedings of the ACM Working Conference on Advanced Visual Interfaces (AVI'04)*, Gallipoli, Italy. http://portal.acm.org/citation.cfm?id=989866

Hinckley, K. and M. Sinclair (1999). Touch-sensing input devices. *ACM CHI'99 Conference on Human Factors in Computing Systems*, Pittsburgh, PA. pp. 223–230.

Hinckley, K., M. Sinclair, E. Hanson, R. Szeliski, and M. Conway (1999). The VideoMouse: A camera-based multi-degree-of-freedom input device. *ACM UIST'99 Symposium on User Interface Software & Technology*, Asheville, NC. pp. 103–112.

Hinckley, K., J. Tullio, R. Pausch, D. Proffitt, and N. Kassell (1997). Usability analysis of 3D rotation techniques. *Proceedings of the ACM UIST'97 Symposium on User Interface Software and Technology*, Banff, Alberta, Canada. ACM Press, New York. pp. 1–10.

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 presented at the *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'07)*, San Jose, CA.

Hiraoka, S., I. Miyamoto, and K. Tomimatsu (2003). Behind touch: A text input method for mobile phone by the back and tactile sense interface. Paper presented at the *Proceedings of Interaction 2003*, Tokyo, Japan. http://sciencelinks.jp/j-east/article/200405/000020040504A0091046.php

Hirshfield, L. M., K. Chauncey, R. Gulotta, A. Girouard, E. T. Solovey, R. J. Jacob, and S. Fantini (2009a). Combining electroencephalograph and functional near infrared spectroscopy to explore users' mental workload. Paper presented at the *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.

Hirshfield, L. M., E. T. Solovey, A. Girouard, J. Kebinger, R. J. K. Jacob, A. Sassaroli, and S. Fantini (2009b). Brain measurement for usability testing and adaptive interfaces: An example of uncovering syntactic workload with functional near infrared spectroscopy. Paper presented at the *Proceedings of the 27th International Conference on Human Factors in Computing Systems*, Boston, MA.

Hix, D., J. Templeman, and R. Jacob (1995). Pre-screen projection: From concept to testing of a new interaction technique. *Proceedings of the CHI'95*, Denver, CO. pp. 226–233.

Holz, C. and P. Baudisch (2010). The generalized perceived input point model and how to double touch accuracy by extracting fingerprints. Paper presented at the *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'10)*, Atlanta, GA. http://dl.acm.org/citation.cfm?id=1753413

Holz, C. and P. Baudisch (2011). Understanding touch. Paper presented at the *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'11)*, Vancouver, British Columbia, Canada. http://dl.acm.org/citation.cfm?id=1979308

Honan, M., E. Serina, R. Tal, and D. Rempel (1995). Wrist postures while typing on a standard and split keyboard. *Proceedings of the Human Factors and Ergonomics Society (HFES) 39th Annual Meeting*, Santa Monica, CA. pp. 366–368.

Horvitz, E., J. Breese, D. Heckerman, D. Hovel, and K. Rommelse (July 1998). The Lumiere project: Bayesian user modeling for inferring the goals and needs of software users. *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, Madison, WI. pp. 256–265. Morgan Kaufmann, San Francisco, CA.

Horvitz, E., A. Jacobs, and D. Hovel (1999). Attention-sensitive alerting. *Proceedings of UAI '99, Conference on Uncertainty and Artificial Intelligence*, Stockholm, Sweden. pp. 305–313.

Hourcade, J. P., K. B. Perry, and A. Sharma (2008). PointAssist: Helping four year olds point with ease. Paper presented at the *Proceedings of the ACM Conference on Interaction Design and Children (IDC '08)*, Chicago, IL. http://portal.acm.org/citation.cfm?id=1463689.1463757

Hudson, S. and I. Smith (1996). Electronic mail previews using non-speech audio. *CHI'96 Companion Proceedings*, Vancouver, British Columbia, Canada. pp. 237–238.

Hurst, A., J. Mankoff, A. K. Dey, and S. E. Hudson (2007). Dirty desktops: Using a patina of magnetic mouse dust to make common interactor targets easier to select. Paper presented at the *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST'07)*, Newport, RI. http://portal.acm.org/citation.cfm?id=1294242

Hwang, F., S. Keates, P. Langdon, and P. J. Clarkson (2003). Multiple haptic targets for motion-impaired computer users. Paper presented at the *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'03)*, Ft. Lauderdale, FL. http://portal.acm.org/citation.cfm?id=642620

Igarashi, T. and J. F. Hughes (2001). Voice as sound: Using non-verbal voice input for interactive control. Paper presented at the *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST'01)*, Orlando, FL. http://portal.acm.org/citation.cfm?id=502348.502372

Igarashi, T., S. Matsuoka, and H. Tanaka (1999). Teddy: A sketching interface for 3D freeform design. *ACM SIGGRAPH'99*, Los Angeles, CA. pp. 409–416.

Ishii, H. and B. Ullmer (1997). Tangible bits: Towards seamless interfaces between people, bits, and atoms. *Proceedings of CHI'97: ACM Conference on Human Factors in Computing Systems*, Atlanta, GA. ACM Press, New York. pp. 234–241.

Iwata, H. (1990). Artificial reality with force-feedback: Development of desktop virtual space with compact master manipulator. *Computer Graphics* **24**(4): 165–170.

Jacob, R. (1991). The use of eye movements in human–computer interaction techniques: What you look at is what you get. *ACM Transactions on Information Systems* **9**(3): 152–169.

Jacob, R., L. Sibert, D. McFarlane, and M. Mullen, Jr. (1994). Integrality and separability of input devices. *ACM Transactions on Computer–Human Interaction* **1**(1): 3–26.

Jansen, Y., P. Dragicevic, and J.-D. Fekete (2012). Tangible remote controllers for wall-size displays. Paper presented at the *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'12)*, Austin, TX. http://dl.acm.org/citation.cfm?id=2208691

Jansen, A., L. Findlater, and J. O. Wobbrock (2011). From the lab to the world: Lessons from extending a pointing technique for real-world use. Paper presented at the *Extended Abstracts of the ACM Conference on Human Factors in Computing Systems (CHI'11)*, Vancouver, British Columbia, Canada. http://portal.acm.org/citation.cfm?id=1979888

Jansen, Y., T. Karrer, and J. Borchers (2010). MudPad: Tactile feedback and haptic texture overlay for touch surfaces. Paper presented at the *Proceedings of the ACM Conference on Interactive Tabletops and Surfaces (ITS '10)*, Saarbrücken, Germany. http://dl.acm.org/citation.cfm?id=1936655

Jellinek, H. and S. Card (1990). Powermice and user performance. *Proceedings of the ACM CHI'90 Conference on Human Factors in Computing Systems*, Seattle, WA. pp. 213–220.

Jacob, R. J. K. (1990). What you look at is what you get: Eye movement-based interaction techniques. *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'90)*, Seattle, WA, April 1–5. ACM Press, New York. pp. 11–18.

Jojic, N., B. Brumitt, B. Meyers, and S. Harris (2000). Detecting and estimating pointing gestures in dense disparity maps. *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition*, Grenoble, France.

Johnson, G., M. D. Gross, J. Hong, and E. Y.-L. Do (2009). Computational support for sketching in design: A review. *Foundations and Trends in Human–Computer Interaction* **2**(1): 1–93. doi: 10.1561/1100000013.

Jones, M., G. Marsden, N. Mohd-Nasir, K. Boone, and G. Buchanan (1999). Improving Web interaction on small displays. *Computer Networks* **31**(11–16): 1129–1137.

Kabbash, P. and W. Buxton (1995). The "Prince" technique: Fitts' law and selection using area cursors. Paper presented at the *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'95)*, Denver, CO. http://portal.acm.org/citation.cfm?id=223939; http://www.billbuxton.com/prince.html

Kabbash, P., W. Buxton, and A. Sellen (1994). Two-handed input in a compound task. *Proceedings of CHI'94: ACM Conference on Human Factors in Computing Systems*, Boston, MA. ACM Press, New York. pp. 417–423.

Kamba, T., S. A. Elson, T. Harpold, T. Stamper, and P. Sukaviriya (1996). Using small screen space more efficiently. *Conference Proceedings on Human Factors in Computing Systems*, Vancouver, British Columbia, Canada. 383p.

Kane, S. K., J. P. Bigham, and J. O. Wobbrock (2008). Slide Rule: Making mobile touch screens accessible to blind people using multi-touch interaction techniques. *Proceedings of the ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '08)*, Halifax, Nova Scotia, Canada, October 13–15. ACM Press, New York. pp. 73–80.

Kane, S. K., J. O. Wobbrock, and R. E. Ladner (2011). Usable gestures for blind people: Understanding preference and performance. Paper presented at the *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'11)*, Vancouver, British Columbia, Canada.

Karat, C., C. Halverson, D. Horn, and J. Karat (1999). Patterns of entry and correction in large vocabulary continuous speech recognition systems. *Proceedings of the ACM CHI'99 Conference on Human Factors in Computing Systems*, Pittsburgh, PA. pp. 568–575.

Karlson, A. K. and B. B. Bederson (2007). ThumbSpace: Generalized one-handed input for touchscreen-based mobile devices. Paper presented at the *Proceedings of the 11th IFIP TC13 International Conference on Human–Computer Interaction (INTERACT'07)*, Rio de Janeiro, Brazil. http://www.springerlink.com/content/p85w127j007153k1

Keates, S., F. Hwang, P. Langdon, P. J. Clarkson, and P. Robinson (2002). Cursor measures for motion-impaired computer users. *Proceedings of the ACM SIGCAPH Conference on Assistive Technologies (ASSETS'02)*, July 8–10, 2002, Edinburgh, Scotland. ACM Press, New York. pp. 135–142.

Kendon, A. (1988). How gestures can become like words. In F. Poyatos (ed.), *Cross Cultural Perspectives in Nonverbal Communication*. Toronto, Ontario, Canada: C. J. Hogrefe. pp. 131–141

Kirsh, D. (1995a). Complementary strategies: Why we use our hands when we think. *Proceedings of 7th Annual Conference of the Cognitive Science Society*, Basel, Switzerland. Lawrence Erlbaum Associates, Hillsdale, NJ. pp. 212–217.

Kirsh, D. (1995b). The intelligent use of space. *Artificial Intelligence* **73**: 31–68.

Kirsh, D. and P. Maglio (1994). On distinguishing epistemic from pragmatic action. *Cognitive Science* **18**(4): 513–549.

Költringer, T. and T. Grechenig (2004). Comparing the immediate usability of Graffiti 2 and virtual keyboard. Paper presented at the *Extended Abstracts of the ACM Conference on Human Factors in Computing Systems (CHI'04)*, Vienna, Austria. http://portal.acm.org/citation.cfm?id=986017

Kramer, A. (1994). Translucent patches—Dissolving windows. *Proceedings of the ACM UIST'94 Symposium on User Interface Software & Technology*, Marina del Rey, CA. pp. 121–130.

Kumar, M., A. Paepcke, and T. Winograd (2007). EyePoint: Practical pointing and selection using gaze and keyboard. *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'07)*, San Jose, CA, April 28–May 3, 2007. ACM Press, New York. pp. 421–430.

Kurihara, K., Goto, M., Ogata, J., and Igarashi, T. (2006). Speech Pen: Predictive handwriting based on ambient multimodal recognition. *Paper presented at the Proceedings of the ACM Conference on Human Factors in Computing Systems* (CHI'05), Montréal, Québec, Canada. http://portal.acm.org/citation.cfm?id=1124772.1124897

Kurtenbach, G. and W. Buxton (1991a). GEdit: A test bed for editing by contiguous gesture. *SIGCHI Bulletin* **23**(2): 22–26.

Kurtenbach, G. and W. Buxton (1991b). Issues in combining marking and direct manipulation techniques. *Proceedings of UIST'91*, Hilton Head, SC. pp. 137–144.

Kurtenbach, G. and W. Buxton (1993). The limits of expert performance using hierarchic marking menus. *Proceedings of the INTERCHI'93*, Amsterdam, the Netherlands. pp. 482–487.

Kurtenbach, G., G. Fitzmaurice, T. Baudel, and B. Buxton (1997). The design of a GUI paradigm based on tablets, two-hands, and transparency. *Proceedings of CHI'97: ACM Conference on Human Factors in Computing Systems*, Atlanta, GA. ACM, New York. pp. 35–42.

Kurtenbach, G., A. Sellen, and W. Buxton (1993). An empirical evaluation of some articulatory and cognitive aspects of 'marking menus'. *Journal of Human Computer Interaction* **8**(1): 1–23.

Landay, J. and Myers, B. A. (1993). Extending an existing user interface toolkit to support gesture recognition. Paper presented at the *Conference Companion of the ACM Conference on Human Factors in Computing Systems* (INTERCHI'93), Amsterdam, The Netherlands. http://portal.acm.org/citation.cfm?id=260123

Lank, E., Cheng, Y.-C. N., and Ruiz, J. (2007a). Endpoint prediction using motion kinematics. Paper presented at the *Proceedings of the ACM Conference on Human Factors in Computing Systems* (CHI'07), San Jose, CA. http://portal.acm.org/citation.cfm?id=1240724

Lank, E., Cheng, Y.-C. N., and Ruiz, J. (2007b). Endpoint prediction using motion kinematics. Paper presented at the *Proceedings of the ACM Conference on Human Factors in Computing Systems* (CHI'07), San Jose, CA. http://portal.acm.org/citation.cfm?id=1240724

Lankford, C. (2000). Effective eye-gaze input into Windows. *Proceedings of the ACM Symposium on Eye Tracking Research and Applications (ETRA'00)*, Palm Beach Gardens, FL, November 6–8, 2000. ACM Press, New York. pp. 23–27.

Lepinski, G. J., Grossman, T., and Fitzmaurice, G. (2010). The design and evaluation of multitouch marking menus. Paper presented at the *Proceedings of the ACM Conference on Human Factors in Computing Systems* (CHI'10), Atlanta, GA. http://dl.acm.org/citation.cfm?id=1753663

Lewis, J., K. Potosnak, and R. Magyar (1997). Keys and keyboards. In M. Helander, T. Landauer, and P. Prabhu (eds.), *Handbook of Human–Computer Interaction*. Amsterdam, the Netherlands: North-Holland. pp. 1285–1316.

Li, Y., Hinckley, K., Guan, Z., and Landay, J. A. (2005). Experimental analysis of mode switching techniques in pen-based user interfaces. Paper presented at the *Proceedings of the ACM Conference on Human Factors in Computing Systems* (CHI'05), Portland, Oregon. http://dl.acm.org/citation.cfm?id=1055036

Li, Y. (2010). Protractor: A fast and accurate gesture recognizer. Paper presented at the *Proceedings of the ACM Conference on Human Factors in Computing Systems* (CHI'10), Atlanta, GA. http://portal.acm.org/citation.cfm?id=1753326.1753654

Lipscomb, J. and M. Pique (1993). Analog input device physical characteristics. *SIGCHI Bulletin* **25**(3): 40–45.

Long, A. C., Landay, J. A., and Rowe, L. A. (1999). Implications for a gesture design tool. Paper presented at the *Proceedings of the ACM Conference on Human Factors in Computing Systems* (CHI'99), Pittsburgh, Pennsylvania. http://portal.acm.org/citation.cfm?id=302985

Long, A. C., Landay, J. A., Rowe, L. A., and Michiels, J. (2000). Visual similarity of pen gestures. Paper presented at the *Proceedings of the ACM Conference on Human Factors in Computing Systems* (CHI'00), The Hague, The Netherlands. http://portal.acm.org/citation.cfm?id=332458

Loomis, J., R. L. Klatzky, R. G. Golledge, and J. W. Philbeck (1999). Human navigation by path integration. In R. G. Golledge (ed.), *Wayfinding: Cognitive Mapping and Other Spatial Processes*. Baltimore, MD: Johns Hopkins. pp. 125–151.

LoPresti, E. F., D. M. Brienza, J. Angelo, L. Gilbertson, and J. Sakai (2000). Neck range of motion and use of computer head controls. *Proceedings of the ACM SIGCAPH Conference on Assistive Technologies (ASSETS'00)*, Arlington, VA, November 13–15, 2000. ACM Press, New York. pp. 121–128.

Lucente, M., G. Zwart, and A. George (1998). Visualization space: A testbed for deviceless multimodal user interface. *Proceedings of the AAAI '98*, Madison, WI.

MacKenzie, I. S. (1992). Fitts' law as a research and design tool in human–computer interaction. *Human–Computer Interaction* **7**(1): 91–139.

MacKenzie, I. S. (1995). Input devices and interaction techniques for advanced computing. In W. Barfield and T. Furness (eds.), *Virtual Environments and Advanced Interface Design*. Oxford, U.K.: Oxford University Press. pp. 437–470.

MacKenzie, I. S. and Y. Guiard (2001). The two-handed desktop interface: Are we there yet? *Proceedings of the ACM CHI 2001 Conference on Human Factors in Computing Systems: Extended Abstracts*, The Hague, the Netherlands. pp. 351–352.

MacKenzie, C. and T. Iberall (1994). *The Grasping Hand*. Amsterdam, the Netherlands: North Holland.

MacKenzie, I. S., T. Kauppinen, and M. Silfverberg (2001). Accuracy measures for evaluating computer pointing devices. *CHI 2001*, Seattle, WA. pp. 9–16.

MacKenzie, I. S. and A. Oniszczak (1998). A comparison of three selection techniques for touchpads. *Proceedings of the ACM CHI'98 Conference on Human Factors in Computing Systems*, Los Angeles, CA. pp. 336–343.

MacKenzie, I. S., A. Sellen, and W. Buxton (1991). A comparison of input devices in elemental pointing and dragging tasks. *Proceedings of the ACM CHI'91 Conference on Human Factors in Computing Systems*, New Orleans, LA. pp. 161–166.

MacKenzie, I. S. and R. W. Soukoreff (2002). A model of two-thumb text entry. *Proceedings of Graphics Interface*, Canadian Information Processing Society, Toronto, Ontario, Canada. pp. 117–124.

MacKenzie, I. S. and K. Tanaka-Ishii (2007). *Text Entry Systems: Mobility, Accessibility, Universality*. San Francisco, CA: Morgan Kaufmann.

MacKenzie, I. S. and C. Ware (1993). Lag as a determinant of human performance in interactive systems. *Proceedings of the ACM INTERCHI'93 Conference on Human Factors in Computing Systems*, Amsterdam, the Netherlands. pp. 488–493.

MacKenzie, I. S. and S. Zhang (1997). The immediate usability of graffiti. *Proceedings of Graphics Interface (GI'97)*, Kelowna, British Columbia, Canada. pp. 129–137. http://portal.acm.org/citation.cfm?id=266792

MacKenzie, I. S. and S. X. Zhang (1999). The design and evaluation of a high-performance soft keyboard. *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'99)*, Pittsburgh, PA, May 15–20, 1999. ACM Press, New York. pp. 25–31.

MacLean, K. E., S. S. Snibbe, and G. Levin (2000). Tagged handles: Merging discrete and continuous control. *Proceedings of the ACM CHI 2000 Conference on Human Factors in Computing Systems*, The Hague, the Netherlands.

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), **2**(1).

Marklin, R. and G. Simoneau (1996). Upper extremity posture of typists using alternative keyboards. *ErgoCon'96*, Palo Alto, CA. pp. 126–132.

Marklin, R., G. Simoneau, and J. Monroe (1997). The effect of split and vertically-inclined computer keyboards on wrist and forearm posture. *Proceedings of the Human Factors and Ergonomics Society (HFES) 41st Annual Meeting*, Santa Monica, CA. pp. 642–646.

Martin, B., and P. Isokoski (2008). EdgeWrite with integrated corner sequence help. Paper presented at the *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'08)*, Florence, Italy. http://portal.acm.org/citation.cfm?id=1357054.1357148

Mathias, E., I. S. MacKenzie, and W. Buxton (1996). One-handed touch typing on a qwerty keyboard. *Human–Computer Interaction* **11**(1): 1–27.

McGuffin, M. and R. Balakrishnan (2002). Acquisition of expanding targets. *CHI Letters* **4**(1): 57–64; *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'02)*, Minneapolis, MN. http://portal.acm.org/citation.cfm?id=503388

McGuffin, M. J. and R. Balakrishnan (2005). Fitts' law and expanding targets: Experimental studies and designs for user interfaces. *ACM Transactions on Computer–Human Interaction* **12**(4): 388–422.

McLoone, H., K. Hinckley, and E. Cutrell (2003). Interaction on the Microsoft Office Keyboard. In *Proceedings of IFIP INTERACT03: Human-Computer Interaction*, Zurich, Switzerland, Sept 01–05. IOS Press: Amsterdam, The Netherlands, pp. 49–56.

McNeill, D. (1992). *Hand and Mind: What Gestures Reveal about Thought*. Chicago, IL: University of Chicago Press.

Mine, M., F. Brooks, and C. Sequin (1997). Moving objects in space: Exploiting proprioception in virtual-environment interaction. *Computer Graphics* **31**: 19–26; *Proceedings of the SIGGRAPH'97*, Los Angeles, CA.

Moffatt, K. and J. McGrenere (2010). Steadied-bubbles: Combining techniques to address pen-based pointing errors for younger and older adults. Paper presented at the *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'10)*, Atlanta, GA. http://portal.acm.org/citation.cfm?id=1753326.1753495

Moran, T., P. Chiu, and W. van Melle (1997). Pen-based interaction techniques for organizing material on an electronic whiteboard. *Proceedings of the 10th Annual ACM UIST'97 Symposium on User Interface Software & Technology*, Banff, Alberta, Canada. pp. 45–54.

Myers, B. A., R. Bhatnagar, J. Nichols, C. H. Peck, D. Kong, R. Miller, and A. C. Long (2002). Interacting at a distance: Measuring the performance of laser pointers and other devices. *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'02)*, Minneapolis, MN, April 2002. ACM Press, New York. pp. 33–40.

Myers, B., K. Lie, and B. Yang (2000). Two-handed input using a PDA and a mouse. *Proceedings of the CHI 2000*, The Hague, the Netherlands. pp. 41–48.

Myers, B. A., R. G. McDaniel, R. C. Miller, A. S. Ferrency, A. Faulring, B. D. Kyle, and P. Doane (1997). The Amulet environment: New models for effective user interface software development. *IEEE Transactions on Software Engineering* **23**(6): 347–365. doi: 10.1109/32.601073.

Myers, B., R. Miller, C. Evankovich, and B. Bostwick (2001). Individual use of hand-held and desktop computers simultaneously (manuscript).

Myers, B., H. Stiel, and R. Gargiulo (1998). Collaboration using multiple PDAs connected to a PC. *Proceedings of the ACM CSCW'98 Conference on Computer Supported Cooperative Work*, Seattle, WA. pp. 285–294.

Mynatt, E. D., T. Igarashi, W. K. Edwards, and A. LaMarca (1999). Flatland: New dimensions in office whiteboards. *ACM SIGCHI Conference on Human Factors in Computing Systems*, Pittsburgh, PA. pp. 346–353.

Ng, A., J. Lepinski, D. Wigdor, S. Sanders, and P. Dietz (2012). Designing for low-latency direct-touch input. Paper presented at the *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology*, Cambridge, MA.

Nguyen, D. H. and E. Mynatt (2001). Towards visibility of a Ubicomp environment, private communication.

Norman, D. (1990). *The Design of Everyday Things*. New York: Doubleday.

Noyes, J. (1983). Chord keyboards. *Applied Ergonomics* **14**: 55–59.

Oakley, I., S. Brewster, and P. Gray (2001). Solving multi-target haptic problems in menu interaction. *Proceedings of the ACM CHI 2001 Conference on Human Factors in Computing Systems: Extended Abstracts*, Seattle, WA. pp. 357–358.

Olsen, D. R. and T. Nielsen (2001). Laser pointer interaction. *Proceedings of the ACM CHI 2001 Conference on Human Factors in Computing Systems*, Seattle, WA. pp. 17–22.

Olson, J. R. and G. M. Olson (1990). The growth of cognitive modeling in human–computer interaction since GOMS. *Human–Computer Interaction* **5**(2–3): 221–266.

Olwal, A., S. Feiner, and S. Heyman (2008). Rubbing and tapping for precise and rapid selection on touch-screen displays. Paper presented at the *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'08)*, Florence, Italy. http://portal.acm.org/citation.cfm?id=1357105

Oviatt, S. (1997). Multimodal interactive maps: Designing for human performance. *Human–Computer Interaction* **12**: 93–129.

Pearson, G. and M. Weiser (1988). Exploratory evaluation of a planar foot-operated cursor-positioning device. *Proceedings of the ACM CHI'88 Conference on Human Factors in Computing Systems*, Washington, DC. pp. 13–18.

Pekelney, R. and R. Chu (1995). Design criteria of an ergonomic mouse computer input device. *Proceedings of the Human Factors and Ergonomics Society (HFES) 39th Annual Meeting*, Santa Monica, CA. pp. 369–373.

Perlin, K. and D. Fox (1993). Pad: An alternative approach to the computer interface. *SIGGRAPH'93*, Anaheim, CA.

Platt, J. (2000). Optimal filtering for patterned displays. *IEEE Signal Processing Letters* **7**(7): 179–183.

Plimmer, B., P. Reid, R. Blagojevic, A. Crossan, and S. Brewster (2011). Signing on the tactile line: A multimodal system for teaching handwriting to blind children. *ACM Transactions on Computer–Human Interaction* **18**(3): 17:11–17:29. doi: 10.1145/1993060.1993067.

Plumlee, M. and C. Ware (2002). Modeling performance for zooming vs multi-window interfaces based on visual working memory. *AVI'02: Advanced Visual Interfaces*, Trento Italy.

Potter, R. L., L. J. Weldon, and B. Shneiderman (1988). Improving the accuracy of touch screens: An experimental evaluation of three strategies. Paper presented at the *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'88)*, Washington, DC, May 15–19. ACM Press, New York. http://portal.acm.org/citation.cfm?id=57167.57171

Poupyrev, I. and S. Maruyama (2003). Tactile interfaces for small touch screens. Paper presented at the *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST'03)*, Vancouver, British Columbia, Canada. http://dl.acm.org/citation.cfm?id=964721

Poupyrev, I., S. Maruyama, and J. Rekimoto (2002). Ambient touch: Designing tactile interfaces for handheld devices. *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST 2002)*, Paris, France. pp. 51–60. http://dl.acm.org/citation.cfm?id=571993

Putz-Anderson, V. (1988). *Cumulative Trauma Disorders: A Manual for Musculoskeletal Diseases of the Upper Limbs*. Bristol, PA: Taylor & Francis Group.

Ramos, G., A. Cockburn, R. Balakrishnan, and M. Beaudouin-Lafon (2007). Pointing lenses: Facilitating stylus input through visual-and motor-space magnification. Paper presented at the *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'07)*, San Jose, CA. http://portal.acm.org/citation.cfm?id=1240624.1240741

Rekimoto, J. (1996). Tilting operations for small screen interfaces. *Proceedings of the UIST'96*, Seattle, WA. pp. 167–168.

Rekimoto, J. (1997). Pick-and-drop: A direct manipulation technique for multiple computer environments. *Proceedings of the ACM UIST'97 Symposium on User Interface Software & Technology*, Banff, Alberta, Canada. pp. 31–39.

Rekimoto, J. (1998). A multiple device approach for supporting whiteboard-based interactions. *Proceedings of the CHI'98*, Los Angeles, CA. pp. 344–351.

Rime, B. and L. Schiaratura (1991). Gesture and speech. In *Fundamentals of Nonverbal Behavior*, R.S. Feldman and B. Rime' (eds.). New York: Press Syndicate of the University of Cambridge. pp. 239–281.

Robertson, G. G., S. K. Card, and J. D. Mackinlay (1989). The cognitive coprocessor architecture for interactive user interfaces. *Proceedings of the UIST'89 Symposium on User Interface Software and Technology*, Williamsburg, VA. pp. 10–18.

Robertson, G., M. Czerwinski, K. Larson, D. Robbins, D. Thiel, and M. van Dantzich (1998). Data mountain: Using spatial memory for document management. *Proceedings of the UIST'98*, San Francisco, CA.

Robertson, G., M. van Dantzich, D. Robbins, M. Czerwinski, K. Hinckley, K. Risden, V. Gorokhovsky, and D. Thiel (1999). The task gallery: A 3D window manager. *ACM CHI 2000*, pp. 494–501. The Hague, the Netherlands.

Robertson, P. K. (1988). Perceptual color spaces. Visualizing color gamuts: A user interface for the effective use of perceptual color spaces in data display. *IEEE Computer Graphics and Applications* **8**(5): 50–64.

Rosen, P., Z. Pizlo, C. Hoffmann, and V. Popescu (2004). Perception of 3D spatial relations for 3D displays. *Stereoscopic Displays* **XI**: 9–16.

Roudaut, A., S. Huot, and E. Lecolinet (2008). TapTap and MagStick: Improving one-handed target acquisition on small touch-screens. Paper presented at the *Proceedings of the ACM Working Conference on Advanced Visual Interfaces (AVI'08)*, Napoli, Italy. http://dl.acm.org/citation.cfm?id=1385594

Rutledge, J. and T. Selker (1990). Force-to-motion functions for pointing. *Proceedings of INTERACT'90: The IFIP Conference on Human–Computer Interaction*, Cambridge, U.K. pp. 701–706.

Saffo, P. (1997). *Communications of the ACM*, February 1997, **40**(2):93–97.

Saponas, T. S., D. S. Tan, D. Morris, R. Balakrishnan, J. Turner, and J. A. Landay (2009). Enabling always-available input with muscle-computer interfaces. Paper presented at the *Proceedings of the 22nd Annual ACM Symposium on User Interface Software and Technology*, Victoria, British Columbia, Canada.

Sawhney, N. and C. 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, B. N., N. I. Adams, and R. Want (1994). Context-aware computing applications. *Proceedings of the IEEE Workshop on Mobile Computing Systems and Applications*, IEEE Computer Society, Santa Cruz, CA. pp. 85–90.

Schmandt, C. M. (1983). Spatial input/display correspondence in a stereoscopic computer graphic work station. *Computer Graphics* **17**(3): 253–262; *Proceedings of the ACM SIGGRAPH'83*, Detroit, MI.

Schmandt, C. M., N. Marmasse, S. Marti, N. Sawhney, and S. Wheeler (2000). Everywhere messaging. *IBM Systems Journal* **39**(3–4): 660–677.

Schmidt, A., K. Aidoo, A. Takaluoma, U. Tuomela, K. Van Laerhove, and W. Van de Velde (1999). Advanced interaction in context. *Handheld and Ubiquitous Computing (HUC'99)*, Karlsruhe, Germany. Springer-Verlag, Berlin, Germany. pp. 89–101.

Schwarz, J., S. E. Hudson, J. Mankoff, and A. D. Wilson (2010). A framework for robust and flexible handling of inputs with uncertainty. Paper presented at the *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST'10)*, New York. http://portal.acm.org/citation.cfm?id=1866039

Schwarz, J., J. Mankoff, and S. E. Hudson (2011). Monte Carlo methods for managing interactive state, action and feedback under uncertainty. Paper presented at the *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST'11)*, Santa Barbara, CA. http://dl.acm.org/citation.cfm?id=2047227

Schwesig, C., I. Poupyrev, and E. Mori (2004). Gummi: A bendable computer. Paper presented at the *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'04)*, Vienna, Austria. http://portal.acm.org/citation.cfm?id=985726

Sears, A. (1993). Investigating touchscreen typing: The effect of keyboard size on typing speed. *Behavior & Information Technology* **12**(1): 17–22.

Sears, A., C. Plaisant, and B. Shneiderman (1992). A new era for high precision touchscreens. *Advances in Human–Computer Interaction* **3**: 1–33; In R. Hartson and D. Hix (eds.), *Advances in Human–Computer Interaction*. Norwood, NJ: Ablex Publishers.

Sears, A. and B. 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.

Serra, L., N. Hern, C. Beng Choon, and T. Poston (1997). Interactive vessel tracing in volume data. *ACM/SIGGRAPH Symposium on Interactive 3D Graphics*, Providence, RI. ACM Press, New York. pp. 131–137.

Sheridan, T. B. (1992). *Telerobotics, Automation, and Human Supervisory Control*. Cambridge, MA: MIT Press.

Sigoma, K. B. (1993). A survey of perceptual feedback issues in dexterous telemanipulation: Part I. Finger force feedback. *Proceedings of the IEEE Virtual Reality Annual International Symposium*, San Jose, CA. pp. 263–270.

Silverstein, D. (1977). Human factors for color display systems. In *Color and the Computer: Concepts, Methods and Research*. New York: Academic Press. pp. 27–61.

Simpson, C. A. and K. Marchionda-Frost (1984). Synthesized speech rate and pitch effects on intelligibility of warning messages for pilots. *Human Factors* **26**: 509–517.

Smailagic, A. and D. Siewiorek (February 1996). Modalities of interaction with CMU wearable computers. *IEEE Personal Communications* **3**(1): 14–25.

Smith, R. B. and A. Taivalsaari (1999). Generalized and stationary scrolling. *CHI Letters* **1**(1): 1–9; *Proceedings of UIST'99*, Asheville, NC.

Snibbe, S. and K. MacLean (2001). Haptic techniques for media control. *CHI Letters* **3**(2): 199–208; *Proceedings of the UIST 2001*, Orlando, FL.

Song, H., F. Guimbretière, S. Izadi, X. Cao, and K. Hinckley (2011). Grips and gestures on a multi-touch pen. Paper presented at the *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'11)*, Vancouver, British Columbia, Canada. http://dl.acm.org/citation.cfm?id=1979138

Soukoreff, R. W. and I. S. MacKenzie (2004). Towards a standard for pointing device evaluation, perspectives on 27 years of Fitts' law research in HCI. *International Journal of Human–Computer Studies* **61**(6): 751–789.

Sporka, A. J., S. H. Kurniawan, M. Mahmud, and P. Slavík (2006a). Non-speech input and speech recognition for real-time control of computer games. Paper presented at the *Proceedings of the ACM SIGACCESS Conference on Computers and Accessibility (ASSETS'06)*, Portland, OR. http://dl.acm.org/citation.cfm?id=1169023

Sporka, A. J., S. H. Kurniawan, and P. Slavík (2004). Whistling user interface (U^{3}I). Paper presented at the *Proceedings of the 8th ERCIM Workshop on User Interfaces for All (UI4All'04)*, Vienna, Austria. http://www.springerlink.com/content/dvfetf9xtjj3bv8k/

Sporka, A. J., S. H. Kurniawan, and P. Slavík (2006b). Non-speech operated emulation of keyboard. In J. Clarkson, P. Langdon, and P. Robinson (eds.), *Designing Accessible Technology*. London, U.K.: Springer. pp. 145–154.

Spiker, A., S. Rogers, and J. Cicinelli (1985). Selecting color codes for a computer-generated topographic map based on perception experiments and functional requirements. *Proceedings of the 3rd Symposium on Aviation Psychology*, Ohio State University, Department of Aviation, Columbus, OH. pp. 151–158.

Stifelman, L. (1996). Augmenting real-world objects: A paper-based audio notebook. *CHI'96 Conference Companion*, Vancouver, British Columbia, Canada. pp. 199–200.

Stokes, A., C. Wickens, and K. Kite (1990). *Display Technology: Human Factors Concepts*. Warrendale, PA: SAE.

Stone, M. C., W. B. Cowan, and J. C. Beatty (1988). Color gamut mapping and the printing of digital color images. *ACM Transactions on Graphics* **7**(4): 249–292.

Stratton, G. (1897). Vision without inversion of the retinal image. *Psychological Review* **4**: 360–361.

Streitz, N. A., J. Geißler, T. Holmer, S. Konomi, C. Müller-Tomfelde, W. Reischl, P. Rexroth, P. Seitz, and R. Steinmetz (1999). i-LAND: An interactive landscape for creativity and innovation. *ACM CHI'99 Conference on Human Factors in Computing Systems*, Pittsburgh, PA. pp. 120–127.

Sugimoto, M. and K. Hiroki (2006). HybridTouch: An intuitive manipulation technique for PDAs using their front and rear surfaces. Paper presented at the *Proceedings of the ACM Conference on Human–Computer Interaction with Mobile Devices and Services (MobileHCI'06)*, Helsinki, Finland. http://portal.acm.org/citation.cfm?id=1152243

Sugiura, A. and Y. Koseki (1998). A user interface using fingerprint recognition—Holding commands and data objects on fingers. *UIST'98 Symposium on User Interface Software & Technology*, San Francisco, CA. pp. 71–79.

Sutherland, I. E. (1968). A head-mounted three dimensional display. *Proceedings of the Fall Joint Computer Conference*, San Francisco, CA. pp. 757–764.

Swaminathan, K. and S. Sato (January–February 1997). Interaction design for large displays. *Interactions,* **4**(1):15–24.

Swigart, S. (2005). Easily write custom gesture recognizers for your tablet PC applications. Tablet PC Technical Articles. Retrieved March 19, 2007, from http://msdn.microsoft.com/en-us/library/aa480673.aspx

Tan, D. S., J. K. Stefanucci, D. R. Proffitt, and R. Pausch (2001). The Infocockpit: Providing location and place to aid human memory. *Workshop on Perceptive User Interfaces*, Orlando, FL.

Tan, D. S., J. K. Stefanucci, D. R. Proffitt, and R. Pausch (2002). Kinesthesis aids human memory. *CHI 2002 Extended Abstracts*, Minneapolis, MN.

Tandler, P., T. Prante, C. Müller-Tomfelde, N. A. Streitz, and R. Steinmetz (2001). Connectables: Dynamic coupling of displays for the flexible creation of shared workspaces. *UIST 2001*, Orlando, FL. pp. 11–20.

Tani, M., K. Yamaashi, K. Tanikoshi, M. Futakawa, and S. Tanifuji (1992). Object-oriented video: Interaction with real-world objects through live video. *Proceedings of ACM CHI'92 Conference on Human Factors in Computing Systems*, Monterey, CA. pp. 593–598, 711–712.

Trevor, J., D. M. Hilbert, B. N. Schilit, and T. K. Koh (2001). From desktop to phonetop: A UI for web interaction on very small devices. *Proceedings of the UIST'01 Symposium on User Interface Software and Technology*, Orlando, FL. pp. 121–130.

Trewin, S., S. Keates, and K. Moffatt (2006). Developing steady clicks: A method of cursor assistance for people with motor impairments. Paper presented at the *Proceedings of the ACM SIGACCESS Conference on Computers and Accessibility (ASSETS'06)*, Portland, OR. http://portal.acm.org/citation.cfm?id=1168987.1168993

Triesman, A. (1985). Preattentive processing in vision. *Computer Vision, Graphics, and Image Processing* **31**: 156–177.

Tufte, E. R. (1983). *The Visual Display of Quantitative Information*. Cheshire, CT: Graphics Press.

Tufte, E. R. (1990). *Envisioning Information*. Cheshire, CT: Graphics Press.

Tufte, E. R. (1997). *Visual Explanations: Images and Quantities, Evidence and Narrative*. Cheshire, CT: Graphics Press.

Ullmer, B. and H. Ishii (1997). The metaDESK: Models and prototypes for tangible user interfaces. Paper presented at the *Proceedings of the 10th Annual ACM Symposium on User Interface Software and Technology*, Banff, Alberta, Canada.

Vogel, D. and P. Baudisch (2007). Shift: A technique for operating pen-based interfaces using touch. Paper presented at the *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'07)*, San Jose, CA. http://portal.acm.org/citation.cfm?id=1240624.1240727

Vogel, D. and G. Casiez (2012). Hand occlusion on a multi-touch tabletop. Paper presented at the *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'12)*, Austin, TX. http://dl.acm.org/citation.cfm?id=2208390

Walker, N. and J. B. Smelcer (1990). A comparison of selection time from walking and bar menus. Paper presented at the *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'90)*, Seattle, WA. http://portal.acm.org/citation.cfm?id=97243.97277

Wang, F., X. Cao, X. Ren, and P. Irani (2009). Detecting and leveraging finger orientation for interaction with direct-touch surfaces. Paper presented at the *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST'09)*, Victoria, British Columbia, Canada. http://dl.acm.org/citation.cfm?id=1622182

Wang, F. and X. Ren (2009). Empirical evaluation for finger input properties in multi-touch interaction. Paper presented at the *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'09)*, Boston, MA. http://dl.acm.org/citation.cfm?id=1518864

Wang, J., S. Zhai, and H. Su (2001). Chinese input with keyboard and eye-tracking: An anatomical study. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Seattle, WA.

Want, R. and G. Borriello (May/June 2000). Survey on information appliances. *IEEE Personal Communications* **20**(3): 24–31.

Want, R., K. P. Fishkin, A. Gujar, and B. L. Harrison (1999). Bridging physical and virtual worlds with electronic tags. *Proceedings of the ACM CHI'99 Conference on Human Factors in Computing Systems*, Pittsburgh, PA. pp. 370–377.

Ware, C. (1988). Color sequences for univariate maps: Theory, experiments, and principles. *IEEE Computer Graphics and Applications* **8**(5): 41–49.

Ware, C. (2000). *Information Visualization: Design for Perception*. San Francisco, CA: Morgan Kaufmann.

Ware, C., K. Arthur, and K. S. Booth (1993). Fish tank virtual reality. *Proceedings of ACM INTERCHI'93 Conference on Human Factors in Computing Systems*, Amsterdam, the Netherlands. pp. 37–41.

Ware, C. and D. R. Jessome (November 1988). Using the bat: A six-dimensional mouse for object placement. *IEEE Computer Graphics and Applications* **8**(6): 65–70.

Ware, C. and J. Rose (1999). Rotating virtual objects with real handles. *ACM Transactions on CHI* **6**(2): 162–180.

Weiss, M. (2010). Bringing everyday applications to interactive surfaces. Paper presented at the *Adjunct Proceedings of the ACM Symposium on User Interface Software and Technology (UIST'10)*, New York. http://dl.acm.org/citation.cfm?id=1866227

Weiss, M., R. Jennings, R. Khoshabeh, J. Borchers, J. Wagner, Y. Jansen, and J. D. Hollan (2009). SLAP widgets: Bridging the gap between virtual and physical controls on tabletops. Paper presented at the *Extended Abstracts of the ACM Conference on Human Factors in Computing Systems (CHI'09)*, Boston, MA. http://dl.acm.org/citation.cfm?id=1520462

Weiser, M. (September 1991). The computer for the 21st century. *Scientific American* **265**(3): 94–104.

Wellner, P. (1993). Interacting with paper on the DigitalDesk. *Communications of the ACM* **36**(7): 87–97.

Wenzel, E. M. (1992). Localization in virtual acoustic displays. *Presence* **1**(1): 80–107.

Westeyn, T., P. Pesti, K.-H. Park, and T. Starner (2005). Biometric identification using song-based blink patterns. *Proceedings of the 11th International Conference on Human–Computer Interaction (HCI Int'l'05)*, Las Vegas, NV, July 22–27, 2005. Lawrence Erlbaum Associates, Mahwah, NJ.

Westheimer, G. (1979). Cooperative neural processes involved in stereoscopic acuity. *Experimental Brain Research* **36**: 585–597.

Wickens, C. (1992). *Engineering Psychology and Human Performance*. New York: HarperCollins.

Wigdor, D., C. Forlines, P. Baudisch, J. Barnwell, and C. Shen (2007). LucidTouch: A see-through mobile device. Paper presented at the *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST'07)*, Newport, RI. http://portal.acm.org/citation.cfm?id=1294211.1294259

Wigdor, D., D. Leigh, C. Forlines, S. Shipman, J. Barnwell, R. Balakrishnan, and C. Shen (2006). Under the table interaction. Paper presented at the *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST'06)*, Montreux, Switzerland. http://portal.acm.org/citation.cfm?id=1166253.1166294

Wigdor, D., S. Williams, M. Cronin, R. Levy, K. White, M. Mazeev, and H. Benko (2009). Ripples: Utilizing per-contact visualizations to improve user interaction with touch displays. Paper presented at the *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST'09)*, Victoria, British Columbia, Canada. http://dl.acm.org/citation.cfm?id=1622180

Wilson, A. and S. Shafer (2003). XWand: UI for intelligent spaces. *ACM CHI 2003 Conference on Human Factors in Computing Systems*, Ft. Lauderdale, FL.

Wilson, F. R. (1998). *The Hand: How Its Use Shapes the Brain, Language, and Human Culture*. New York: Pantheon Books.

Wisneski, C., H. Ishii, A. Dahley, M. Gorbet, S. Brave, B. Ullmer, and P. Yarin (1998). Ambient displays: Turning architectural space into an interface between people and digital information. *Lecture Notes in Computer Science* **1370**: 22–32.

Wobbrock, J. O., H. H. Aung, B. Rothrock, and B. A. Myers (2005). Maximizing the guessability of symbolic input. Paper presented at the *Extended Abstracts of the ACM Conference on Human Factors in Computing Systems (CHI'05)*, Portland, OR, April 2–7. ACM Press, New York. pp. 1869–1872. http://portal.acm.org/citation.cfm?doid=1056808.1057043

Wobbrock, J. O., D. H. Chau, and B. A. Myers (2007). An alternative to push, press, and tap-tap-tap: Gesturing on an isometric joystick for mobile phone text entry. Paper presented at the *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'07)*, San Jose, CA. http://portal.acm.org/citation.cfm?id=1240624.1240728

Wobbrock, J. O., E. Cutrell, S. Harada, and I. S. MacKenzie (2008). An error model for pointing based on Fitts' law. *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'08)*, Florence, Italy, April 5–10, 2008. ACM Press, New York. pp. 1613–1622.

Wobbrock, J. O., J. Fogarty, S.-Y. Liu, S. Kimuro, and S. Harada (2009a). The angle mouse: Target-agnostic dynamic gain adjustment based on angular deviation. Paper presented at the *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'09)*, Boston, MA. http://portal2.acm.org/citation.cfm?id=1518701.1518912

Wobbrock, J. O., J. Forlizzi, S. E. Hudson, and B. A. Myers (2002). WebThumb: Interaction techniques for small-screen browsers. *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST'02)*, Paris, France, October 27–30. ACM Press, New York. pp. 205–208.

Wobbrock, J. O., A. Jansen, and K. Shinohara (2011). Modeling and predicting pointing errors in two dimensions. *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'11)*, Vancouver, British Columbia, Canada, May 7–12. ACM Press, New York. pp. 1653–1656.

Wobbrock, J. O., M. R. Morris, and A. D. Wilson (2009b). User-defined gestures for surface computing. Paper presented at the *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'09)*, Boston, MA. http://portal2.acm.org/citation.cfm?id=1518701.1518866

Wobbrock, J. O., B. A. Myers, and H. H. Aung (2008). The performance of hand postures in front- and back-of-device interaction for mobile computing. *International Journal of Human–Computer Studies* **66**(12): 857–875.

Wobbrock, J. O., B. A. Myers, and J. A. Kembel (2003). EdgeWrite: A stylus-based text entry method designed for high accuracy and stability of motion. Paper presented at the *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST'03)*, Vancouver, British Columbia, Canada, November 2–5. ACM Press, New York. pp. 61–70. http://portal.acm.org/citation.cfm?id=964703

Wobbrock, J.O., K. Shinohara, and A. Jansen (2011). The effects of task dimensionality, endpoint deviation, throughput calculation, and experiment design on pointing measures and models. *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '11)*, May 7–12, 2011, Vancouver, British Columbia, Canada. ACM Press, New York. pp. 1639–1648.

Wobbrock, J. O., A. D. Wilson, and Y. Li (2007). Gestures without libraries, toolkits or training: A $1 recognizer for user interface prototypes. Paper presented at the *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST'07)*, Newport, RI. http://portal.acm.org/citation.cfm?id=1294211.1294238

Worden, A., N. Walker, K. Bharat, and S. E. Hudson (1997). Making computers easier for older adults to use: Area cursors and sticky icons. Paper presented at the *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'97)*, Atlanta, GA. http://portal.acm.org/citation.cfm?id=258724

Wyszecki, G. and W. S. Styles (1982). *Color Science*, 2nd edn., Wiley, New York.

Xin, Y., X. Bi, and X. Ren (2011). Acquiring and pointing: An empirical study of pen-tilt-based interaction. Paper presented at the *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'11)*, Vancouver, British Columbia, Canada. http://dl.acm.org/citation.cfm?id=1978942.1979066

Xin, Y., X. Bi, and X. Ren (2012). Natural use profiles for the pen: An empirical exploration of pressure, tilt, and azimuth. Paper presented at the *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'12)*, Austin, TX. http://dl.acm.org/citation.cfm?id=2208518

Yin, J., and X. Ren (2007). The beam cursor: A pen-based technique for enhancing target acquisition. In N. Bryan-Kinns, A. Blanford, P. Curzon, and L. Nigay (eds.), *People and Computers XX: Proceedings of HCI 2006*. London, U.K.: Springer. pp. 119–134.

Zeleznik, R., K. Herndon, and J. Hughes (1996). SKETCH: An interface for sketching 3D scenes. *Proceedings of SIGGRAPH'96*, New Orleans, LA. pp. 163–170.

Zhai, S. (1998). User performance in relation to 3D input device design. *Computer Graphics* **32**(8): 50–54.

Zhai, S., S. Conversy, M. Beaudouin-Lafon, and Y. Guiard (2003). Human on-line response to target expansion. *CHI 2003 Conference on Human Factors in Computing Systems*, Ft. Lauderdale, FL. pp. 177–184. http://portal.acm.org/citation.cfm?id=642644

Zhai, S., M. Hunter, and B. A. Smith (2000). The metropolis keyboard—An exploration of quantitative techniques for virtual keyboard design. *CHI Letters* **2**(2): 119–128.

Zhai, S. and P. Milgram (1993). Human performance evaluation of isometric and elastic rate controllers in a 6DoF tracking task. *Proceedings of the SPIE Telemanipulator Technology*, **2057**:130–141.

Zhai, S., P. Milgram, and W. Buxton (1996). The influence of muscle groups on performance of multiple degree-of-freedom input. *Proceedings of CHI'96: ACM Conference on Human Factors in Computing Systems*, Vancouver, British Columbia, Canada. ACM Press, New York. pp. 308–315.

Zhai, S., C. Morimoto, and S. Ihde (1999). Manual and Gaze Input Cascaded (MAGIC) pointing. *Proceedings of the ACM CHI'99 Conference on Human Factors in Computing Systems*, Pittsburgh, PA. pp. 246–253.

Zhai, S., B. A. Smith, and T. Selker (1997). Improving browsing performance: A study of four input devices for scrolling and pointing tasks. *Proceedings of the INTERACT'97: The Sixth IFIP Conference on Human–Computer Interaction*, Sydney, New South Wales, Australia. pp. 286–292.

Zhao, S., M. Agrawala, and K. Hinckley (2006). Zone and polygon menus: Using relative position to increase the breadth of multi-stroke marking menus. Paper presented at the *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'06)*, Montreal, Quebec, Canada. http://dl.acm.org/citation.cfm?id=1124933

Zhao, S. and R. Balakrishnan (2004). Simple vs. compound mark hierarchical marking menus. Paper presented at the *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST'04)*, Santa Fe, NM. http://dl.acm.org/citation.cfm?id=1029639