Assistive Technologies and Computer Access for Motor Disabilities

Georgios Kouroupetroglou
University of Athens, Greece

A volume in the Advances in Medical Technologies and Clinical Practice (AMTCP) Book Series

Medical Information Science
REFERENCE
An Imprint of IGI Global

Managing Director:	Lindsay Johnston
Editorial Director:	Joel Gamon
Production Manager:	Jennifer Yoder
Publishing Systems Analyst:	Adrienne Freeland
Development Editor:	Austin DeMarco
Assistant Acquisitions Editor:	Kayla Wolfe
Typesetter:	Christina Barkanic
Cover Design:	Jason Mull

Published in the United States of America by
Medical Information Science Reference (an imprint of IGI Global)
701 E. Chocolate Avenue
Hershey PA 17033
Tel: 717-533-8845
Fax: 717-533-8661
E-mail: cust@igi-global.com
Web site: http://www.igi-global.com

Library of Congress Cataloging-in-Publication Data

Assistive technologies and computer access for motor disabilities / Georgios Kouroupetroglou, editor. p. ; cm.
Includes bibliographical references and index.
Summary: "This book investigates solutions to the problem of technology access by highlighting the principles, methods and advanced technological solutions for all those with motor impairments"--Provided by publisher.
ISBN 978-1-4666-4438-0 (hardcover) -- ISBN 978-1-4666-4439-7 (ebook) -- ISBN 978-1-4666-4440-3 (print & perpetual access) I. Kouroupetroglou, Georgios, editor of compilation. [DNLM: 1. Self-Help Devices. 2. Disabled Persons--rehabilitation. 3. Motor Activity. 4. User-Computer Interface. WB 320]
HV1569.5
617'.033--dc23
2013019756

This book is published in the IGI Global book series Advances in Medical Technologies and Clinical Practice (AMTCP) (ISSN: 2327-9354; eISSN: 2327-9370)

British Cataloguing in Publication Data
A Cataloguing in Publication record for this book is available from the British Library.

Chapter 8
Improving Pointing in Graphical User Interfaces for People with Motor Impairments Through Ability-Based Design

Jacob O. Wobbrock
University of Washington, USA

ABSTRACT

Pointing to targets in graphical user interfaces remains a frequent and fundamental necessity in modern computing systems. Yet for millions of people with motor impairments, children, and older users, pointing—whether with a mouse cursor, a stylus, or a finger on a touch screen—remains a major access barrier because of the fine-motor skills required. In a series of projects inspired by and contributing to ability-based design, we have reconsidered the nature and assumptions behind pointing, resulting in changes to how mouse cursors work, the types of targets used, the way interfaces are designed and laid out, and even how input devices are used. The results from these explorations show that people with motor difficulties can acquire targets in graphical user interfaces when interfaces are designed to better match the abilities of their users. Ability-based design, as both a design philosophy and a design approach, provides a route to realizing a future in which people can utilize whatever abilities they have to express themselves not only to machines, but to the world.

INTRODUCTION

For many people today, the word “computer” is synonymous with a machine that displays a graphical user interface: depictions on a screen that convey information to a user and enable a user to convey information back to a machine. Although computers existed for decades prior to graphical user interfaces, and although many computers exist today without any visual display—for example, computers embedded in automotive systems, satellites, or home appliances—people’s notions of computers are still dominated by graphical user interfaces. It seems that wherever users go, and on whatever platform users operate, pixels arranged mostly in rectangular shapes are there to greet them. Most users today even carry at least one graphical user interface in their pocket, the immensely popular smartphone that has become more “computer” than ever it was a “telephone.”

Along with the popularity of graphical user interfaces has come the related need for users

DOI: 10.4018/978-1-4666-4438-0.ch008

to point to graphically portrayed objects. An early famous example is Ivan Sutherland's 1963 Sketchpad, whose direct-pointing approach using a light pen has enjoyed a modern rebirth in the form of stylus- and finger-based direct-touch devices like smartphones (Sutherland, 1963). Douglas Engelbart's 1968 NLS demo first unveiled the relative-pointing scheme of the mouse (Engelbart, 1963), which was adopted by Xerox PARC for use with their bitmapped graphical displays of the late 1970s (Johnson, Roberts, Verplank, Smith, Irby, Beard, & Mackey, 1989), and later successfully commercialized by the Apple Macintosh in 1984 (Williams, 1984) and Microsoft Windows 1.0 in 1985 (Markoff, 1983).

Despite the revolutionary hardware and software advances that have driven computer evolution, a truth has remained: to operate a graphical user interface, a user must be able to successfully point-and-click on graphical targets rapidly, reliably, and repeatedly. Along with text entry, pointing comprises the essential substrate of interactive computer use. There is precious little way of escaping it, as even command-line aficionados must admit. Studies show that depending on the tasks being performed, 31-65% of computer users' time is spent using the mouse, and one-third to one-half of that time is spent dragging, a complex human motor operation (Johnson, Dropkin, Hewes, & Rempel, 1993). More recent studies show that mouse usage outweighs keyboard usage by three to five times (Chang, Amick, Menendez, Katz, Johnson, Robertson, & Dennerlein, 2007; Mikkelsen, Vilstrup, Lassen, Kryger, Thomsen, & Andersen, 2007). And yet, despite the inescapable requirement of pointing, it still represents a major obstacle to successful computer use for millions of people with motor impairments and motor-related difficulties (Riviere & Thakor, 1996). Any of the three "r" words above can be significant challenges:

- **Rapidly:** Some people with motor impairments can point only extremely slowly, which means operating a computer can be an excruciating and arduous process.
- **Reliably:** Some people with motor impairments have a great deal of variation in their movements, which means the outcomes of their aimed pointing attempts are neither consistent nor predictable.
- **Repeatedly:** Some people with motor impairments fatigue quickly, which means repeated use degrades their performance before they can accomplish their tasks.

At its most basic, pointing on a graphical display results in indicating a one-pixel island in an ocean of surrounding pixels. In practice, that single pixel usually belongs to a group of pixels comprising a user interface object of some kind—a button, hyperlink, menu item, scrollbar, or similar. Placing an indicator—whether a mouse cursor, stylus, or finger—inside a small screen area and doing so rapidly, reliably, and repeatedly requires an abundance of motor skills (Sutter & Ziefle, 2005). These skills include the ability to grip a device or position the hand, the ability to exert controlled force, and the ability to make fine submovement corrections during the final phases of target acquisition (Meyer, Abrams, Kornblum, Wright, & Smith, 1988; Meyer, Smith, Kornblum, Abrams, & Wright, 1990). Pointing requires both gross and fine motor control. About half of all users point with a mouse by lifting and suspending the arm; the other half point primarily using their wrists and fingers (Balakrishnan & MacKenzie, 1997; Johnson et al., 1993). When pointing, a user's psychomotor system must be able to receive ongoing visual feedback and couple that visual feedback to proprioceptive and kinesthetic feedback to rapidly issue and execute accurate movements (Crossman & Goodeve, 1963, 1983). It is no surprise, given the complex human perceptual-motor systems involved, that pointing can be difficult for a variety of reasons.

Problems that compromise pointing may be neurological, perceptual, muscular, skeletal, or combinations thereof (Sears, Young, & Feng, 2008). Some motor impairments result in the inability to make fast movements, but permit slow accurate movements, such as for some forms of muscular dystrophy or spinal cord injury. Other motor impairments cause spasms, resulting in fast movements without the ability to make fine submovement corrections, such as for some people with cerebral palsy. Other motor impairments may take the form of an unrelenting baseline essential tremor, causing the hands to shake, or repetitive stress injury, causing pain or stiffness in the wrists. Inflammation and joint stiffness due to arthritis may be the most prevalent source of motor impairment, affecting, in the U.S. alone, 49.9 million people (Cheng, Hootman, Murphy, Langmaid, & Helmick, 2010). Even being young or old can cause the perceptual-motor system to perform worse than in other phases of life. Additional manifestations of motor impairment can include rapid fatigue, poor coordination, low strength, slow reaction time, difficulty gripping, difficulty holding, lack of sensation, and difficulty controlling movement direction or distance. As a country's population ages, the prevalence of these symptoms only increases. At the same time, the importance of computer access in people's lives also increases, as individuals with motor impairments look to computers as important sources of information, communication, employment, and entertainment (Muller, Wharton, McIver, & Laux, 1997). Given the ubiquity of pointing and the barrier to access it poses for many people with motor impairments, the need to make pointing more accessible is paramount. Figure 1 shows some of the motor-impaired hand postures utilized for pointing that we have observed in our studies.

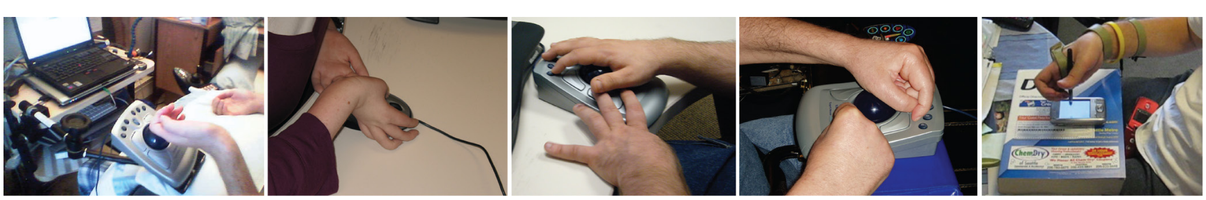
The approach taken in the work reported in this chapter[1] is to improve pointing for people with motor impairments through ability-based design (Wobbrock, Kane, Gajos, Harada, & Froehlich, 2011a), which seeks to create accessible technologies by focusing on what users can do (Chickowski, 2004), as opposed to focusing on "dis-abilities," or what users cannot do, and by allowing users, not systems, to dictate the abilities necessary for system operation. A primary tenet of ability-based design is to "leave users as they are," meaning that designers look to change systems to accommodate users, not the other way around. Ability-based design, explained in more detail in the next section, is both a design philosophy and a design approach, refined and informed in part by the projects described in this chapter.

In seeking to leverage users' abilities, we have created multiple software solutions that improve pointing for people with motor impairments by changing the following fundamental aspects of interactive computing systems:

- The mouse cursor.
- The target types.
- The interface design.
- The input device.

The objectives of this chapter are: (1) to describe ability-based design, (2) to give an overview of pointing facilitation for users with motor impairments, and (3) to describe our own projects

Figure 1. Hand postures utilized with commodity input devices by some people with motor impairments



in motor-impaired pointing facilitation, and by so doing, illustrate ability-based design in action. Researchers may learn new ways of thinking about accessibility and accessible design. Practitioners may gain exposure to a new design philosophy and design approach, becoming aware of the results it produces and motivating its adoption in practice. Students may gain familiarity with pointing facilitation generally, and our ability-based design approach to pointing facilitation specifically. Before describing our projects, some background on ability-based design and on research in accessible pointing facilitation is in order. We take each of these topics in turn.

OVERVIEW OF ABILITY-BASED DESIGN

Ability-based design is both a design philosophy and a design approach (Wobbrock et al., 2011a). Philosophically, ability-based design resists the "assistive," "rehabilitative," "functional-restorative," and so-called "universal" and "for-all" approaches to accessible design, favoring instead the notion that all people, at any point in time, in any given environment, have certain exercisable abilities that should be expressible through technology, and that ideally, this technology should be aware of and responsive to the specific abilities of its users. Although this ideal may be unachievable in general, striving for it promotes a different set of priorities than striving to "assist" a deficient user, striving to "replace" or "restore" lost function, or striving to produce designs that simultaneously accommodate vast swaths of hypothetical users. Certainly, the "universal" approaches have many successes, but computing technologies are quite unlike the built environments that instigated their inception (Bowe, 1987). Unlike buildings, stairways, sidewalks, and canes, computers can sense users' behavior, measure and model it, compare it to histories or others' behavior, and adapt to it or suggest a useful set of adaptations (Hwang, Keates, Langdon, Clarkson, & Robinson, 2001). The capabilities of computing open the door to something greater than assistive technology or design-for-all. They open the door to "the universal application of *design-for-one*" (Harper, 2007; Ringbauer, Peissner, & Gemou, 2007; Wobbrock *et al.*, 2011a).

As a design approach, ability-based design espouses principles that attempt to shift the focus of a system designer or builder. Just as user-centered design shifted the focus from *systems* to *users* (Gould & Lewis, 1985), ability-based design attempts to shift the focus from *disability* to *ability*, the positive affirmation that all living humans have abilities and discovering and exploiting these abilities instead of attempting to "restore" lost abilities, or "assist" people in satisfying the demands of oblivious systems, must be our priority. Ability-based design also affirms that users *want* to feel and be perceived as capable, not disabled, and make sovereign choices based on their desires (Bowe, 1988; McCuaig & Frank, 1991). Therefore, the more aware designers, engineers, and systems can be of users' abilities, the greater the chances for a successful match between a system's operational demands and what users can do.

Ability-based design grew out of *ability-based user interfaces* (Gajos, Wobbrock, & Weld, 2007, 2008, 2010), which are sophisticated examples of how adaptivity can be used to improve interfaces for people with motor impairments. Whereas ability-based user interfaces are *artifacts*, ability-based design expands its purview beyond artifacts to the design philosophy and design approach, including the designer's stance, users' contexts, and certain value assertions. Although deep adaptivity like the kind employed by ability-based user interfaces is extremely useful for realizing ability-based design, adaptation is only part of a larger vision. Ability-based design can be upheld in the design process by retaining a clear focus on a user's abilities, on how those abilities are expressed, and on how those abilities change with context and time. Using an ability-based design approach, therapists may make numerous performance measurements with computational support to find not only the most appropriate intervention but the best way to change the system to suit the user. An ability-based stance does not allow available hardware or software to dictate the prescribed intervention, but allows what people *can do* to dictate what interventions must be procured or created.

Ability-based design also considers *changes* to ability, which can be spurred by fatigue, medication, disease progression, boredom, environment, or context. In a future world where sensors are spread throughout environments and data from human activity is widely available, it is conceivable that a person's abilities could be tracked over time, insofar as that person permits. Human performance measurement is an important feature of ability-based design, one that is highly relevant to accessible pointing because of pointing's observable, quantifiable nature.

Ability-based design is open to the whole range of human ability, including the high end of the ability spectrum. An ability-based design process should just as readily result in solutions for high-functioning individuals to utilize their extraordinary mental, perceptual, or physical abilities as it does for low-functioning individuals. To date, the focus of ability-based design has been on achieving accessible systems for people with disabilities, but there is nothing inherent in ability-based design that prevents its use for high-functioning individuals as well.

Situational Impairments

Also relevant to ability-based design are the ways in which abilities can be compromised by situational and environmental factors, factors that induce *situational impairments* (Sears, Lin, Jacko, & Xiao, 2003; Sears et al., 2008). Situational impairments may affect otherwise able-bodied users who are out in the world, moving through space, subject to ambient inputs like light, forces, and noise, and obligated to flit their attentional resources rapidly among multiple stimuli. Situational impairments are partly why automobile drivers writing text messages are prone to crashing, or why people walking while texting may overstep the curb into a busy street. With the ubiquity of mobile devices, gone are the days when one can assume a "computer user" means without exception someone sitting in a temperature-controlled indoor environment with a large screen, ample lighting, a stable desk surface, and no distractions. Today, a "computer user" is just as likely, or even *more* likely, to be holding a palm-sized touch screen, walking outdoors, tapping small virtual keys, avoiding obstacles, and thinking about a text message while rushing to a meeting. A host of situational impairments may impact a person's ability to interact: vibration, divided attention, distraction, diverted gaze, device out-of-sight, intervening objects, body motion, vehicle motion, uneven terrain, physical obstacles, awkward postures or grips, occupied hands, cold temperatures, impeding clothing (*e.g.*, gloves), encumbering baggage, rainwater, light levels (*e.g.*, darkness, glare), ambient noise, social interactions (*e.g.*, interruptions), stress, fatigue, haste, or even intoxication (Goel, Findlater, & Wobbrock, 2012; Kane, Wobbrock, & Smith, 2008). There are undoubtedly more. Figure 2 depicts some scenarios where situational impairments occur.

Ability-based design places emphasis on abilities *exercisable in a given context*, and therefore considers the ways in which context impairs performance. Most graphical user interfaces, whether desktop or mobile, are designed with implicit assumptions that users' contexts are static and unimpairing, permitting the accurate selection of targets, the rapid entry of text, and the perceptibility of tiny on-screen indicators. But of course, these assumptions are not valid for anything but the most "serene" desktop conditions, meaning that as situational factors become more detrimental, these assumptions, and the interfaces reliant on them, become more problematic. Imagine a soldier on a battlefield dressed in heavy gear stuck in the mud with deafening explosions

Figure 2. Situational impairments may take many forms and derive from many sources, but all affect people's ability to interact successfully with computers

surrounding him trying to compose a text message while wearing thick combat gloves (Newell, 1995); or a spacewalking astronaut trying to do the same while in zero-gravity. These examples are extreme, but they illustrate an important point: The innate abilities of soldiers and astronauts are impressive, but what matters for accessibility is what these soldiers and astronauts can do *in their environments*; innate abilities are, in a sense, irrelevant. The same is true for less extreme examples, like making song lists on mobile music players more readable while walking (Kane et al., 2008), or making touch screen keyboards more accurate while walking (Goel et al., 2012), both projects we have pursued. A recent intellectual descendant of ability-based design called "personalized dynamic accessibility" has made mobile context a key feature in its argument for personalized adaptive interfaces (Gajos, Hurst, & Findlater, 2012).

Ability to Procure Technology

As with most approaches to accessibility, an important aspect of ability-based design is to reduce barriers to access, and chief among such barriers is *cost* (Fichten, Barile, Asuncion, & Fossey, 2000; LaPlante, Hendershot, & Moss, 1992). The prices of specialized interventions created specifically for people with disabilities can be significantly higher than the prices of mass-market technologies due to smaller production runs and higher certification costs (Bowe, 1995). And specialized devices are more likely to be abandoned due to their higher complexity, configuration, and maintenance needs than simpler, cheaper, more-easily-replaced devices (Dawe, 2005, 2006; Koester, 2003; Phillips & Zhao, 1993; Riemer-Reiss & Wacker, 2000). As a result, studies show that of people indicating a need for adaptations, fewer than 60% of people actually utilize them (Fichten et al., 2000). Ability-based design therefore insists on another set of abilities, namely the ability to procure technology, the ability to configure and maintain it, and the ability to replace it if it is lost or broken. Thus, ability-based design is about the holistic abilities of the user, including their ability to procure technology, not just their sensory or motor abilities.

Instead of high-end specialized devices, everyday input devices like mice, touchpads, and trackballs are widely used by people with motor impairments due to these devices' availability, convenience, low cost, simplicity, lack of necessary maintenance, and reduced or eliminated social stigma (Bowe, 1995; Casali, 1992; Enders, 1995; Gitlin, 1995; Vance, 2009). Everyday input devices are a form of *transparent equipment* in that they do not call attention to a user's disability status (McCuaig & Frank, 1991), which enables the user to retain greater sovereignty over when, how, and to whom their disability status is revealed (Bispo & Branco, 2009). Another benefit of software-based accessibility solutions designed for everyday input devices is that such solutions can be downloaded from the web without having to procure custom hardware. Utilizing everyday input devices, and transparent equipment in general, is therefore the approach favored by ability-based design and adopted in this work.

Principles of Ability-Based Design

The principles of ability based design shown in Table 1 capture the foregoing rationale to aid researchers and practitioners in thinking through possibilities and maintaining an ability-centered perspective. Importantly, these principles are not *rules*, but guidelines; only the first two are required to claim an ability-based design approach has been undertaken.

Comparison to Assistive Technology

It is useful to briefly contrast ability-based design with two predominant approaches to achieving accessible design: assistive technology and universal design. Traditionally, the field of assistive technology, as its name suggests, has "assisted" people to function in ways compatible with the built environment: buildings, vehicles, roads, and so on (Cook & Hussey, 2002; Vanderheiden, 1998). The approach is to augment, supplement, or alter humans to make them amenable to these built environments, or to augment, supplement, or alter the built environment to provide access for people with special needs. A wheelchair may assist a human to move, and a metal wheelchair ramp may sit atop a cement staircase to provide building access for wheelchairs. The wheelchair and the ramp work together to assist people to enter a building in a way that might be termed "separate

Table 1. The principles of ability-based design. The principles remain unchanged, but their descriptions have been updated from their first publication (Wobbrock et al., 2011a). Not all principles must be upheld for a design to be ability-based. Only the first two principles are strictly required.

	Principle	**Description**
Stance	1. Ability	Designers will focus on users' abilities, not *dis*-abilities, striving to leverage all that users *can* do in a given situation, context, or environment.
	2. Accountability	Designers will respond to poor performance by changing systems, not users, leaving users as they are.
Interface	3. Adaptation	Interfaces may be adaptive or adaptable to provide the best possible match to users' abilities.
	4. Transparency	Interfaces may give users awareness of adaptive behaviors and what governs them, and the means to inspect, override, discard, revert, store, retrieve, preview, alter, or test those behaviors.
System	5. Performance	Systems may monitor, measure, model, display, predict, or otherwise utilize users' performance to provide the best possible match between systems and users' abilities.
	6. Context	Systems may sense, measure, model, portray, or otherwise utilize context, situation, or environment to anticipate and accommodate effects on users' abilities.
	7. Commodity	Systems may comprise low-cost, inexpensive, readily available commodity software, hardware, or other materials that users have the ability to procure.

but equal" (Hazard, 2008) from the mainstream method, *i.e.*, the cement stairs.

In computing, assistive technologies are often interventions that "sit between" users and systems to make users able to produce the inputs systems require, or to receive the outputs systems produce, all while systems remain oblivious to the abilities of their users (Bowe, 1987). For example, if a person without use of his hands holds a pencil in his mouth as a cheap aid for typing on a keyboard, the pencil can be said to "assist" this person to type. The contortion this person endures is to "appear a typist" to the computer, whose keyboard requires the striking of small physical keys with certain force. The pencil sits between the user and the system and is neither part of the user nor part of the system; it must be procured separately from the system. The system knows neither that text is being entered using a pencil held in the mouth, nor that the human user cannot use his hands. The user has done all the work to conform himself to the demands of the system.

By contrast, an ability-based design approach asks what this specific user *can* do—speak? hum? turn his head? move her foot? blink? —and try to leverage that. Ideally, the system itself would be an ability-based one, able to accommodate a wide range of human abilities and not require that text be entered via a keyboard at all. Adaptations would be offered by the system and not have to be separately procured by the user. The system would do the work to sense, model, and adapt or suggest adaptations based on the specific observed abilities of the user (Hwang et al., 2001; Keates, Clarkson, & Robinson, 2000; Keates, Langdon, Clarkson, & Robinson, 2002).

Even the pencil scenario could be made more ability-based if the system were to observe and model the key-down and key-up times produced by the user, and accommodate by making adaptations such as setting the debounce time, key-repeat time, Sticky Keys, and so on (Koester, LoPresti, & Simpson, 2007; Trewin & Pain, 1997). Although this design still leaves room for improvement, it begins to take on the characteristics of an ability-based design.

Although technologies can be conceived that seemingly blur the distinction between assistive technology and ability-based design, the distinction lies less in the technologies themselves and more in the approaches of designers working from an ability-based stance or something else. Ability-based design attempts to shift the focus from "people needing assistance" to "empowering people to utilize their abilities to the greatest effect." The technology employed may indeed "assist" its user, but such a view places agency on the wrong party. Rather than a technology assisting its user, the user is exercising his or her abilities through the technology. The technology is not remedying something missing, but is providing expression for abilities already present. In this way, the stance assumed by ability-based design differs sharply from that found in traditional assistive technology. That said, if a technology-based distinction *is* to be found, it usually lies in ability-based designs having more awareness of their users than conventional assistive technologies, which are often completely unaware—but this distinction is not a required one.

Comparison to the "Universal" Approaches

With universal design (Mace, Hardie, & Place, 1991; Steinfeld, 1994; Story, 1998), universal usability (Meiselwitz, Wentz, & Lazar, 2009; Shneiderman, 2000; Vanderheiden, 2000), design-for-all (Keates & Clarkson, 1999; Stary, 1997; Stephanidis, 2001), inclusive design (Keates, Clarkson, Harrison, & Robinson, 2000; Keates & Clarkson, 2003; Newell & Gregor, 2000), and related approaches (Newell, 1995; Pullin & Newell, 2007), the goal is generally to create designs that are usable by as many people as possible without much, if any, alteration or personalization. Universal design, the best known and most celebrated of these, comes from architecture.

Resisting the "separate but equal" outcomes of assistive technology, universal design seeks to make products and environments usable by as many people as possible. Instead of metal ramps overlaying concrete steps, a gradual concrete slope usable by wheelchairs and walkers alike would be preferred under universal design.

In computing, the so-called "universal" approaches often focus on anticipating and avoiding the unwitting creation of barriers to access. For example, the World Wide Web has been a testing ground for universal design, and recommendations such as the Web Content Accessibility Guidelines (WCAG) have relied on universal design to formulate their principles. The four WCAG 2.0 design guidelines for web accessibility[2] are *perceivable*, *operable*, *understandable*, and *robust*, the first three of which are among the seven principles of universal design, albeit under slightly different names (Connell, Jones, Mace, Mueller, Mullick, Ostroff, Sanford, Steinfeld, Story, & Vanderheiden, 1997). Although these guidelines are important, they may require web designers to think of "everyone" generally and try, as best they can, to anticipate design decisions that may exclude. Designing defensively in this way is difficult, as there are possibly infinite things *not* to do.

By contrast, an ability-based design approach thinks less about what hypothetical people possibly *can't* do, and instead focuses on what a target user or users *can* do. It would make options for a wide range of adaptations including color, layout, widget size, and so on; it would include logic to observe user difficulties, such as in clicking, typing, or scrolling; and it would either recommend adaptations or automatically adapt itself based on users' observed behavior. Of course, to create such infrastructure would be unrealistic for every web designer, but web browsers themselves could incorporate such features. Even operating systems could, one day, support these features for all applications running on them. It may be argued that achieving all of this would be tantamount to achieving universal design, but a crucial distinction is that ability-based design would do so by focusing on affirmative abilities, whereas universal design, in trying to anticipate and avoid potential access barriers, would focus on potential *dis*-abilities. Although barrier-avoidance is essential, barrier-avoidance in ability-based design occurs as a byproduct of focusing on abilities. As it was observed twenty-five years ago: "When society makes a commitment to making new technologies accessible to everyone, the focus will no longer be on what people cannot do, but rather on what skills and interests they bring to their work." (Bowe, 1987). Such is the aim of ability-based design.

PRIOR WORK ON ACCESSIBLE POINTING

Pointing-and-clicking has long been recognized as a barrier to computer access, spurring research investigations for as long as graphical user interfaces have been commonplace. There is a large body of work on pointing facilitation techniques in general, although most of this work has not been directed towards people with disabilities, children, or older adults. For a general review of pointing facilitation techniques, the reader is directed to a prior survey (Balakrishnan, 2004). Our review here mainly considers techniques intended for improving computer access for people whose motor abilities are challenged. Some of our own projects are placed in context in this review, but their full treatments are left for the next section.

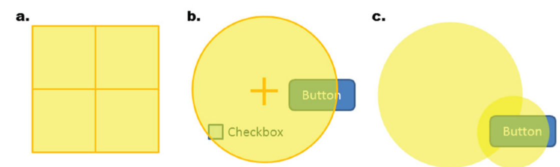
Area Cursors

One pointing facilitation technique that has received considerable attention is the *area cursor* (Kabbash & Buxton, 1995). With the conventional point cursor—the familiar little arrow—a single pixel defines the hot-spot at its tip. Area cursors, by contrast, are often square or circular, and have entire regions that are "hot," that is, capable of selecting targets. The result is that targets are easier to acquire because area cursors are more forgiving than point cursors, demanding less accuracy (Worden, Walker, Bharat, & Hudson, 1997). A downside of conventional area cursors is that they may overlap multiple targets at once, resulting in ambiguity that must be resolved (Mankoff, Hudson, & Abowd, 2000). Conventionally, in the case of overlapping targets, area cursors degenerate to point cursors at their center, often depicted with a crosshairs.

There have been numerous iterations on the area cursor. The fastest general-audience cursor is the *Bubble Cursor*, a circular area cursor that dynamically expands and contracts its radius such that the target nearest to the cursor's center is always contained within the cursor's dynamic bounds (Grossman & Balakrishnan, 2005). Such an approach requires the Bubble Cursor to be fully *target-aware*, meaning that the locations, dimensions, and enabled/disabled states of all visible targets must be knowable at all times by the cursor. In practice, producing such information for target-awareness in commercial systems is a considerable practical and theoretical challenge; it was accomplished on Windows systems only recently by using pixel-based reverse engineering to create the first deployable Bubble Cursor (Dixon, Fogarty, & Wobbrock, 2012). Nonetheless, a limitation of the Bubble Cursor is that it can only point to objects (Guiard, Blanch & Beaudouin-Lafon, 2004), not arbitrary screen coordinates (*i.e.*, pixels), a limitation remedied by *DynaSpot*, a cursor that resizes its radius based on its velocity (Chapuis, Labrune, & Pietriga, 2009). After coming to a stop, DynaSpot progressively shrinks to a point cursor capable of selecting arbitrary screen coordinates. In a formal comparison, there was no detectable speed difference between the Bubble Cursor and DynaSpot. Figure 3 shows some area cursors.

Neither the Bubble Cursor nor DynaSpot were initially evaluated for people with motor impairments. In our own subsequent evaluations, the Bubble Cursor was found to perform poorly in the case of small, densely-packed targets (Findlater, Jansen, Shinohara, Dixon, Kamb, Rakita, & Wobbrock, 2010). Such targets cause the Bubble Cursor to degrade, ultimately all the way to a point cursor, *e.g.*, in the case when moving over a paint canvas in which every pixel is a target. To remedy this limitation, we created a set of four *enhanced area cursors* (Findlater *et al.*, 2010), described in the next section. Briefly, two of these cursors turned overlapping targets into *crossing goals*, and the other two cursors magnified overlapping targets in *motor-space* or in *visual-and-motor-space*. Results generally showed fewer submovement corrections and reduced error rates for small targets using our enhanced area cursors. In the case of the most densely-packed targets, two of the cursors were also significantly faster than the Bubble Cursor.

Figure 3. Area cursors have been explored to make pointing easier: (a) square area cursor, (b) circular area cursor overlapping two targets, (c) the dynamic area cursor known as the Bubble Cursor (Grossman & Balakrishnan, 2005)



Automatic Control-Display Gain Adjustment

The previous subsection referred to "motor-space" and "visual-space." The former refers to the space in which a physical mouse moves across a physical desk (or equivalently, the space in which a physical trackball rotates, or the space in which a physical finger moves across a touchpad). Motor-space may be measured in millimeters. For example, a user may slide his mouse 100 mm and do so in 5 seconds, resulting in an average motor-space velocity of 20 mm/s. Knowing this, however, tells one very little of the cursor movement that results on the computer screen, which occurs in visual-space and is governed by a *transfer function* (Hinckley, 2008). The transfer function takes as input a physical movement in motor-space and produces a corresponding cursor movement in visual-space. These two spaces and the transfer function that relates them have provided ample opportunity for improving pointing for people with disabilities.

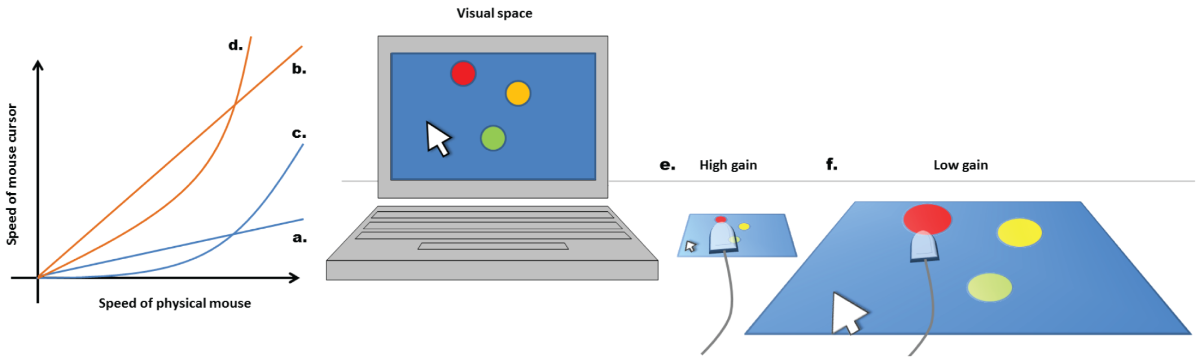
Customarily, mice, trackballs, and touchpads are used as *relative pointing* devices, meaning a physical movement in motor-space produces a relative cursor movement in visual-space. By contrast, a touch screen involves *absolute pointing*, as the motor-space and visual-space are coupled, and the user's physical fingertip (or pen-tip) serves as the "cursor" moving through the unified visual-and-motor-space. Now, indeed there have been explorations into using cursors on touch screens to make relative pointing possible (Albinsson & Zhai, 2003; Benko, Wilson, & Baudisch, 2006; Potter, Weldon, & Shneiderman, 1988; Vogel & Baudisch, 2007). Also, it is not uncommon to have touchpads or touch-tablets behave in absolute pointing mode (Arnaut & Greenstein, 1986; Buxton, Hill, & Rowley, 1985; Buxton & Myers, 1986; Sears & Shneiderman, 1991). In fact, even trackballs can be made to operate in a "nearly absolute" manner (Wobbrock & Myers, 2006). But despite these variations, we focus here on accessibility improvements to relative pointing, for which the majority of techniques have been invented.

In relative pointing, the ratio of movement in visual-space to movement in motor-space is called the *control-display gain*, or C-D gain (MacKenzie & Riddersma, 1994). Its inverse is called the *control-display ratio*, or C-D ratio (Blanch, Guiard, & Beaudouin-Lafon, 2004). Gain and ratio are mathematically equivalent; therefore, we will employ the former. Most modern operating systems allow the user to manually set the control-display gain.[3] Unfortunately, uniform changes to C-D gain do not provide much of a human performance benefit. A uniformly low gain makes targets effectively bigger in motor-space, but also pushes them farther apart by the same proportion. A uniformly high gain has the opposite effect, producing targets that are closer but smaller by the same proportion. In terms of the famous model of aimed pointing performance known as Fitts' law (Fitts, 1954; MacKenzie, 1992), such uniform changes make no difference to overall pointing time. Therefore, any performance improvement from C-D gain adjustment *must* be non-uniform in its application of gain-change. Such automatic approaches to adjusting the C-D gain have been explored with the intention of making targets bigger in motor-space when the cursor is near or inside those targets, and closer together when the cursor is moving between targets.

Conventional *pointer acceleration* has been implemented in commercial systems for decades and has been studied carefully (Casiez, Vogel, Balakrishnan, & Cockburn, 2008). The rationale is simple enough: when the input device moves slowly, the C-D gain is kept low, thus making fine cursor movements easier to produce, and making small targets easier to acquire. But when the input device moves quickly, the C-D gain is increased such that distances otherwise requiring substantial physical movement are made possible with less effort. An important feature of pointer acceleration is that it is *target-agnostic* (Wobbrock *et al.*, 2009)—no information about targets is necessary for its implementation. Pointer acceleration may make pointing easier for people with motor impairments by making small targets easier to acquire, and by reducing the amount of physical motion required to reach distant targets. Figure 4 illustrates transfer functions and the relationship between visual-space and motor-space.

The *Input Device Agent (IDA)*, a project upholding many principles of ability-based design, attempts to discover the optimal C-D gain for people with motor impairments (Koester, LoPres-

Figure 4. Transfer functions, visual-space, and motor-space: (a) low constant gain, (b) high constant gain, (c) low variable gain with pointer acceleration, (d) high variable gain with pointer acceleration, (e) motor-space for high gain, and (f) motor-space for low-gain



ti, & Simpson, 2005). The IDA administers a small battery of pointing trials and, based on performance comparisons, tries to recommend an optimal setting. Unfortunately, in an evaluation, the IDA-recommended settings did not ultimately perform better (or worse) than the Windows default C-D gain, perhaps because the effect of C-D gain on pointing was simply not sufficiently pronounced for the 12 participants tested. It is unclear whether for participants with more severe disabilities the results would be different. In any case, creating systems to observe users' performance and either make or recommend settings is a powerful approach, one encouraged by ability-based design.

More benefit has been found for *sticky icons*, which are targets that reduce C-D gain when the cursor moves inside them, giving a feeling of stickiness (Worden et al., 1997). To prevent unwanted targets from trapping the cursor as it passes over them, researchers have allowed stickiness to take effect only after the cursor has passed below a certain velocity threshold (Rodgers, Mandryk & Inkpen, 2006; Worden et al., 1997). At this point, the sticky icons technique becomes a target-aware technique because the cursor must know at all times whether or not it is within a target. For older adults, results have shown that sticky icons reduce target acquisition times for the smallest targets. Others have found benefits of sticky icons even when evaluating them for people *without* motor difficulties (Blanch et al., 2004; Cockburn & Firth, 2003; Keyson, 1997; Mandryk & Gutwin, 2008).

Related to sticky icons, which are made bigger in motor-space only, are *expanding targets*, which are made bigger in visual-and-motor-space[4] (Cockburn & Firth, 2003; McGuffin & Balakrishnan, 2002; McGuffin & Balakrishnan, 2005; Zhai, Conversy, Beaudouin-Lafon, & Guiard, 2003). Studies show that even when an intended target does not expand until the mouse cursor has traversed 90% of its required distance, performance still improves because 40%-55% of a pointing movement's time is spent in the last 10% of its distance. To date, expanding targets have not been employed for people with motor impairments, but if they can be made practical for real-world use, they may improve interface accessibility. A significant challenge in implementing expanding targets, besides target-awareness, is discerning *which* targets to expand, a concern not addressed in prior work. Target-prediction or endpoint-prediction may be necessary to fully realize expanding targets in everyday systems (Asano, Sharlin, Kitamura, Takashima, & Kishino, 2005; Lane, Peres, Sandor, & Napier, 2005; Lank, Cheng, & Ruiz, 2007; Murata, 1998; Ziebart, Dey, & Bagnell, 2012).

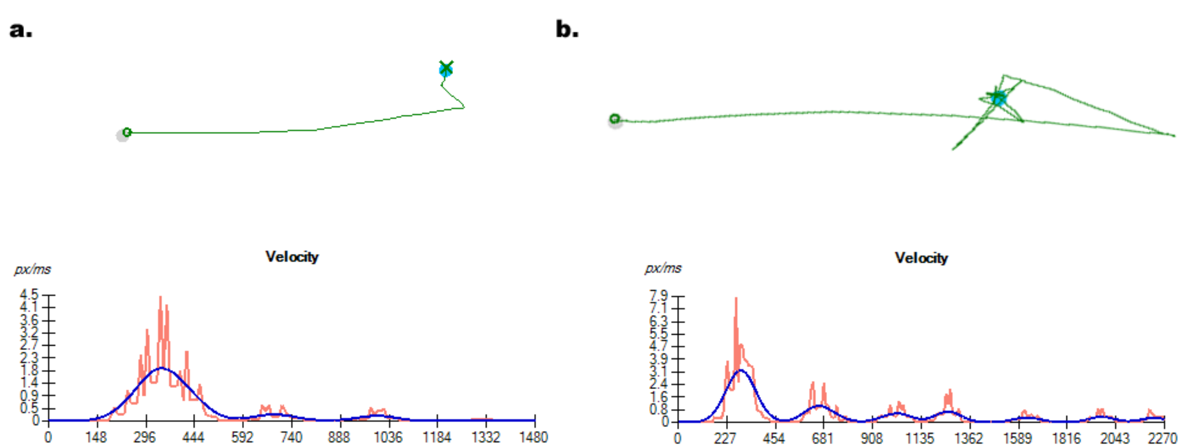
Like older adults and people with motor impairments, young children (ages 4-5) also have difficulty pointing with a mouse, especially as they near targets and try to stay within them (Hourcade, Bederson, Druin, & Guimbretière, 2004; Hourcade, 2006). *PointAssist* attempts to remedy this situation by observing when a child's pointing submovements become consecutively short and slow, and then reducing the C-D gain to magnify motor-space (Hourcade, Perry, & Sharma, 2008). PointAssist works in a target-agnostic fashion because it only examines pointing submovements, which are identifiable as the peaks in a movement's velocity profile (see Figure 5). PointAssist does not require knowledge of target locations, dimensions, or other properties. Excessive submovements have been noted as indicative of pointing problems in a number of studies, so using them as a signal to trigger an automatic accommodation is a powerful ability-based approach (Findlater *et al.*, 2010; Hourcade, 2006; Hwang, Keates, Langdon, & Clarkson, 2004; Ketcham, Seidler, Van Gemmert, & Stelmach, 2002; Walker, Philbin, & Fisk, 1997; Wobbrock & Gajos, 2008). Besides being useful to children, PointAssist was also found to help older adults point with greater accuracy (Hourcade, Nguyen, Perry, & Denburg, 2010). Most recently, PointAssist has been found to improve pointing performance for people with motor impairments as well (Salivia & Hourcade, 2013).

Related to PointAssist is our own *Angle Mouse*, which looks at the "spread" of angles created during a movement (Wobbrock et al., 2009). These angles represent the mouse cursor's movement direction, sampled periodically as the cursor moves. During the initial stages of an aimed pointing movement, even users with motor impairments often produce enough force to drive the mouse cursor in a fairly straight line, causing little spread in the angles created. But as users near targets and try to maneuver the cursor inside them, their angles diverge, often drastically. This divergence is measured, and the C-D gain is dropped accordingly, magnifying motor-space. The Angle Mouse is described in the next section.

Gravity, Forces, and Haptics

Related to C-D gain are *gravity wells*, also sometimes called *force fields*, which dynamically affect cursor position by pulling the mouse cursor into nearby targets. These techniques can be

Figure 5. (a) A classic aimed pointing movement containing three submovements. (b) A challenged aimed pointing movement containing seven submovements. Submovements can be seen as peaks in the smoothed velocity profiles. Note that most of the time taken occurs after the initial ballistic phase



implemented in software alone as *pseudo-haptic techniques* (Ahlström, Hitz, & Leitner, 2006; Hurst, Mankoff, Dey, & Hudson, 2007; Lécuyer, Coquillart, Kheddar, Richard, & Coiffet, 2000; Lécuyer, Burkhardt, & Etienne, 2004), or in combination with specialized haptic input devices such as haptic mice (Akamatsu & Sato, 1994; Akamatsu & MacKenzie, 1996; Dennerlein, 2001; Münch & Dillmann, 1997) or haptic trackballs (Engel, Goossens, & Haakma, 1994; Keuning, Monné, IJsselsteijn, & Houtsma, 2005; Keyson, 1997), input devices that provide tactile sensations to users by way of physical actuators.

Gravity wells and force fields may be "passive," such that they amplify motion already underway provided that motion occurs near or towards a target. Alternatively, gravity wells and force fields may be "active," such that they pull even an otherwise stationary cursor into targets. An extreme form of a gravity well is snapping, common in many drawing programs, for which an object being dragged by the cursor (e.g., the endpoint of a line segment) jumps to attach itself to a handle as it draws near a target (Bier & Stone, 1986; Feiner, Nagy, & Van Dam, 1981; Hudson, 1990; Sutherland, 1963). Such behavior can be thought of as an "instantaneous active gravity well." The chief drawback of traditional snapping—making certain screen coordinates inaccessible—was remedied by using increased motor-space at the snap location, that is, turning it into a sticky icon (Baudisch, Cutrell, Hinckley, & Eversole, 2005).

Although there has been substantial work in the use of haptic input devices for gaming, telesurgery, and robotics, haptic input devices have been explored less thoroughly for aiding people with motor impairments. Work specifically investigating the use of haptic input devices for people with motor impairments has shown that force-feedback mice, in combination with gravity wells around targets, can substantially reduce both target acquisition time and error rate, even in the presence of multiple targets that serve as "distractors" (Holbert & Huber, 2008; Hwang, Keates, Langdon, & Clarkson, 2003; Keates, Langdon, Clarkson, & Robinson, 2000; Langdon, Keates, Clarkson, & Robinson, 2000; Langdon, Hwang, Keates, Clarkson, & Robinson, 2002a; Langdon, Hwang, Keates, Clarkson, & Robinson, 2002b). Unfortunately, despite the promise of haptics for people with motor impairments, haptic devices and gravity wells have not been deployed in commercial interfaces, and haptic input devices have not found wide adoption. A practical challenge is whether a full-blown interface could remain "haptically intelligible" as myriad targets imbued with gravity create an overwhelming landscape of forces. As with many interaction techniques, real-world deployments are necessary to investigate such questions (Dixon et al., 2012). Also, it should be noted that the approach of using custom hardware is not in keeping with efforts to use cheap everyday input devices and commodity hardware in ability-based design. The computer games industry has provided some impetus for cheap haptic mice and joysticks, but it remains unclear whether such devices, if coupled with appropriate software, could significantly improve the accessibility of commercial systems for people with motor impairments.

Pointing on Touch Screens

In some respects, *all* users are motor-impaired when using small touch screens because fingers are fat, screens are small, and devices are less physically stable than desktop computers. While there have been a variety of pointing facilitation techniques created for touch screens (Benko *et al.*, 2006; Olwal, Feiner, & Heyman, 2008; Ramos, Cockburn, Balakrishnan, & Beaudouin-Lafon, 2007; Vogel & Baudisch, 2007), only a few techniques have been created explicitly for people with motor impairments. One such technique is *Swabbing*, which enables users with intention tremor to slide their finger on a touch screen toward their intended target to indicate it (Mertens, Jochems, Schlick, Dünnebacke, & Dornberg, 2010). By sliding on the screen, a user's tremor is dampened from constant pressure, the same principle at work in the stylus-based *EdgeWrite* text entry method (Wobbrock, Myers, & Kembel, 2003). In a formal evaluation comparing Swabbing to tapping, Swabbing was found to be more accurate and satisfying, but not faster for older users (Wacharamanotham, Hurtmanns, Mertens, Kronenbuerger, Schlick, & Borchers, 2011). Others have conducted studies comparing tapping on targets, crossing over targets, exiting targets, and directional gesturing (Guerreiro, Nicolau, Jorge, & Gonçalves, 2010a; Guerreiro, Nicolau, Jorge, & Gonçalves, 2010b). Results indicate that many people with motor impairments can both tap and cross successfully, but that the minimum recommended target size is 12 mm instead of 7-10 mm, target sizes usable by most able-bodied people but not by most people with motor impairments (Lee & Zhai, 2009; Parhi, Karlson, & Bederson, 2006).

Studies also show that targets placed on the screen edges are more easily acquired than targets placed elsewhere (Perry & Hourcade, 2008). This result is consistent with the oft-cited finding that on the desktop, Macintosh-style menus are easier to acquire due to their placement along an impenetrable screen edge than Windows-style menus, which lack such an edge (Accot & Zhai, 2002; Appert, Chapuis, & Beaudouin-Lafon, 2008; Walker & Smelcer, 1990). Our *barrier pointing* project, described in the next section, utilizes the elevated physical edges around some mobile device screens to aid stylus motion when selecting targets (Froehlich, Wobbrock, & Kane, 2007). *Barrier targets* placed along screen edges and activated on take-off can help people with motor impairments become more accurate using styli.

Another technique for making stylus-based interactions more successful, in this case for older adults, is the *Steadied-Bubble* (Moffatt & McGrenere, 2010). This technique is based on *Steady Clicks* (Trewin, Keates, & Moffatt, 2006), a desktop technique that freezes the mouse cursor whenever a mouse button is held down to prevent slipping and accidental dragging (Trewin, 1996; Trewin & Pain, 1999). In their pen-based adaptation, Moffat and McGrenere (2010) created the Steadied-Bubble by combining Steady Clicks and the Bubble Cursor. Designed for use on Tablet PCs that have a pen-hover state, the Steadied-Bubble employs a Bubble Cursor centered at the pen location during hover. When the pen descends to the screen, the bubble is "steadied," freezing in place with the nearest target captured. Slipping with the pen does not, therefore, move the bubble. And the bubble itself reduces miss errors because of its ability to capture nearby targets. Results for older adults show that the Steadied-Bubble made fewer target-misses and fewer target slip-offs than the conventional stylus, Steady Clicks alone, or the Bubble Cursor alone.

In addition to the foregoing touch screen techniques created for people with motor impairments, there have been a few recent techniques created to address situational impairments that affect motor control while on-the-go (Sears et al., 2003, 2008). When walking, users' input performance may be compromised as workload increases relative to standing or sitting (Lin, Goldman, Price, Sears, & Jacko, 2007; MacKay, Dearman, Inkpen, & Watters, 2005). One touch screen technique that can address this issue is *Bezel Swipe*, which interprets gestures that move from the screen bezel *onto* the screen as having meanings distinct from other taps and swipes that originate on the screen (Roth & Turner, 2009). Although initially invented to support actions like cut, copy, paste, and multiple selection, generalized *bezel gestures*, specifically right-angle gestures, were found to be faster and more accurate than button-tapping for walking, distracted users (Bragdon, Nelson, Li, & Hinckley, 2011).

Researchers have also focused on how to enable the thumb holding a mobile device to reach all targets on the screen. With *ThumbSpace*, a user drags her thumb—the one from the same hand holding the device—along a diagonal that delineates the extent of the thumb's reach (Karlson & Bederson, 2007). This area then becomes a semi-transparent motor-space thumbnail that maps absolutely to the entire screen such that when a user's thumb touches the miniaturized space, the corresponding object on the screen highlights. A user can slide his thumb to adjust his selection as needed and then use take-off selection to acquire targets. Similarly, with *AppLens* and *LaunchTile*, a user's thumb articulates gestures to control interface zooming levels for accessing on-screen application tiles (Karlson, Bederson, & SanGiovanni, 2005). AppLens operates using a fisheye view, whereas LaunchTile operates using traditional zoom-and-pan.

Finally, our *WalkType* project makes mobile text entry on touch screens more accurate using machine learning to classify touches as accurate key-presses (Goel et al., 2012).WalkType combines properties of touches on the touch screen with readings from built-in accelerometers to classify each touch according to its most likely key. In an evaluation with walking users, WalkType reduced typing errors by about half while increasing typing speed by over 10%. Researchers are just beginning to explore the benefits of using machine learning to make touch more accurate (Weir, Rogers, Murray-Smith, & Löchtefeld, 2012), an approach that could be employed for people with motor impairments in the near future.

Pointing with the Eyes

Given the importance and prevalence of pointing, it is no surprise that researchers have explored pointing with human capabilities other than motor control. For example, projects have explored pointing with the eyes for people with disabilities (Hutchinson, White, Martin, Reichert, & Frey, 1989). Basic techniques for target selection include dwelling with the eyes, also known as fixating, or pressing a button or key to select the currently looked-at target. Dragging an object is also possible when using a button to signal when to pick up and put down the eye-controlled object, or when using a progressive multi-state dwell cursor that enables dragging as well as clicking (Jacob, 1990, 1991; Lankford, 2000; Salvucci & Anderson, 2000). *EyeDraw*, a drawing program for children with motor impairments, uses a two-state cursor that delineates "looking" from "drawing" (Hornof & Cavender, 2005). Techniques for eye-based computer input are reviewed in greater detail elsewhere (Jacob & Karn, 2003; Majaranta & Räihä, 2002).

For improving the accuracy of eye-based pointing, *zooming* has been used to increase target size (Bates & Istance, 2002; Kumar, Paepcke, & Winograd, 2007; Lankford, 2000). One form of zooming that retains peripheral context is the fisheye lens, which has been used in eye-pointing in such a way that the fisheye appears during fixations but not during traversals across the screen (Ashmore, Duchowski, & Shoemaker, 2005). Another accuracy-enhancing scheme is target expansion coupled with an approach to handling jitter whereby the gaze-point does not need to not stay perfectly within the target during a fixation, but simply must not make a saccade elsewhere (Miniotas, Špakov, & MacKenzie, 2004). *MAGIC Pointing* and similar techniques rely on eye-gaze to place the mouse cursor in the general vicinity of the desired target, but leave final positioning inside the target to the mouse (Yamato, Monden, Matsumoto, Inoue, & Torii, 2000; Zhai, Morimoto, & Ihde, 1999). To work effectively for people with motor impairments, MAGIC Pointing would probably require PointAssist, the Angle Mouse, or something similar.

Pointing with the Voice

Researchers and developers have also enabled the human voice to acquire targets on computer screens. Three commercial speech-based target acquisition techniques are *mouse motion voice commands*, *Mouse Grid*, and *Show Numbers* (Nuance Communications, 2011; Odell & Mukerjee, 2007; Pugliese & Gould, 1998). Mouse motion voice commands—what some researchers have called the "Speech Cursor" (Harada, Landay, Malkin, Li, & Bilmes, 2006)—are provided in Nuance's *Dragon Naturally Speaking* product and produce mouse movement with commands like "move mouse up" and "move mouse right." The mouse cursor moves at a steady pace until the user says "stop." While these commands may be adequate for fine pointing adjustments, they are too slow for rapidly traversing the screen, and are meant to be complemented by the Mouse Grid. When a user says "mouse grid," he or she is presented with a nine-cell grid, each assigned a digit (1-9). When the user speaks a digit, the entire grid shrinks to occupy the spoken cell, thereby enabling the user to recursively subdivide the screen until the desired click location is reached. In the Show Numbers technique, when the user says "show numbers," every target registered with the accessibility framework is displayed with an overlaid numeric value that, when spoken, is tantamount to clicking on that target. Unfortunately, not all possible targets are known to the system in this manner, as it requires developers to fully comply with the accessibility framework. Also, it is not possible to click on an arbitrary screen coordinate with this approach, unlike with the first two approaches.

Neither Show Numbers nor Mouse Grid supports the ability to steer along smooth paths. For that, *non-speech* voice-based input techniques have been explored. For example, projects have taken advantage of humming (Igarashi & Hughes, 2001; Sporka, Kurniawan, Mahmud, & Slavík, 2006; Sporka, Kurniawan, & Slavík, 2006) and whistling (Sporka, Kurniawan, & Slavík, 2004). Our own *VoiceDraw* project, in the same spirit as EyeDraw before it, enables people with motor impairments to move a paintbrush across a canvas by uttering the vowel sounds defined by the *Vocal Joystick* (Harada et al., 2006; Harada, Wobbrock, & Landay, 2007). Our VoiceDraw project is described in the next section.

Having toured numerous approaches to improving pointing for people with motor impairments, children, and older users, we now turn to our own attempts to make progress on this important topic. Informed by ability-based design, we investigated approaches to changing interfaces to make them better suited to users' abilities.

MAKING POINTING ACCESSIBLE FOR PEOPLE WITH MOTOR IMPAIRMENTS

Motivated by a desire to change systems to better match users' abilities, and to take full advantage of the abilities users have, we sought not to alter users to fit systems, but to question fundamental aspects of modern graphical user interfaces, and to investigate "new ways of doing old things." The mantra for this section is, *"change not thy user, change instead thy…,"* and we complete this mantra with the following possibilities:

- Mouse Cursor
  - Angle Mouse
  - Pointing Magnifier
- Target Types
  - Accessible goal crossing
  - Barrier pointing
- Interface Design
  - Ability-based user interfaces
  - Walking user interfaces
- Input Device
  - Non-speech voice-based control

The above list serves as an outline of subsections to follow. Each project has its strengths and weaknesses, and each project therefore constitutes a set of tradeoffs in the design space of possibilities. Each project also embodies at least some of the principles of ability-based design (Table 1), which will be drawn out in each project's description.

Changing the Mouse Cursor: The Angle Mouse and Pointing Magnifier

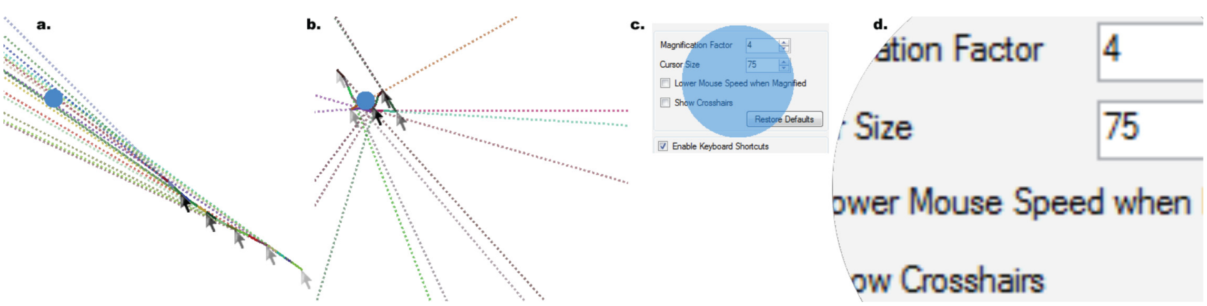
Pointing comprises a *ballistic phase*, when the cursor "shoots out" toward the target, and a *corrective phase*, when final adjustments are made to maneuver the cursor inside the target. When trying to minimize overall pointing time, these phases are in tension: faster ballistic movements reduce accuracy and therefore result in more corrective submovements, while slower ballistic movements are more accurate but themselves take more time. Optimizing this tradeoff so as to reduce overall movement time is at the heart of Meyer's leading theory of human pointing performance and provides the best explanatory account of Fitts' law (Fitts, 1954; Meyer et al., 1988; 1990).

People with motor impairments, older adults, and children often have difficulty in the corrective phase of pointing (Findlater et al., 2010; Hourcade, 2006; Hwang et al., 2004; Ketcham et al., 2002; Salivia & Hourcade, 2013; Walker et al., 1997; Wobbrock & Gajos, 2008). Whereas their ballistic phase often contains sufficient force to overwhelm any neuromuscular noise in their motor system, and thus produce relatively straight motion, the "light touch" required in the corrective phase is often corrupted by neuromuscular noise (Harris & Wolpert, 1998; Walker et al., 1997), causing overshoots, undershoots, and offshoots that demand even more corrective motion. The result can be extremely frustrating for users as they wrestle with sometimes dozens of repeated, fatiguing submovement corrections.

Based on our observations of the above phenomenon, the *Angle Mouse* (Figures 6a-b) attempts to alleviate users' struggles with corrective motion by magnifying motor-space as the cursor attempts to move inside a target (Wobbrock et al., 2009). Importantly, the Angle Mouse is target-agnostic, examining only the mouse cursor's movement to determine when to magnify motor-space. As its name suggests, the Angle Mouse observes the angles created during a pointing movement. It gathers these angles by repeatedly sampling movement points at least 8 pixels apart and determining the angle between them. When the deviation of angles is low, the C-D gain is kept high, and the assumption is that the user is moving ballistically toward a target. However, when the deviation of angles is high, the C-D gain is dropped proportionally, magnifying motor-space. High angular deviation occurs mostly when users perform corrective motion, trying to place their mouse cursor inside a target in the final phase of movement.

In a study of 16 people comparing the Angle Mouse to sticky icons and to the default point cursor, the Angle Mouse was found to exhibit over 10% higher throughput[5] for people with motor

Figure 6. The Angle Mouse with angle visualizations during movement with (a) low angular deviation, and (b) high angular deviation. The Pointing Magnifier and part of its configuration dialog when it is (c) an unmagnified area cursor, and (d) a magnified lens



impairments. The Angle Mouse was also significantly more accurate and led to significantly fewer submovement corrections than the default point cursor. For able-bodied users, Angle Mouse throughput was not detectably different from that of the default point cursor, confirming that the Angle Mouse only "kicked in" for users exhibiting motor-control difficulties.

The principles of ability-based design upheld by the Angle Mouse are:

1. **Ability:** Many people with motor impairments *can* point successfully in graphical user interfaces when given targets magnified at the right time in motor-space.
2. **Accountability:** With the Angle Mouse, it is the software's job to accommodate the observed pointing behavior of the user and make it more accurate.
3. **Adaptation:** The Angle Mouse continually adapts the C-D gain based on the behavior of the user.
4. **Transparency:** The Angle Mouse exposes many settings that govern it, allowing the user to change those settings. The Angle Mouse also visualizes how those settings affect adaptation during movement.
5. **Performance:** The Angle Mouse continually monitors the angular deviation of mouse movements.
7. **Commodity:** The Angle Mouse is a freely available[6] software-only solution for use with everyday input devices like mice, touchpads, and trackballs.

The same observations of users' difficulties with corrective-phase motion inspired the creation of the *Pointing Magnifier* (Figures 6c-d) (Jansen, Findlater, & Wobbrock, 2011). The Pointing Magnifier, first described as the *Visual-Motor-Magnifier* enhanced area cursor (Findlater et al., 2010), is a semitransparent circular area cursor of arbitrary size that operates in three stages. First, the user places the area cursor over the desired target. Second, the user clicks, causing the contents beneath the area cursor to magnify by a user-defined amount. Third, the user moves a conventional point cursor within the magnified view to their enlarged target and clicks. Thus, the Pointing Magnifier trades two clicks for one compared to the default point cursor, but removes the need for corrective motion due to having a large area cursor and enlarged targets. Because the Pointing Magnifier magnifies not just visual-space but also motor-space, it provides an aid to pointing unlike typical screen magnifiers, which magnify only visual-space and therefore do not aid motor performance. Like the Angle Mouse, the Pointing Magnifier is target-agnostic, as screen pixels can be magnified and mouse clicks can be routed from the magnified view to the proper screen coordinates without any knowledge of underlying targets.

In a study of 12 people with motor impairments by Findlater et al. (2010)[7], compared to the Bubble Cursor and the default point cursor, the Pointing Magnifier significantly reduced selection time of small, densely packed targets by about 26% and 34%, respectively. It also reduced the error rate by about 55% and 80%, respectively. As with the Angle Mouse, submovement corrections were significantly reduced with the Pointing Magnifier compared to the default point cursor. Subjective responses by users indicated that the Pointing Magnifier was liked best, while no users preferred the Bubble Cursor. The default point cursor was the most disliked. The Pointing Magnifier also required less mental effort, less overall effort, and was less frustrating than the default point cursor.

The principles of ability-based design upheld by the Pointing Magnifier are:

1. **Ability:** Many people with motor impairments *can* point successfully in graphical user interfaces when given large area cursors that work as visual-and-motor-space lenses to magnify targets.

2. **Accountability:** With the Pointing Magnifier, it is the software's job to magnify pixels and route mouse clicks, enabling users to point successfully with whatever dexterity they have.
3. **Adaptation:** The Pointing Magnifier, while not adaptive, is highly adaptable, providing a range of settings to the user, including the size of the semitransparent area cursor and the magnification factor of the lens, which together determine the size of the resulting magnified view.
7. **Commodity:** The Pointing Magnifier is a freely available[8] software-only solution for use with everyday input devices like mice, touchpads, and trackballs.

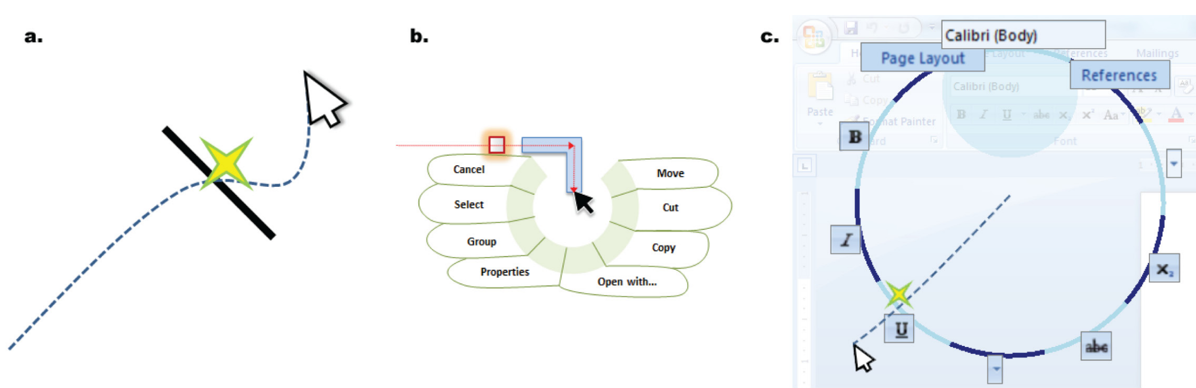
Interestingly, a community of graphic artists and graphic designers have found the Pointing Magnifier to be a valuable tool for locally magnifying portions of their canvases in programs like Adobe Photoshop, which provides only global magnification of the entire canvas, sacrificing context. In creating the Pointing Magnifier for people with motor impairments, we certainly did not anticipate its usefulness as a graphics editing tool, but are pleased to learn of this unanticipated use!

Changing the Target Types: Accessible Goal Crossing and Barrier Pointing

The Angle Mouse and the Pointing Magnifier both enable motor-impaired users to point-and-click more accurately, but pointing-and-clicking itself is worth questioning. *Goal crossing* (Figure 7) has been proposed as an alternative for able-bodied users, usually for use on pen-based touch screens (Accot & Zhai, 1997; Accot & Zhai, 2002; Apitz & Guimbretière, 2004). With goal crossing, users do not click on a specific display region, but instead, pass over a threshold or boundary, called a "goal." This scheme allows users to accelerate over a threshold and continue for whatever distance is comfortable. Pointing-and-clicking, on the other hand, requires uses to decelerate into a bounded region to issue a click, which may slip off the target (Trewin, 1996; Trewin & Pain, 1999; Trewin et al., 2006). Despite the known advantages of goal crossing for general use, until our efforts, goal crossing had not been investigated as an accessibility strategy for people with motor-control difficulties.

We first sought to understand whether mouse-based goal crossing offered any human performance benefits compared to pointing-and-click-

Figure 7. (a) The concept of goal crossing. (b) A mouse-based crossing design for accessing a context menu, inspired by Hover Widgets (Grossman, Hinckley, Baudisch, Agrawala & Balakrishnan, 2006). (c) Turning pointing targets into crossing targets in Microsoft Word with the Click-&-Cross cursor



ing. We conducted a study of 8 able-bodied people and 8 motor-impaired people (Wobbrock & Gajos, 2007, 2008). Of the latter group, 4 people used mice, 2 people used trackballs, and 2 people used both. To equate our two different target types, the lengths of our crossing goals were made to match the diameters of our circular area targets. No mouse buttons were used while crossing. Our results showed that goal crossing was significantly faster than pointing-and-clicking for both participant groups, but it was also more error prone. (It is worth noting that crossing errors were defined very strictly: passing by a goal anywhere outside its delineated line segment was considered an error.) Goal crossing had a higher throughput than pointing-and-clicking for people with motor impairments (2.88 *vs*. 2.34 bits/s), but the opposite was true for able-bodied users (3.61 *vs*. 4.72 bits/s). Path analyses showed that for the motor-impaired users, there were significantly fewer task axis crossings, movement direction changes, and orthogonal direction changes with crossing than with pointing; there was also significantly less movement variability, movement error, movement offset, and path length with crossing than with pointing (Keates, Hwang, Langdon, Clarkson, & Robinson, 2002; MacKenzie, Kauppinen, & Silfverberg, 2001). Subjective responses indicated that motor-impaired users liked goal crossing more and thought it was easier than pointing-and-clicking; the reverse sentiments were true of the able-bodied group. Thus, for users with motor impairments, goal crossing indeed seemed to offer significant advantages in speed, accuracy, efficiency, movement control, and desirability than pointing-and-clicking.

With our study findings clearly showing the advantages of goal crossing for people with motor impairments, the question became whether practical user interface designs could be created for mouse cursor-based interfaces using goal crossing as the fundamental method of target acquisition (see, e.g., Figure 7b). Although pen-based crossing interfaces have been shown to be viable (Apitz & Guimbretière, 2004), a pen, stylus, or finger can "fly in," cross a target, and "fly out." Not so for mouse cursor-based interfaces because the mouse cursor is persistent and cannot move to a location without crossing everything in its path. Depressing mouse buttons while crossing has been explored for able-bodied users (Cockburn & Firth, 2003), but such maneuvers amount to *dragging*, which is more difficult than pointing, especially when using trackballs (MacKenzie, Sellen, & Buxton, 1991; Trewin, 1996). A different approach is needed.

Initially, we sought to embed goal crossing functionality within *crossing widgets* (Choe, Shinohara, Chilana, Dixon, & Wobbrock, 2009). We devised general crossing schemes that would allow intentional crosses to be distinguished from unintentional ones (Wobbrock & Gajos, 2008). For example, to activate a button, a user might cross it and cross back over it, or cross and make a gesture (e.g., a pigtail, a 90°-turn), or even cross and click *anywhere*. We found from generating myriad possible schemes, however, that a fundamental tension emerges: crossing widgets that are easy to operate are easy to accidentally trigger, but crossing widgets that are safe from accidental triggering are too difficult to operate intentionally (Choe et al., 2009). Put another way, "ease-of-operation" and "error prevention" are at fundamental odds. We also found that dense target layouts, like toolbars, make any crossing scheme difficult because crossing, by its very nature, requires screen real-estate for the cursor's approach and follow-through.

Our successful solution finally came when we decided to only *temporarily* create crossing goals out of conventional widgets when they are needed. We created two enhanced area cursors, the *Click-&-Cross* cursor (Figure 7c) and the *Cross-&-Cross* cursor, which are semitransparent area cursors that, when triggered, temporarily turn every overlapped target into a crossing goal (Findlater et al., 2010). Crossing a widget's goal activates the widget that owns it. The trigger for the Click-&-Cross cursor is a mouse click. The trigger for the Cross-&-Cross cursor is to cross the back edge of the cursor itself, the edge opposite the point cursor, which persists inside the Cross-&-Cross cursor and drags it around like a *tracking menu* (Fitzmaurice, Khan, Pieké, Buxton, & Kurtenbach, 2003). Thus, in the Cross-&-Cross cursor, no clicking whatsoever is required. Both crossing cursors are target-aware, as they must know which widgets they overlap in order to turn those widgets into temporary crossing goals.

In a study of 12 people with motor impairments (Findlater et al., 2010), we showed that the Cross-&-Cross cursor was slow, but the Click-&-Cross cursor was significantly faster than the Bubble Cursor and the default point cursor for small, dense targets—about 33% and 40%, respectively. Furthermore, the speed of the crossing cursors did not increase with decreasing target size or inter-target spacing. Both crossing cursors were also more accurate for small, dense targets than the Bubble Cursor and the default point cursor—by about 70% and 85%, respectively. Like the Angle Mouse and the Pointing Magnifier, the Click-&-Cross cursor produced significantly fewer submovements than the default point cursor. Subjective responses showed that the Pointing Magnifier (Figures 6c-d) was liked more than either the Click-&-Cross (Figure 7c) or Cross-&-Cross cursors, which were liked about the same and still more than the Bubble Cursor (Figure 3c) or default point cursor. However, subjective workload ratings indicated the crossing cursors both required more effort than the Pointing Magnifier or Bubble Cursor, but still less effort than the default point cursor.

The principles of ability-based design upheld by the accessible goal crossing projects are:

1. **Ability:** Many people with motor impairments *can* cross goals better than they can point-and-click on bounded regions of the screen.
2. **Accountability:** With accessible goal crossing, interfaces' on-screen targets are changed to become crossing goals, either permanently or just-in-time for selection.
3. **Adaptation:** The crossing cursors, while not adaptive, are both highly adaptable, providing a range of settings to the user, including the sizes of their semitransparent area cursors, the maximum number of targets they are permitted to overlap, and the radii of their circles that contain their temporary crossing arcs.
7. **Commodity:** The accessible goal crossing project employs software-only solutions for use with everyday input devices like mice, touchpads, and trackballs.

Related to crossing a goal is acquiring a target along an impenetrable boundary, or *edge*. Edges are known to improve target acquisition because they serve as a "backboard" for cursors, pens, or fingers, allowing them to accelerate into targets and effectively overshoot by arbitrary amounts (Appert et al., 2008; Farris, Jones, & Anders, 2001, 2002a, 2002b; Johnson, Farris, & Jones, 2003; Perry & Hourcade, 2008; Walker & Smelcer, 1990). Edges are also known to improve the accuracy of gestures (Wobbrock, 2003). The EdgeWrite text entry method for touch screens, touchpads, and joysticks takes full advantage of edges and corners by defining a unistroke alphabet whose characters are made by moving along the edges of a small plastic square hole, providing physical stability to people with motor impairments, especially tremor (Wobbrock et al., 2003; Wobbrock, Myers, & Aung, 2004; Wobbrock, Myers, Aung, & LoPresti, 2004). EdgeWrite served as the inspiration for barrier pointing.

Barrier pointing attempts to take advantage of the elevated physical edges surrounding the touch screens of many mobile devices (Froehlich et al., 2007). *Barrier targets* are placed along the edges of the screen such that the impenetrable physical barrier sits behind them and aids in their selection. Many different edge-based selection schemes are possible: normal tapping; sliding along the edge and taking-off from the desired target; sliding along the edge, pausing within the desired target, and then shooting rapidly into a nearby corner; and, sliding along the edge and changing direction inside the desired target.

In a study of the above selection techniques along with conventional targets as a baseline, we found that subjects with poor coordination and spasticity were most helped by barrier pointing, in some cases making subjects dramatically more accurate. For example, one subject with spastic cerebral palsy missed 66.7% of the time with conventional targets placed in the center of the screen, but with barrier targets and take-off selection, he missed only 13.3% of the time, and his selections were 48.5% faster besides. Another subject with tetraplegia from a spinal cord injury missed over 29% of the time with conventional targets, but only 4.2% of the time with barrier targets and take-off selection, and his selections were 40.5% faster besides. Thus, it seems that for some users, the ability to place continual pressure on the screen while sliding along an edge is of tremendous benefit in terms of stability and speed. Placing continual pressure on the screen also has been exploited in recent designs for motor-impaired pointing on touch screens (Guerreiro et al., 2010a; Wacharamanotham et al., 2011).

The principles of ability-based design upheld by barrier pointing are:

1. **Ability:** Many people with motor impairments *can* select targets on touch screens using styli, provided systems use barrier targets with take-off selection instead of conventional targets placed without a supporting edge.
2. **Accountability:** With barrier pointing on a touch screen device, conventional targets give way to barrier targets selected using take-off selection.
7. **Commodity:** Barrier pointing is a software-only solution for use with everyday touch screen devices that have elevated screen edges and styli.

Changing the Interface Design: Ability-Based User Interfaces and Walking User Interfaces

Many of the approaches outlined thus far for improving motor-impaired pointing performance in graphical user interfaces have made changes to either the mouse cursor or the targets on which the cursor acts. These are very "local" changes, but more "global" changes—namely, adapting the layout and design of interfaces as a whole—can also improve pointing performance. Ability-based user interfaces and walking user interfaces do this.

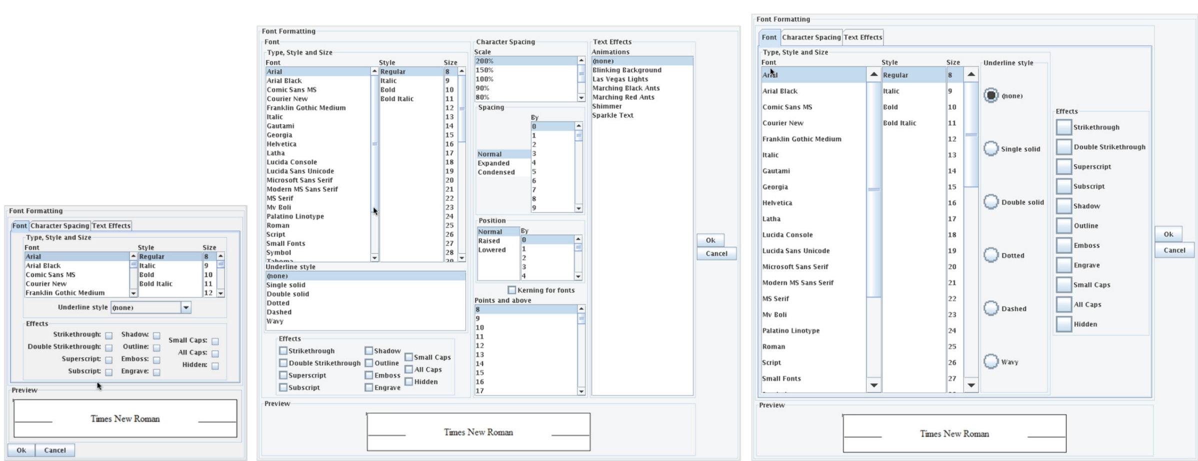
The pioneering work giving rise to *ability-based user interfaces* is the SUPPLE automatic user interface generator (Gajos & Weld, 2004). Although automatic user interface generation has been pursued using heuristics, templates, and constraints for a long time, SUPPLE was the first generator to frame the problem in terms of cost optimization. The "cost" of an interface in SUPPLE was initially based on how well the interface honored device constraints, usage traces, and preferences expressed by users in the ARNAULD system, which presented A/B interface comparisons to users (Gajos & Weld, 2005; Gajos, Long, & Weld, 2006). But users' preferences may not reflect users' actual abilities, and for ability-based user interfaces, preferences are exchanged for performance (Gajos et al., 2007, 2008, 2010). SUPPLE first presents batteries of pointing, dragging, clicking, and list-operation trials to users. Then it measures and models users' performance, using that information to custom-generate a user interface whose layout, widgets, widget groupings, and widget sizes are all chosen to minimize the total operation time of the interface by the user. Thus, the resulting interfaces are "ability-based" in that it is users' measured abilities, not just their preferences, that determine the generated interfaces without any declarative or heuristic knowledge about users' health or impairment. Figure 8 shows examples of generated interfaces.

In a study of 11 people with motor impairments (Gajos et al., 2008), SUPPLE's ability-based user interfaces were about 20% faster and 64% more accurate than its preference-based user interfaces, and about 26% faster and 73% more accurate than manufacturers' default interfaces. (Example defaults were the dialog boxes for formatting fonts and for printing from Microsoft Word 2003). Furthermore, users felt that ability-based interfaces were most preferred, "and they found those interfaces the easiest to use, the most efficient, and least physically tiring" (1265). Thus, without any actual pointing facilitation technique, subjects were made faster and more accurate simply by changing interface layout, widget types, and widget sizes.

The principles of ability-based design upheld by the ability-based user interfaces generated by SUPPLE are:

1. **Ability:** Many people with motor impairments *can* point successfully in graphical user interfaces when interfaces are generated using layouts, widgets, and widget sizes that optimize people's performance.
2. **Accountability:** It is SUPPLE's job to accommodate the measured pointing abilities of users to make users more efficient.
3. **Adaptation:** SUPPLE creates custom personalized interfaces for each user based on that user's measured abilities.
4. **Transparency:** SUPPLE gives users the power to inspect and override the choices it makes.
5. **Performance:** SUPPLE measures the performance of users in pointing, dragging, clicking, and list-operation trials. It models this performance and uses it to drive adaptation.
7. **Commodity:** SUPPLE is a software-only solution for use with everyday input devices like mice, touchpads, and trackballs.

Figure 8. Microsoft's default font-formatting dialog box and two automatically-generated interfaces from SUPPLE, one for a person with muscular dystrophy and one for a person with cerebral palsy. Images courtesy of Krzysztof Z. Gajos

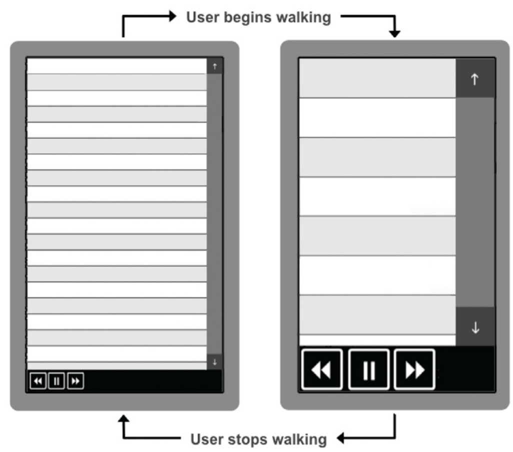


Recall that ability-based design places emphasis on abilities exercisable in a given context. When working on desktop computers in office environments, contextual factors do not readily or drastically change: surfaces are stable, lighting is ample, seating is comfortable, temperatures are controlled, noises are minimal, and so on. But all of these factors (and more) are challenged by mobile contexts, whether for users walking, riding, or working in the field. In such cases, surfaces may be moving, lighting may be dim, seating may be missing, temperatures may be cold, and noises may be distracting. As computing moves ever-further from the desktop, systems will need to accommodate these issues. For ability-based design, we care specifically about how these contextual factors affect users' abilities to interact with their devices.

To explore whether interface adaptivity can improve people's performance while walking, we created the concept of *walking user interfaces,* which automatically adapt to improve pointing performance while users are walking. In two separate explorations, we created two distinct walking user interfaces.

Our first walking user interface is a touch-based mobile music player with a playlist of songs (Kane et al., 2008). When the user stands still, the interface displays small buttons, fonts, and playlist items. When the user begins walking, the interface increases the button size, font size, and list item height by a factor of about three (see Figure 9). Targets become bigger, but at the expense of fewer of them showing on the screen, thereby requiring more scrolling. In a study of 29 able-bodied subjects selecting and playing a fixed set of songs, we found that walking increased task time for static interfaces by 17.7%, but with our adaptive interface, walking did not affect task time. We also found that there was no penalty incurred for adaptation: the adaptive interface performed almost identically to its two component static interfaces in terms of speed and errors. This is noteworthy in light of prior concerns that adaptivity itself can result in performance decreases (Findlater & McGrenere, 2004; Mitchell & Shneiderman, 1989). Finally, we found that cloudy weather reduced task time by 18.5% compared to partly cloudy or sunny weather, validating the presence of weather-related situational impairments, in this case, glare from sunlight.

Figure 9. The targets within a walking user interface music player adapt based on a user standing or walking. Song names have been removed to enable the targets themselves to be more visible. Image courtesy of Shaun K. Kane



Our second walking user interface is called *WalkType*, which uses machine learning to improve the accuracy of mobile text entry on touch screens while walking (Goel et al., 2012). Using a decision tree, WalkType classifies key-presses on a mobile touch screen keyboard by incorporating touch features, accelerometer data, inference about gait, and key-center anchors (Gunawardana, Paek, & Meek, 2010). WalkType is invisible to the user, but can be regarded as changing the interface design because WalkType effectively decouples the motor-space from the visual-space of the on-screen keyboard, continually "redesigning" the motor-space layout of the interface as the user types. In a study of 16 able-bodied subjects, we found that WalkType reduced text entry errors while walking by 45.2% and increased text entry speed by 12.9%. In addition, 14 of 16 subjects preferred WalkType without ever knowing which condition employed WalkType and which did not.

The principles of ability-based design upheld by walking user interfaces are:

1. **Ability:** People *can* point more successfully even when experiencing situational impairments if interfaces adapt their target sizes to accommodate the effects of walking.
2. **Accountability:** With a walking user interface, it is the software's job to accommodate the walking-related situational factors affecting users to make users more efficient.
3. **Adaptation:** Our music player adapts its visual level-of-detail based on whether or not a user is walking. Similarly, WalkType uses a decision tree model to adapt the behavior of its keyboard as the user taps keys while walking.
5. **Performance:** WalkType builds a user-independent model based on the performance of users tapping keys while entering training phrases.
6. **Context:** Our music player adapts to the user's walking or standing context. Similarly, WalkType utilizes accelerometer data to infer when each foot strikes the ground, accommodating users' taps accordingly.
7. **Commodity:** Our music player and WalkType both utilize only commodity mobile devices and built-in sensors.

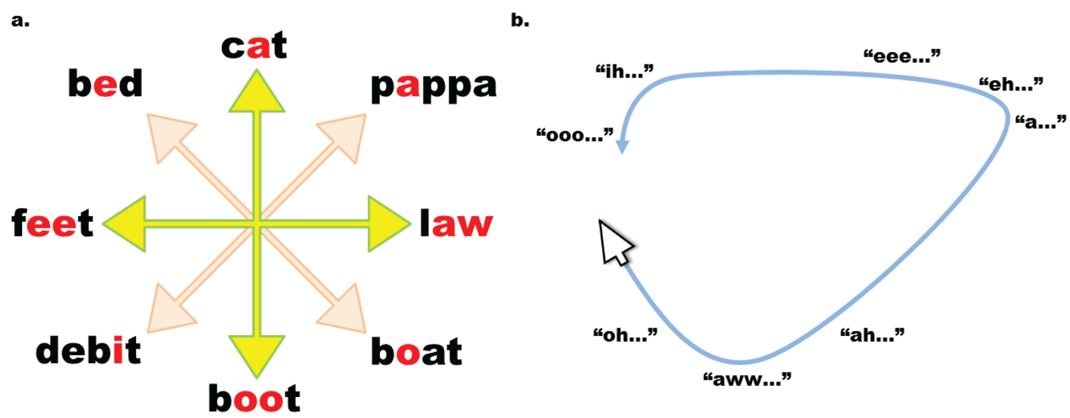
Changing the Input Device: Non-Speech Voice-Based Control

Our final strategy for improving pointing with ability-based design lies outside the software-only solutions of changing cursors, targets, and interface designs. This time, it is to change the hardware input device. Of course, this strategy runs the risk of imposing on users the need to procure expensive or customized hardware unless the "commodity" principle of ability-based design is upheld and everyday input devices can still be used. Input devices do not amount only to mice, touchpads, joysticks, and trackballs. An often neglected input device is the computer microphone, and with proper software, it can be used to facilitate pointing in graphical user interfaces for people with motor impairments.

Conventional speech recognition is unsatisfying for maneuvering smoothly through space because *discrete* words like "move left" and "move down" do not naturally map to *continuous* outcomes. When moving a mouse cursor with discrete commands, a speech recognition user may have to say "slower" or "faster" to alter the speed of a cursor already in motion. If the word "stop" is misrecognized, the moving cursor will fail to halt, creating errors and frustration. The flowing stroke of a master painter can hardly be reproduced by uttering discrete words in sequence.

Our work utilizes the ***Vocal Joystick***, a recognition engine for non-speech voice-based sounds that enables the smooth control of anything from a mouse cursor to a robotic arm (Bilmes, Li, Malkin, Kilanski, Wright, Kirchhoff, Subramanya, Harada, Landay, Dowden & Chizeck, 2005; Harada *et al.*, 2006). The Vocal Joystick responds to vocalizations of vowel sounds, where different sounds indicate different directions. For the cardinal directions, the "a" sound in "cat," when continually voiced, moves a mouse cursor straight up, or north. This sound morphs into the "aw" sound in "law" to move east, the "oo" sound in "boot" to move south, and the "ee" sound in "feet" to move west. The phonetic midpoints between these sounds move in the ordinal directions: the "e" sound in "bed" moves to the northwest, the "a" sound in "pappa" moves to the northeast, the "o" sound in "boat" moves to the southeast, and the "i" sound in "debit" moves to the southwest. Thus, an entire "vowel map" is formed that creates a circle, enabling fluid control of the mouse cursor in any direction by blending vowel sounds (*see* Figure 10).

Figure 10. (a) The Vocal Joystick vowel map for continuous movement control, and (b) a sample movement labeled with sounds at different stages



In a single-session laboratory study of four able-bodied experts, it was discovered that the Vocal Joystick, despite not being a hand-controlled input device, could nevertheless be modeled by Fitts' law (Harada et al., 2006). The throughput of the Vocal Joystick in this study was 1.65 bits/s, much lower than that of the mouse at 5.48 bits/s but similar to conventional rate-controlled joysticks (Epps, 1986; Epps, Snyder, & Muto, 1986). A follow-up study with nine novices showed that the Vocal Joystick was significantly faster than Speech Cursor and not detectably different than Mouse Grid after only five minutes of practice with each technique.

Learning the vowel map requires some time, and we wanted to see how learning unfolds, so we conducted a 2.5-week 10-session longitudinal study of five people with motor impairments and four able-bodied controls (Harada, Wobbrock, Malkin, Bilmes, & Landay, 2009). With 99 hours of data resulting from 90 total sessions, we found that all subjects learned the vowel map fluently by the fifth session, and by the tenth session, the motor-impaired group achieved 1.17 bits/s while the able-bodied group averaged 1.64 bits/s, nearly the same as the experts from the prior study. Thus, the subjects with motor impairments reached about 71% of the able-bodied expert level of performance.

The Vocal Joystick has provided the voice-recognition engine for other projects of ours, including *VoiceDraw* (Harada et al., 2007), a non-speech voice-based analog to EyeDraw (Hornof & Cavender, 2005); *Voice Games*, a controller for playing computer games hands-free (Harada, Wobbrock, & Landay, 2011); and *VoiceLabel*, a method of labeling mobile sensor data for training machine learning algorithms (Harada, Lester, Patel, Saponas, Fogarty, Landay, & Wobbrock, 2008).

The principles of ability-based design upheld by non-speech voice-based control are:

1. **Ability:** Many people with motor impairments *can* point successfully by using non-speech vocalizations to fluidly control a mouse cursor.
2. **Accountability:** With non-speech voice-based control, it is the software's job to accommodate the vocalizations of the user and move the mouse cursor fluidly in response.
3. **Adaptation:** The Vocal Joystick adapts itself to the vocalizations of the user so that it can better recognize the user's intentions.
5. **Performance:** The Vocal Joystick observes each user's performance in order to initialize a profile for each user. These user-specific profiles inform adaptation to make the Vocal Joystick perform better for each user.

7. **Commodity:** The Vocal Joystick is a freely available[9] software-only solution that uses the commodity microphone found with many desktops and built into most laptops.

REFLECTIONS AND FUTURE WORK

As this chapter has shown, there are many things that can be changed to improve pointing in graphical user interfaces for people with motor impairments besides the people themselves! For every technology described, users with motor impairments would approach computing systems in exactly the same way as their able-bodied counterparts, bringing nothing to augment or alter themselves—the whole solution would already exist "on the machine." Besides the practical benefits of this, there are social benefits as well, benefits that take us one step closer to being an integrated information society in which all people can participate. In our studies, users were generally happy to be using the computer "like everyone else." A number of subjects even made comments to that effect. Our aforementioned projects show just how much people with motor impairments *can* do when systems are changed in deep, fundamental ways. The attitude is no longer one of trying to make users "seem like" able-bodied people to computers, but of making computers fundamentally more flexible, aware of users' abilities, and bearing the burden of adaptation. It is the users who are accommodated, not the machines.

Out of seven projects, the principles of *ability*, *accountability*, and *commodity* were upheld by all seven. The principle of *adaptation* was upheld by six, *performance* was upheld by four, and *transparency* was upheld by two. Finally, the principle of *context* was only upheld by one, walking user interfaces, although two technology projects were described. (As our projects mostly concerned pointing in graphical user interfaces, it is natural to see a lack of emphasis on context. If more projects concerned mobile devices and off-desktop scenarios, the context principle would be more prevalent.)

It is important to realize that while every project had its successes, there were often a few subjects in our studies that fared poorly with the given innovations. And yet, with other innovations, those may be the same subjects to fare best of all. This highlights the well-known fact that people with motor impairments exhibit a wide range of abilities and no single solution is likely to be successful for all users, an observation that continues to challenge the premise of universal design and similar approaches.

One could imagine usefully combining many of the aforementioned projects to create even more useful ability-based systems that make pointing even easier for people with motor impairments. For example, a user could employ the Angle Mouse running invisibly "beneath" the cursor while the cursor itself was a crossing cursor that turned clicked-upon targets into crossing goals. One could also imagine barrier targets being used in the "walking" state of an adaptive walking user interface—as a user started moving, the most important targets would float to the sides of the screen and anchor themselves there. Finally, one could imagine just about any interface, whether desktop or mobile, pointing or crossing, manually-controlled or voice-controlled, being an ability-based user interface in the style of SUPPLE, allowing users to provide a performance model through a simple battery of trials that then informed the automatic generation of personalized interfaces.

Admittedly, it may be inconvenient, annoying, or fatiguing for users to have to perform explicit trials to build a performance model. A better approach would be to capture the performance of users "in the wild" as they work, without requiring an artificial test bed. Making precise measurements from everyday computer use is difficult, as it becomes difficult or even impossible to divine users' intentions at every step. We have made significant progress on this problem for measuring text entry and mouse pointing performance on desktop computers with the *Input Observer* tool (Evans & Wobbrock, 2011, 2012). This tool is capable of discerning text entry errors from everyday text editing, and can compute mouse pointing errors even without definitively knowing the user's intended target. To date, the tool has been validated only for able-bodied users who conform to certain movement patterns (Meyer et al., 1988, 1993), which may not apply to people with motor impairments. Other approaches using machine learning may address this challenge (Gajos, Reinecke, & Herrmann, 2012; Hurst, Hudson, Mankoff, & Trewin, 2008).

However it may be obtained, a user's performance model could, in the future, be used to inform *every* device about the user who is using it. Such "ability profiles" could be built up over ongoing everyday use, and could follow a user to whatever device or platform he or she is currently using. The concept of transferrable ability profiles has only begun to be explored (Montague, Hanson, & Cobley, 2012).

Being the capstone project that gave rise to ability-based user interfaces, and ultimately ability-based design, Supple upheld six of the seven ability-based design principles, lacking only the *context* principle. Just as Supple has the capacity to integrate both preference-based feedback from Arnauld and performance-based feedback from trial batteries, Supple could also integrate mobile device sensor readings such as light levels, temperatures, accelerometer values, and so on. Ultimately, the concern over context is due to the need to anticipate, rather than purely react to, changes in users' performance. For example, it is better to increase touch screen button size due to cold weather than to wait for cold weather to cause a user's fingers to lose their dexterity. Future work is needed to discover how context affects abilities in order to anticipate useful adaptations correctly. Without doubt, doing so could have major advantages for improving pointing on mobile devices for people experiencing situational impairments.

Pointing in graphical user interfaces is such a fundamental, necessary, and pervasive requirement that until we make it completely accessible to everyone, more work is needed. Perhaps one day we will have a *universal input device* capable of being used easily by anyone capable of making any discernible indication with their mind or body. While initial steps towards such a concept have been made (Carter, Hurst, Mankoff, & Li, 2006; Wang & Mankoff, 2003), we are nowhere near such a breakthrough. It may be that, again, the term "universal" is a hindrance, as the abilities of one user may diametrically oppose the abilities of another. Perhaps what we should seek is an *ability-based input device*, and perhaps having more than just one is a fine, if multifaceted, outcome.

At a higher level, an avenue for future work lies in combining ability-based design, which is focused on human performance, with a complementary design approach focused on human values. *Value-sensitive design* is a tripartite design method that uses conceptual, technical, and empirical investigations into technologies in an effort to understand the values implicitly or explicitly upheld or violated by technology designs (Friedman, Kahn, & Borning, 2006). Values of interest include autonomy, privacy, trust, informed consent, freedom from bias, and others. Among the values most important to people with disabilities is *autonomy*, a sense that they can govern themselves, live independently, and feel empowered to carry out their lives (Bowe, 1988; McCuaig & Frank, 1991). Augmenting ability-based design to consider how people with disabilities perceive the values their technologies embody, or the values using a particular technology makes them *experience*, is a promising avenue for future research.

CONCLUSION

Pointing in graphical user interfaces is part of the bedrock of modern interactive computer use, and yet it remains inaccessible to many people because of the motor-control challenges it presents. People with motor impairments, children, and older users may all struggle to point-and-click with a mouse or tap with a stylus or finger. In this chapter, we have seen technologies that improve pointing-and-clicking for people with motor impairments. We have also seen alternatives that avoid the need for pointing-and-clicking in the first place, such as with goal crossing or barrier pointing. Our ability-based projects have been guided by the principles of ability-based design, and have sought to change fundamental system concepts, properties, and capabilities. In so doing, they have reconsidered the design and use of mouse cursors, target types, interface designs, and input devices. While each approach offers both strengths and weaknesses, and while no single "universal" solution works for all people, these projects comprise strong evidence that target acquisition in graphical user interfaces can be improved for people with motor impairments by taking advantage of what people *can* do, rather than by trying to restore or remedy what people have lost. In its uncompromising stance towards *ability*, ability-based design finds a voice that insists that machines must accommodate people, not vice versa, and that all people must be empowered to express themselves clearly—to machines, and to the world.

ACKNOWLEDGMENT

The author acknowledges the substantial contributions of former and current doctoral and postdoctoral students and fellows: Krzysztof Z. Gajos, Leah Findlater, Shaun K. Kane, Susumu Harada, Alex Jansen, Jon Froehlich, and Mayank Goel. The author also thanks Batya Friedman. This work was supported in part by Microsoft Research and by the U.S. National Science Foundation under grants IIS-0811063, IIS-0939684, and IIS-0952786. Any opinions, findings, conclusions or recommendations expressed in this work are those of the author and do not necessarily reflect those of the U.S. National Science Foundation or any other supporter or collaborator.

REFERENCES

Accot, J., & Zhai, S. (1997). Beyond Fitts' law: Models for trajectory-based HCI tasks. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '97)*, 295-302. New York: ACM Press.

Accot, J., & Zhai, S. (2002). More than dotting the i's: Foundations for crossing-based interfaces. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '02)*, 73-80. New York: ACM Press.

Ahlström, D., Hitz, M., & Leitner, G. (2006). An evaluation of sticky and force enhanced targets in multi target situations. In *Proceedings of the Nordic Conference on Human-Computer Interaction (NordiCHI '06)*, 58-67. New York: ACM Press.

Akamatsu, M., & MacKenzie, I. S. (1996). Movement characteristics using a mouse with tactile and force feedback. *International Journal of Human-Computer Studies*, *45*(4), 483–493. doi:10.1006/ijhc.1996.0063.

Akamatsu, M., & Sato, S. (1994). A multi-modal mouse with tactile and force feedback. *International Journal of Human-Computer Studies*, *40*(3), 443–453. doi:10.1006/ijhc.1994.1020.

Albinsson, P.-A., & Zhai, S. (2003). High precision touch screen interaction. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '03)*, 105-110. New York: ACM Press.

Apitz, G., & Guimbretière, F. (2004). CrossY: A crossing-based drawing application. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST '04),* 3-12. New York: ACM Press.

Appert, C., Chapuis, O., & Beaudouin-Lafon, M. (2008). Evaluation of pointing performance on screen edges. In *Proceedings of the ACM Working Conference on Advanced Visual Interfaces (AVI '08)*, 119-126. New York: ACM Press.

Arnaut, L. Y., & Greenstein, J. S. (1986). Optimizing the touch tablet: The effects of control-display gain and method of cursor control. *Human Factors*, *28*(6), 717–726.

Asano, T., Sharlin, E., Kitamura, Y., Takashima, K., & Kishino, F. (2005). Predictive interaction using the Delphian Desktop. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST '05)*, 133-141. New York: ACM Press.

Ashmore, M., Duchowski, A. T., & Shoemaker, G. (2005). Efficient eye pointing with a fisheye lens. In Proceedings of Graphics Interface (GI '05), 203-210. Toronto:Canadian Information Processing Society.

Balakrishnan, R. (2004). "Beating" Fitts' law: Virtual enhancements for pointing facilitation. *International Journal of Human-Computer Studies*, *61*(6), 857–874. doi:10.1016/j.ijhcs.2004.09.002.

Balakrishnan, R., & MacKenzie, I. S. (1997). Performance differences in the fingers, wrist, and forearm in computer input control. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '97)*303-310. New York: ACM Press.

Bates, R., & Istance, H. (2002). Zooming interfaces! Enhancing the performance of eye controlled pointing devices. In *Proceedings of the ACM SIGCAPH Conference on Assistive Technologies (ASSETS '02)*, 119-126. New York: ACM Press.

Baudisch, P., Cutrell, E., Hinckley, K., & Eversole, A. (2005). Snap-and-go: Helping users align objects without the modality of traditional snapping. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '05)*, 301-310. New York: ACM Press.

Benko, H., Wilson, A. D., & Baudisch, P. (2006). Precise selection techniques for multi-touch screens. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '06)*, 1263-1272. New York: ACM Press.

Bier, E. A., & Stone, M. C. (1986). Snap-dragging. In *Proceedings of the ACM Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '86)*, 233-240. New York: ACM Press.

Bilmes, J. A., Li, X., Malkin, J., Kilanski, K., Wright, R., Kirchhoff, K., et al. (2005). The vocal joystick: A voice-based human-computer interface for individuals with motor impairments. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT/EMNLP '05)*, 995-1002. Stroudsburg, PA: Association for Computational Linguistics.

Bispo, R., & Branco, V. (2009). Designing out stigma–The potential of contradictory symbolic imagery. In *Proceedings of the 5th Internationall Conference on Inclusive Design (Include '09)*. London: Royal College of Art.

Blanch, R., Guiard, Y., & Beaudouin-Lafon, M. (2004). Semantic pointing: Improving target acquisition with control-display ratio adaptation. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '04)*, 519-526. New York: ACM Press.

Bowe, F. (1987). Making computers accessible to disabled people. *MIT's Technology Review*, *90*(1), 52–59, 72.

Bowe, F. (1988). Why seniors don't use technology. *MIT's Technology Review*, *91*(6), 34–40.

Bowe, F. G. (1995). The political and economic issues that drive and derail assistive technology development. *Generations: Journal of the American Society on Aging*, *19*(1), 37–40.

Bragdon, A., Nelson, E., Li, Y., & Hinckley, K. (2011). Experimental analysis of touch-screen gesture designs in mobile environments. In *Proceedings of the ACM Conference in Human Factors in Computing Systems (CHI '11)*, 403-412. New York: ACM Press.

Buxton, W., Hill, R., & Rowley, P. (1985). Issues and techniques in touch-sensitive tablet input. In *Proceedings of the ACM Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '85)*, 215-223. New York: ACM Press.

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

Carter, S., Hurst, A., Mankoff, J., & Li, J. (2006). Dynamically adapting GUIs to diverse input devices. In *Proceedings of the ACM SIGACESS Conference on Computers and Accessibility (ASSETS '06)*, 63-70. New York: ACM Press.

Casali, S. P. (1992). Cursor control device performance by persons with physical disabilities: Implications for hardware and software design. In *Proceedings of the 36th Annual Meeting of the Human Factors Society*, 311-315. Santa Monica, CA: ACM Press.

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

Chang, C.-J., Amick, B. C., Menendez, C. C., Katz, J. N., Johnson, P. W., Robertson, M., & Dennerlein, J. T. (2007). Daily computer usage correlated with undergraduate students' musculoskeletal symptoms. *American Journal of Industrial Medicine*, *50*(6), 481–488. doi:10.1002/ajim.20461 PMID:17450542.

Chapuis, O., Labrune, J.-B., & Pietriga, E. (2009). DynaSpot: Speed-dependent area cursor. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '09)*, 1391-1400. New York: ACM Press.

Cheng, Y. J., Hootman, J. M., Murphy, L. B., Langmaid, G. A., & Helmick, C. G. (2010). Prevalence of doctor-diagnosed arthritis and arthritis-attributable activity limitation—United States, 2007–2009. Morbidity and Mortality Weekly Report, 59(39), 1261-1265. Atlanta, Georgia: Centers for Disease Control and Prevention (CDC).

Chickowski, E. (2004). It's all about access. *Alaska Airlines Magazine, 28*(12), 26-31, 80-82.

Choe, E. K., Shinohara, K., Chilana, P. K., Dixon, M., & Wobbrock, J. O. (2009). Exploring the design of accessible goal crossing desktop widgets. *Extended Abstracts of the ACM Conference on Human Factors in Computing Systems (CHI '09)*, 3733-3738. New York: ACM Press.

Cockburn, A., & Firth, A. (2003). Improving the acquisition of small targets. In *Proceedings of the British HCI Group Annual Conference (HCI '03)*, 181-196. Swindon, England: British Computer Society.

Connell, B. R., Jones, M., Mace, R. L., Mueller, J., Mullick, A., & Vanderheiden, G. et al. (1997). *The principles of universal design, version 2.0*. Raleigh, NC: North Carolina State University.

Cook, A. M., & Hussey, S. M. (2002). *Assistive Technologies: Principles and Practice* (2nd ed.). St. Louis: Mosby.

Crossman, E. R. F. W., & Goodeve, P. J. (1963/1983). Feedback control of hand-movement and Fitts' law. *The Quarterly Journal of Experimental Psychology*, *35*(2), 251–278. PMID:6571310.

Dawe, M. (2005). Caregivers, cost, and complexity: Understanding technology usage by individuals with cognitive disabilities. *Accessibility and Computing: The Newsletter of ACM SIGACCESS*, *81*, 20–23. doi:10.1145/1055674.1055678.

Dawe, M. (2006). Desperately seeking simplicity: How young adults with cognitive disabilities and their families adopt assistive technologies. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '06)*,1143-1152. New York: ACM Press.

Dennerlein, J. T. Y., & Maria, C. (2001). Haptic force-feedback devices for the office computer: Performance and musculoskeletal loading issues. *Human Factors*, *43*(2), 278–286. doi:10.1518/001872001775900850 PMID:11592668.

Dixon, M., Fogarty, J., & Wobbrock, J. O. (2012). A general-purpose target-aware pointing enhancement using pixel-level analysis of graphical interfaces. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '12)*, 3167-3176. New York: ACM Press.

Enders, A. (1995). The role of technology in the lives of older people. *Generations: Journal of the American Society on Aging*, *19*(1), 7–12.

Engel, F. L., Goossens, P., & Haakma, R. (1994). Improved efficiency through I- and E-feedback: A trackball with contextual force feedback. *International Journal of Human-Computer Studies*, *41*(6), 949–974. doi:10.1006/ijhc.1994.1087.

Engelbart, D. C. (1963). A conceptual framework for the augmentation of man's intellect. In Howerton, P. W., & Weeks, D. C. (Eds.), *Vistas in Information Handling*. Washington, D.C.: Spartan Books.

Epps, B. W. (1986). Comparison of six cursor control devices based on Fitts' law models. In *Proceedings of the Human Factors Society 30th Annual Meeting*, 327-331. Santa Monica, CA, Human Factors Society.

Epps, B. W., Snyder, H. L., & Muto, W. H. (1986). Comparison of six cursor devices on a target acquisition task. In *Proceedings of the Society for Information Display*. San Jose, CA: Society for Information Display.

Evans, A., & Wobbrock, J. O. (2011). Input observer: Measuring text entry and pointing performance from naturalistic everyday computer use. *Extended Abstracts of the ACM Conference on Human Factors in Computing Systems (CHI '11)*, 1879-1884. New York: ACM Press.

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

Farris, J. S., Jones, K. S., & Anders, B. A. (2001). Acquisition speed with targets on the edge of the screen: An application of Fitts' law to commonly used web browser controls. In *Proceedings of the Human Factors and Ergonomics Society 45th Annual Meeting (HFES '01)*, 1205-1209. Santa Monica, CA: Human Factors and Ergonomics Society.

Farris, J. S., Jones, K. S., & Anders, B. A. (2002a). Using impenetrable borders in a graphical web browser: Are all angles equal? In *Proceedings of the Human Factors and Ergonomics Society 46th Annual Meeting (HFES '02)*, 1251-1255. Santa Monica, CA: Human Factors and Ergonomics Society.

Farris, J. S., Jones, K. S., & Anders, B. A. (2002b). Using impenetrable borders in a graphical web browser: How does distance influence target selection speed? In *Proceedings of the Human Factors and Ergonomics Society 46th Annual Meeting (HFES '02)*, 1300-1304. Santa Monica, CA: Human Factors and Ergonomics Society.

Feiner, S., Nagy, S., & Van Dam, A. (1981). An integrated system for creating and presenting complex computer-based documents. In *Proceedings of the ACM Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '81)*., 181-189. New York: ACM Press.

Fichten, C. S., Barile, M., Asuncion, J. V., & Fossey, M. E. (2000). What government, agencies, and organizations can do to improve access to computers for postsecondary students with disabilities: Recommendations based on Canadian empirical data. *International Journal of Rehabilitation Research. Internationale Zeitschrift fur Rehabilitationsforschung. Revue Internationale de Recherches de Readaptation*, *23*(3), 191–199. doi:10.1097/00004356-200023030-00009 PMID:11131621.

Findlater, L., Jansen, A., Shinohara, K., Dixon, M., Kamb, P., Rakita, J., & Wobbrock, J. O. (2010). Enhanced area cursors: Reducing fine-pointing demands for people with motor impairments. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST '10)*, 153-162. New York: ACM Press.

Findlater, L., & McGrenere, J. (2004). A comparison of static, adaptive, and adaptable menus. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '04)*, 89-96. New York: ACM Press.

Fitts, P. M. (1954). The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, *47*(6), 381–391. doi:10.1037/h0055392 PMID:13174710.

Fitzmaurice, G., Khan, A., Pieké, R., Buxton, B., & Kurtenbach, G. (2003). Tracking menus. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST '03)*. New York: ACM Press.

Friedman, B., Kahn, P. H., & Borning, A. (2006). Value sensitive design and information systems. In D. Galletta and P. Zhang (eds.), Human-Computer Interaction in Management Information Systems: Foundations (348-372). Armonk, NY: M.E. Sharpe.

Froehlich, J., Wobbrock, J. O., & Kane, S. K. (2007). Barrier Pointing: Using physical edges to assist target acquisition on mobile device touch screens. In *Proceedings of the ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '07)*, 19-26. New York: ACM Press.

Gajos, K., & Weld, D. S. (2004). SUPPLE: Automatically generating user interfaces. In *Proceedings of the ACM Conference on Intelligent User Interfaces (IUI '04)*, 93-100. New York: ACM Press.

Gajos, K., & Weld, D. S. (2005). Preference elicitation for interface optimization. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST '05)*, 173-182. New York: ACM Press.

Gajos, K. Z., Hurst, A., & Findlater, L. (2012). Personalized dynamic accessibility. *interactions*, *19*(2), 69–73. doi:10.1145/2090150.2090167.

Gajos, K. Z., Long, J. J., & Weld, D. S. (2006). Automatically generating custom user interfaces for users with physical disabilities. In *Proceedings of the ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '06)*, 243-244. New York: ACM Press.

Gajos, K. Z., Reinecke, K., & Herrmann, C. (2012). Accurate measurements of pointing performance from *in situ* observations. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '12)*, 3157-3166. New York: ACM Press.

Gajos, K. Z., Weld, D. S., & Wobbrock, J. O. (2010). Automatically generating personalized user interfaces with SUPPLE. *Artificial Intelligence*, *174*(12-13), 910–950. doi:10.1016/j.artint.2010.05.005.

Gajos, K. Z., Wobbrock, J. O., & Weld, D. S. (2007). Automatically generating user interfaces adapted to users' motor and vision capabilities. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST '07)*, 231-240. New York: ACM Press.

Gajos, K. Z., Wobbrock, J. O., & Weld, D. S. (2008). Improving the performance of motor-impaired users with automatically-generated, ability-based interfaces. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '08)*, 1257-1266. New York: ACM Press.

Gitlin, L. N. (1995). Why older people accept or reject assistive technology. *Generations: Journal of the American Society on Aging*, *19*(1), 41–46.

Goel, M., Findlater, L., & Wobbrock, J. O. (2012). WalkType: Using accelerometer data to accommodate situational impairments in mobile touch screen text entry. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI' 12)*, 2687-2696. New York: ACM Press.

Gould, J. D., & Lewis, C. (1985). Designing for usability: Key principles and what designers think. *Communications of the ACM*, *28*(3), 300–311. doi:10.1145/3166.3170.

Grossman, T., & Balakrishnan, R. (2005). The Bubble Cursor: Enhancing target acquisition by dynamic resizing of the cursor's activation area. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '05)*, 281-290. New York: ACM Press.

Grossman, T., Hinckley, K., Baudisch, P., Agrawala, M., & Balakrishnan, R. (2006). Hover Widgets: Using the tracking state to extend the capabilities of pen-operated devices. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '06)*, 861-870. New York: ACM Press.

Guerreiro, T., Nicolau, H., Jorge, J., & Gonçalves, D. (2010a). Towards accessible touch interfaces. In *Proceedings of the ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '10)*, 19-26. New York: ACM Press.

Guerreiro, T., Nicolau, H., Jorge, J., & Gonçalves, D. (2010b). Assessing mobile touch interfaces for tetraplegics. In *Proceedings of the ACM Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '10)*, 31-34. New York: ACM Press.

Guiard, Y., Blanch, R., & Beaudouin-Lafon, M. (2004). Object pointing: A complement to bitmap pointing in GUIs. In *Proceedings of Graphics Interface (GI '04)*, 9-16. Toronto: Canadian Information Processing Society.

Gunawardana, A., Paek, T., & Meek, C. (2010). Usability guided key-target resizing for soft keyboards. In *Proceedings of the ACM Conference on Intelligent User Interfaces (IUI '10)*, 111-118. New York: ACM Press.

Gutwin, C. (2002). Improving focus targeting in interactive fisheye views. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '02)*, 267-274. New York: ACM Press.

Harada, S., Landay, J. A., Malkin, J., Li, X., & Bilmes, J. A. (2006). The Vocal Joystick: Evaluation of voice-based cursor control techniques. In *Proceedings of the ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '06)*, 197-204. New York: ACM Press.

Harada, S., Lester, J., Patel, K., Saponas, T. S., Fogarty, J., Landay, J. A., & Wobbrock, J. O. (2008). VoiceLabel: Using speech to label mobile sensor data. *Proceedings of the ACM Int'l Conference on Multimodal Interfaces (ICMI '08)*, 69-76. New York: ACM Press.

Harada, S., Wobbrock, J. O., & Landay, J. A. (2007). VoiceDraw: A hands-free voice-driven drawing application for people with motor impairments. In *Proceedings of the ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '07)*, 27-34. New York: ACM Press.

Harada, S., Wobbrock, J. O., & Landay, J. A. (2011). Voice Games: Investigation into the use of non-speech voice input for making computer games more accessible. In *Proceedings of the 13th IFIP TC13 Int'l Conference on Human-Computer Interaction (INTERACT '11)*, 11-29. Berlin: Springer-Verlag.

Harada, S., Wobbrock, J. O., Malkin, J., Bilmes, J. A., & Landay, J. A. (2009). Longitudinal study of people learning to use continuous voice-based cursor control. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '09)*, 347-356. New York: ACM Press.

Harper, S. (2007). Is there design for all? *Universal Access in the Information Society*, *6*(1), 111–113. doi:10.1007/s10209-007-0071-2.

Harris, C. M., & Wolpert, D. M. (1998). Signal-dependent noise determines motor planning. *Nature*, *394*(6695), 780–784. doi:10.1038/29528 PMID:9723616.

Hazard, B. L. (2008). Separate but equal? A comparison of content on library web pages and their text versions. *Journal of Web Librarianship*, *2*(2-3), 417–428. doi:10.1080/19322900802191801.

Hinckley, K. (2008). Input technologies and techniques. In A. Sears & J. A. Jacko (eds), The Human-Computer Interaction Handbook (161-176). New York: Lawrence Erlbaum Associates.

Holbert, B., & Huber, M. (2008). Design and evaluation of haptic effects for use in a computer desktop for the physically disabled. In *Proceedings of the 1st Int'l Conference on Pervasive Technologies Related to Assistive Environments (PETRA '08)*, Article No. 9. New York: ACM Press.

Hornof, A. J., & Cavender, A. (2005). EyeDraw: Enabling children with severe motor impairments to draw with their eyes. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '05)*, 161-170. New York: ACM Press.

Hourcade, J. P. (2006). Learning from preschool children's pointing sub-movements. In *Proceedings of the ACM Conference on Interaction Design and Children (IDC '06)*, 65-72. New York: ACM Press.

Hourcade, J. P., Bederson, B. B., Druin, A., & Guimbretière, F. (2004). Differences in pointing task performance between preschool children and adults using mice. *ACM Transactions on Computer-Human Interaction*, *11*(4), 357–386. doi:10.1145/1035575.1035577.

Hourcade, J. P., Nguyen, C. M., Perry, K. B., & Denburg, N. L. (2010). PointAssist for older adults: Analyzing sub-movement characteristics to aid in pointing tasks. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '10)*, 1115-1124. New York: ACM Press.

Hourcade, J. P., Perry, K. B., & Sharma, A. (2008). PointAssist: Helping four year olds point with ease. In *Proceedings of the ACM Conference on Interaction Design and Children (IDC '08)*, 202-209. New York: ACM Press.

Hudson, S. E. (1990). Adaptive semantic snapping-A technique for semantic feedback at the lexical level. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '90)*, 65-70. New York: ACM Press.

Hurst, A., Hudson, S. E., Mankoff, J., & Trewin, S. (2008). Automatically detecting pointing performance. In *Proceedings of the ACM Conference on Intelligent User Interfaces (IUI '08)*, 11-19. New York: ACM Press.

Hurst, A., Mankoff, J., Dey, A. K., & Hudson, S. E. (2007). Dirty desktops: Using a patina of magnetic mouse dust to make common interactor targets easier to select. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST '07)*, 183-186. New York: ACM Press.

Hutchinson, T. E., White, P. K., Martin, W. N., Reichert, K. C., & Frey, L. A. (1989). Human-computer interaction using eye-gaze input. *IEEE Transactions on Systems, Man, and Cybernetics*, *19*(6), 1527–1534. doi:10.1109/21.44068.

Hwang, F., Keates, S., Langdon, P., & Clarkson, P. J. (2003). Multiple haptic targets for motion-impaired computer users. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '03)*, 41-48. New York: ACM Press.

Hwang, F., Keates, S., Langdon, P., & Clarkson, P. J. (2004). Mouse movements of motion-impaired users: A submovement analysis. In *Proceedings of the ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '04)*. New York: ACM Press.

Hwang, F., Keates, S., Langdon, P., Clarkson, P. J., & Robinson, P. (2001). Perception and haptics: Towards more accessible computers for motion-impaired users. In *Proceedings of the ACM Workshop on Percetive User Interfaces (PUI '01)*, 1-9. New York: ACM Press.

Igarashi, T., & Hughes, J. F. (2001). Voice as sound: Using non-verbal voice input for interactive control. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST '01)*, 155-156. New York: ACM Press.

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

Jacob, R. J. K. (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*(2), 152–169. doi:10.1145/123078.128728.

Jacob, R. J. K., & Karn, K. S. (2003). Eye-tracking in human-computer interaction and usability research: Ready to deliver the promises. In J. Hyona, R. Radach, and H. Deubel (eds) The Mind's Eye: Cognitive and Applied Aspects of Eye Movement Research (573-605). Amsterdam: Elsevier.

Jansen, A., Findlater, L., & Wobbrock, J. O. (2011). From the lab to the world: Lessons from extending a pointing technique for real-world use. *Extended Abstracts of the ACM Conference on Human Factors in Computing Systems (CHI '11)*, 1867-1872. New York: ACM Press.

Johnson, B. R., Farris, J. S., & Jones, K. S. (2003). Selection of web browser controls with and without impenetrable borders: Does width make a difference? In *Proceedings of the Human Factors and Ergonomics Society 47th Annual Meeting (HFES '03)*, 1380-1384. Santa Monica, CA: Human Factors and Ergonomics Society.

Johnson, J., Roberts, T. L., Verplank, W., Smith, D. C., Irby, C. H., Beard, M., & Mackey, K. (1989). The Xerox Star: A retrospective. *IEEE Computer, 22*(9), 11-26, 28-29.

Johnson, P. W., Dropkin, J., Hewes, J., & Rempel, D. M. (1993). Office ergonomics: Motion analysis of computer mouse usage. In *Proceedings of the American Industrial Hygiene Conference and Exposition (AIHCE '93)*, 12-13. Falls Church, VA: American Industrial Hygiene Association.

Kabbash, P., & Buxton, W. (1995). The "Prince" technique: Fitts' law and selection using area cursors. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '95)*, 273-279. New York: ACM Press.

Kane, S. K., Wobbrock, J. O., & Smith, I. E. (2008). Getting off the treadmill: Evaluating walking user interfaces for mobile devices in public spaces. In *Proceedings of the ACM Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '08)*, 109-118. Amsterdam: ACM Press.

Karlson, A. K., & Bederson, B. B. (2007). ThumbSpace: Generalized one-handed input for touchscreen-based mobile devices. In *Proceedings of the 11th IFIP TC13 Int'l Conference on Human-Computer Interaction (INTERACT '07)*, 324-338. Berlin: Springer-Verlag.

Karlson, A. K., Bederson, B. B., & SanGiovanni, J. (2005). AppLens and LaunchTile: Two designs for one-handed thumb use on small devices. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '05)*, 201-210. Portland, OR: ACM Press.

Keates, S., & Clarkson, P. J. (1999). Towards a generic approach for designing for all users. In *Proceedings of the RESNA 22nd Annual Conference (RESNA '99)*, 97-99. Arlington, VA: RESNA Press.

Keates, S., & Clarkson, P. J. (2003). Countering design exclusion through inclusive design. In *Proceedings of the ACM Conference on Universal Usability (CUU '03)*, 69-76. Vancouver, BC: ACM Press.

Keates, S., Clarkson, P. J., Harrison, L.-A., & Robinson, P. (2000). Towards a practical inclusive design approach. In *Proceedings of the ACM Conference on Universal Usability (CUU '00)*, 45-52. Arlington, VA: ACM Press.

Keates, S., Clarkson, P. J., & Robinson, P. (2000). Investigating the applicability of user models for motion-impaired users. In *Proceedings of the ACM SIGCAPH Conference on Assistive Technologies (ASSETS '00)*, 129-136. Arlington, VA: ACM Press.

Keates, S., Hwang, F., Langdon, P., Clarkson, P. J., & Robinson, P. (2002). Cursor measures for motion-impaired computer users. In *Proceedings of the ACM SIGCAPH Conference on Assistive Technologies (ASSETS '02)*, 135-142. Edinburgh, Scotland: ACM Press.

Keates, S., Langdon, P., Clarkson, J., & Robinson, P. (2000). Investigating the use of force feedback for motion-impaired users. In *Proceedings of the 6th ERCIM Workshop on User Interfaces for All (UI4All '00)*. Hellas, Crete, Greece: ICS-FORTH.

Keates, S., Langdon, P., Clarkson, P. J., & Robinson, P. (2002). User models and user physical capability. *User Modeling and User-Adapted Interaction*, *12*(2-3), 139–169. doi:10.1023/A:1015047002796.

Ketcham, C. J., Seidler, R. D., Van Gemmert, A. W. A., & Stelmach, G. E. (2002). Age-related kinematic differences as influenced by task difficulty, target size, and movement amplitude. *The Journals of Gerontology. Series B, Psychological Sciences and Social Sciences*, *57B*(1), 54–64. doi:10.1093/geronb/57.1.P54 PMID:11773224.

Keuning, H., Monné, T. K. J., IJsselsteijn, W. A., & Houtsma, A. J. M. (2005). The form of augmented force-feedback fields and the efficiency and satisfaction in computer-aided pointing tasks. *Human Factors*, *47*(2), 418–429. doi:10.1518/0018720054679560 PMID:16170947.

Keyson, D. V. (1997). Dynamic cursor gain and tactual feedback in the capture of cursor movements. *Ergonomics*, *40*(12), 1287–1298. doi:10.1080/001401397187379 PMID:9416013.

Koester, H. H. (2003). Abandonment of speech recognition by new users. In *Proceedings of the RESNA 26th Annual Conference (RESNA '03)*. Arlington, VA: RESNA Press.

Koester, H. H., LoPresti, E., & Simpson, R. C. (2005). Toward Goldilocks' pointing device: Determining a "just right" gain setting for users with physical impairments. In *Proceedings of the ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '05)*, 84-89. New York: ACM Press.

Koester, H. H., LoPresti, E., & Simpson, R. C. (2007). Toward automatic adjustment of keyboard settings for people with physical impairments. *Disability and Rehabilitation. Assistive Technology*, *2*(5), 261–274. doi:10.1080/17483100701284265 PMID:19263532.

Kumar, M., Paepcke, A., & Winograd, T. (2007). EyePoint: Practical pointing and selection using gaze and keyboard. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '07)*, 421-430. New York: ACM Press.

Lane, D. M., Peres, S. C., Sandor, A., & Napier, H. A. (2005). A process for anticipating and executing icon selection in graphical user interfaces. *International Journal of Human-Computer Interaction*, *19*(2), 241–252. doi:10.1207/s15327590ijhc1902_5.

Langdon, P., Hwang, F., Keates, S., Clarkson, P. J., & Robinson, P. (2002a). Investigating haptic assistive interfaces for motion-impaired users: Force-channels and competitive attractive-basins. In *Proceedings of EuroHaptics 2002*, 122-127. Paris, France: EuroHaptics Society.

Langdon, P., Keates, S., Clarkson, P. J., & Robinson, P. (2000). Using haptic feedback to enhance computer interaction for motion-impaired users. In *Proceedings of the 3rd International Conference on Disability, Virtual Reality and Associated Technologies (ICDVRAT '00)*, 25-32. Reading, England: University of Reading.

Langdon, P. M., Hwang, F., Keates, S., Clarkson, P. J., & Robinson, P. (2002b). Developing assistive interfaces for motion-impaired users using cursor movement analysis in conjunction with haptic feedback. In *Proceedings of the 4th Int'l Conference on Disability, Virtual Reality and Associated Technologies (ICDVRAT '02)*, 223-230. Reading, England: University of Reading.

Lank, E., Cheng, Y.-C. N., & Ruiz, J. (2007). Endpoint prediction using motion kinematics. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '07)*, 637-646. New York: ACM Press.

Lankford, C. (2000). Effective eye-gaze input into Windows. In *Proceedings of the ACM Symposium on Eye Tracking Research and Applications (ETRA '00)*, 23-27. New York: ACM Press.

LaPlante, M. P., Hendershot, G. E., & Moss, A. J. (1992). *Assistive technology devices and home accessibility features: Prevalence, payment, need, and trends. Advance Data; no. 217. U.S. Department of Health and Human Services*. Hyattsville, MD: National Center for Health Statistics.

Lécuyer, A., Burkhardt, J.-M., & Etienne, L. (2004). Feeling bumps and holes without a haptic interface: The perception of pseudo-haptic textures. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '04)*, 83-90. New York: ACM Press.

Lécuyer, A., Coquillart, S., Kheddar, A., Richard, P., & Coiffet, P. (2000). Can isometric input devices simulate force feedback? In *Proceedings of the IEEE Conference on Virtual Reality (VR '00)*. Washington, D.C.: IEEE Computer Society.

Lee, S., & Zhai, S. (2009). The performance of touch screen soft buttons. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '09)*, 309-318. New York: ACM Press.

Lin, M., Goldman, R., Price, K. J., Sears, A., & Jacko, J. (2007). How do people tap when walking? An empirical investigation of nomadic data entry. *International Journal of Human-Computer Studies*, *65*(9), 759–769. doi:10.1016/j.ijhcs.2007.04.001.

Mace, R. L., Hardie, G. J., & Place, J. P. (1991). Accesible environments: Toward universal design. In W. E. Preiser, J. C. Vischer, & E. T. White (eds), Design Intervention: Toward a More Humane Architecture (155-176). New York: Van Nostrand Reinhold.

MacKay, B., Dearman, D., Inkpen, K., & Watters, C. (2005). Walk 'n scroll: A comparison of software-based navigation techniques for different levels of mobility. In *Proceedings of the ACM Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '05)*, 183-190. New York: ACM Press.

MacKenzie, I. S. (1992). Fitts' law as a research and design tool in human-computer interaction. *Human-Computer Interaction*, *7*(1), 91–139. doi:10.1207/s15327051hci0701_3.

MacKenzie, I. S., & Isokoski, P. (2008). Fitts' throughput and the speed-accuracy tradeoff. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '08)*, 1633-1636. New York: ACM Press.

MacKenzie, I. S., Kauppinen, T., & Silfverberg, M. (2001). Accuracy measures for evaluating computer pointing devices. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '01)*, 9-16. New York: ACM Press.

MacKenzie, I. S., & Riddersma, S. (1994). Effects of output display and control-display gain on human performance in interactive systems. *Behaviour & Information Technology*, *13*(5), 328–337. doi:10.1080/01449299408914613.

MacKenzie, I. S., Sellen, A., & Buxton, W. (1991). A comparison of input devices in elemental pointing and dragging tasks. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '91)*, 161-166. New York: ACM Press.

Majaranta, P., & Räihä, K.-J. (2002). Twenty years of eye typing: Systems and design issues. In *Proceedings of the ACM Symposium on Eye Tracking Research and Applications (ETRA '02)*, 15-22. New York: ACM Press.

Mandryk, R. L., & Gutwin, C. (2008). Perceptibility and utility of sticky targets. In *Proceedings of Graphics Interface (GI '08)*, 65-72. Toronto: Canadian Information Processing Society.

Mankoff, J., Hudson, S. E., & Abowd, G. D. (2000). Interaction techniques for ambiguity resolution in recognition-based interfaces. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST '00)*, 11-20. New York: ACM Press.

Markoff, J. (1983). Microsoft does Windows. *InfoWorld*, *5*(47), 32–36.

McCuaig, M., & Frank, G. (1991). The able self: Adaptive patterns and choices in independent living for a person with cerebral palsy. *The American Journal of Occupational Therapy.*, *43*(3), 224–234. doi:10.5014/ajot.45.3.224 PMID:1827706.

McGuffin, M., & Balakrishnan, R. (2002). Acquisition of expanding targets. In*Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '02)*. New York: ACM Press.

McGuffin, M. J., & Balakrishnan, R. (2005). Fitts' law and expanding targets: Experimental studies and designs for user interfaces. *ACM Transactions on Computer-Human Interaction*, *12*(4), 388–422. doi:10.1145/1121112.1121115.

Meiselwitz, G., Wentz, B., & Lazar, J. (2009). Universal Usability: Past, present, and future. *Foundations and Trends in Human-Computer Interaction*, *3*(4), 213–333. doi:10.1561/1100000029.

Mertens, A., Jochems, N., Schlick, C. M., Dünnebacke, D., & Dornberg, J. H. (2010). Design pattern TRABING: Touchscreen-based input technique for people affected by intention tremor. In *Proceedings of the ACM Symposium on Engineering Interactive Computing Systems (EICS '10)*. New York: ACM Press.

Meyer, D. E., Abrams, R. A., Kornblum, S., Wright, C. E., & Smith, J. E. K. (1988). Optimality in human motor performance: Ideal control of rapid aimed movements. *Psychological Review*, *95*(3), 340–370. doi:10.1037/0033-295X.95.3.340 PMID:3406245.

Meyer, D. E., Smith, J. E. K., Kornblum, S., Abrams, R. A., & Wright, C. E. (1990). Speed-accuracy tradeoffs in aimed movements: Toward a theory of rapid voluntary action. In M. Jeannerod (ed), Attention and Performance XIII (173-226). Hillsdale, NJ: Lawrence Erlbaum.

Mikkelsen, S., Vilstrup, I., Lassen, C. F., Kryger, A. I., Thomsen, J. F., & Andersen, J. H. (2007). Validity of questionnaire self-reports on computer, mouse and keyboard usage during a four-week period. *Occupational and Environmental Medicine*, *64*(8), 541–547. doi:10.1136/oem.2005.026351 PMID:17387136.

Miniotas, D., Špakov, O., & MacKenzie, I. S. (2004). Eye gaze interaction with expanding targets. *Extended Abstracts of the ACM Conference on Human Factors in Computing Systems (CHI '04)*. New York: ACM Press.

Mitchell, J., & Shneiderman, B. (1989). Dynamic versus static menus: An exploratory comparison. *SIGCHI Bulletin*, *20*(4), 33–37. doi:10.1145/67243.67247.

Moffatt, K., & McGrenere, J. (2010). Steadied-bubbles: Combining techniques to address pen-based pointing errors for younger and older adults. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '10)*, 1125-1134. New York: ACM Press.

Montague, K., Hanson, V. L., & Cobley, A. (2012). Designing for individuals: Usable touch-screen interaction through shared user models. In *Proceedings of the ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '12)*, 151-158. New York: ACM Press.

Muller, M. J., Wharton, C., McIver, W. J., & Laux, L. (1997). Toward an HCI research and practice agenda based on human needs and social responsibility. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '97)*, 155-161. New York: ACM Press.

Münch, S., & Dillmann, R. (1997). Haptic output in multimodal user interfaces. In *Proceedings of the ACM Conference on Intelligent User Interfaces (IUI '97)*, 105-112. New York: ACM Press.

Murata, A. (1998). Improvement of pointing time by predicting targets in pointing with a PC mouse. *International Journal of Human-Computer Interaction*, *10*(1), 23–32. doi:10.1207/s15327590ijhc1001_2.

Newell, A. F. (1995). Extra-ordinary human-computer interaction. In A. D. N. Edwards (ed), Extra-Ordinary Human-Computer Interaction: Interfaces for Users with Disabilities (3-18). Cambridge, UK: Cambridge University Press.

Newell, A. F., & Gregor, P. (2000). "User sensitive inclusive design"—in search of a new paradigm. In *Proceedings of the ACM Conference on Universal Usability (CUU '00)*, 39-44. New York: ACM Press.

Nuance Communications. (2011). *Moving the mouse pointer and clicking the mouse. Dragon User Guide, Version 11.5*, 63-68. Burlington, MA: Nuance Communications, Inc..

Odell, J., & Mukerjee, K. (2007). Architecture, user interface, and enabling technology in Windows Vista's speech systems. *IEEE Transactions on Computers*, *56*(9), 1156–1168. doi:10.1109/TC.2007.1065.

Olwal, A., Feiner, S., & Heyman, S. (2008). Rubbing and tapping for precise and rapid selection on touch-screen displays. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '08)*, 295-304. New York: ACM Press.

Parhi, P., Karlson, A. K., & Bederson, B. B. (2006). Target size study for one-handed thumb use on small touchscreen devices. In *Proceedings of the ACM Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '06)*, 203-210. New York: ACM Press.

Perry, K. B., & Hourcade, J. P. (2008). Evaluating one handed thumb tapping on mobile touchscreen devices. *Proceedings of Graphics Interface (GI '08)*, 57-64. Toronto: Canadian Information Processing Society.

Phillips, B., & Zhao, H. (1993). Predictors of assistive technology abandonment. *Assistive Technology*, *5*(1), 36–45. doi:10.1080/10400435.1993.10132205 PMID:10171664.

Potter, R. L., Weldon, L. J., & Shneiderman, B. (1988). Improving the accuracy of touch screens: An experimental evaluation of three strategies. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '88)*, 27-32. New York: ACM Press.

Pugliese, P. J., & Gould, J. M. (1998). Voice controlled cursor movement. *United States patent #5,818,423.* Washington DC: U.S. Patent Office.

Pullin, G., & Newell, A. (2007). Focussing on extra-ordinary users. In *Proceedings of the 4th Int'l Conference on Universal Access in Human-Computer Interaction (UAHCI '07)*, 253-262. Berlin: Springer-Verlag.

Ramos, G., Cockburn, A., Balakrishnan, R., & Beaudouin-Lafon, M. (2007). Pointing lenses: Facilitating stylus input through visual-and motor-space magnification. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '07)*, 757-766. New York: ACM Press.

Riemer-Reiss, M. L., & Wacker, R. R. (2000). Factors associated with assistive technology discontinuance among individuals with disabilities. *Journal of Rehabilitation*, *66*(3), 44–50.

Ringbauer, B., Peissner, M., & Gemou, M. (2007). From "design for all" towards "design for one"–A modular user interface approach. In *Proceedings of the 4th International Conference on Universal Access in Human-Computer Interaction (UAHCI '07)*, 517-526. Berlin: Springer-Verlag.

Riviere, C. N., & Thakor, N. V. (1996). Effects of age and disability on tracking tasks with a computer mouse: Accuracy and linearity. *Journal of Rehabilitation Research and Development*, *33*(1), 6–15. PMID:8868412.

Rodgers, M. E., Mandryk, R. L., & Inkpen, K. M. (2006). Smart sticky widgets: Pseudo-haptic enhancements for multi-monitor displays. In *Proceedings of the 6th International Symposium on Smart Graphics (SG '06)*, 194-205. Berlin: Springer-Verlag.

Roth, V., & Turner, T. (2009). Bezel Swipe: Conflict-free scrolling and multiple selection on mobile touch screen devices. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '09)*, 1523-1526. New York: ACM Press.

Salivia, G., & Hourcade, J. P. (2013). PointAssist: Assisting individuals with motor impairments. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '13)*, 1213-1222. New York: ACM Press.

Salvucci, D. D., & Anderson, J. R. (2000). Intelligent gaze-added interfaces. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '00)*, 273-280. New York: ACM Press.

Sears, A., Lin, M., Jacko, J., & Xiao, Y. (2003). When computers fade... Pervasive computing and situationally-induced impairments and disabilities. In *Proceedings of the 10th Int'l Conference on Human-Computer Interaction (HCI International '03)*, 1298-1302. Mahwah, NJ: Lawrence Erlbaum Associates.

Sears, A., & Shneiderman, B. (1991). High precision touchscreens: Design strategies and comparisons with a mouse. *International Journal of Man-Machine Studies*, *34*(4), 593–613. doi:10.1016/0020-7373(91)90037-8.

Sears, A., Young, M., & Feng, J. (2008). Physical disabilities and computing technologies: An analysis of impairments. In J. A. Jacko and A. Sears (eds), The Human-Computer Interaction Handbook (829-852). New York: Lawrence Erlbaum Associates.

Shneiderman, B. (2000). Universal usability. *Communications of the ACM*, *43*(5), 84–91. doi:10.1145/332833.332843.

Soukoreff, R. W., & MacKenzie, I. S. (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. doi:10.1016/j.ijhcs.2004.09.001.

Sporka, A. J., Kurniawan, S. H., Mahmud, M., & Slavík, P. (2006). Non-speech input and speech recognition for real-time control of computer games. In *Proceedings of the ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '06)*, 213-220. New York: ACM Press.

Sporka, A. J., Kurniawan, S. H., & Slavík, P. (2004). Whistling user interface (U^3I). In *Proceedings of the 8th ERCIM Workshop on User Interfaces for All (UI4All '04)*, 472-478. Berlin: Springer-Verlag.

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

Stary, C. (1997). The role of design and evaluation principles for user interfaces for all. In *Proceedings of the 7th Int'l Conference on Human-Computer Interaction (HCI Int'l '97)*. The Netherlands: Elsevier Science.

Steinfeld, E. (1994). The concept of universal design. In *Proceedings of the Sixth Ibero-American Conference on Accessibility*. Rio de Janeiro: National Academy Press.

Stephanidis, C. (2001). *User Interfaces for All: Concepts, Methods, and Tools*. Mahwah, NJ: Lawrence Erlbaum.

Story, M. F. (1998). Maximizing usability: The principles of universal design. *Assistive Technology*, *10*(1), 4–12. doi:10.1080/10400435.1998.10131955 PMID:10181150.

Sutherland, I. E. (1963). Sketchpad: A man-machine graphical communication system. In *Proceedings of the AFIPS Spring Joint Computer Conference, Volume 23*, 329-346. Santa Monica, CA, MI: AFIPS Press.

Sutter, C., & Ziefle, M. (2005). Psychomotor user characteristics as predictors for a successful usage of small input devices. In *Proceedings of the 11th International Conference on Human-Computer Interaction (HCI Int'l '05)*. Mahwah, NJ: Lawrence Erlbaum Associates.

Trewin, S. (1996). A study of input device manipulation difficulties. In *Proceedings of the ACM SIGCAPH Conference on Assistive Technologies (ASSETS '96)*. New York: ACM Press.

Trewin, S., Keates, S., & Moffatt, K. (2006). Developing Steady Clicks: A method of cursor assistance for people with motor impairments. In *Proceedings of the ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '06)*, 26-33. New York: ACM Press.

Trewin, S., & Pain, H. (1997). Dynamic modelling of keyboard skills: Supporting users with motor disabilities. In *Proceedings of the 6th International Conference on User Modeling (UM '97)*, 135-146. Vienna, NY: Springer.

Trewin, S., & Pain, H. (1999). Keyboard and mouse errors due to motor disabilities. *International Journal of Human-Computer Studies*, *50*(2), 109–144. doi:10.1006/ijhc.1998.0238.

Vance, A. (2009, Spetember, 15). Insurers fight speech-impairment remedy. *The New York Times*, (pp. A1). Retrieved from http://www.nytimes.com.

Vanderheiden, G. (1998). Universal design and assistive technology in communication and information technologies: Alternatives or complements? *Assistive Technology*, *10*(1), 29–36. doi:10.1080/10400435.1998.10131958 PMID:10181148.

Vanderheiden, G. (2000). Fundamental principles and priority setting for universal usability. In *Proceedings of the ACM Conference on Universal Usability (CUU '00)*, 32-37. New York: ACM Press.

Vogel, D., & Baudisch, P. (2007). Shift: A technique for operating pen-based interfaces using touch. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '07)*, 657-666. New York: ACM Press.

Wacharamanotham, C., Hurtmanns, J., Mertens, A., Kronenbuerger, M., Schlick, C., & Borchers, J. (2011). Evaluating swabbing: A touchscreen input method for elderly users with tremor. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '11)*, 623-626. New York: ACM Press.

Walker, N., Philbin, D. A., & Fisk, A. D. (1997). Age-related differences in movement control: Adjusting submovement structure to optimize performance. *The Journals of Gerontology. Series B, Psychological Sciences and Social Sciences*, *52B*(1), 40–52. doi:10.1093/geronb/52B.1.P40 PMID:9008674.

Walker, N., & Smelcer, J. B. (1990). A comparison of selection time from walking and bar menus. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '90)*, 221-225. New York: ACM Press.

Wang, J., & Mankoff, J. (2003). Theoretical and architectural support for input device adaptation. In *Proceedings of the ACM Conference on Universal Usability (CUU '03)*, 85-92. New York: ACM Press.

Weir, D., Rogers, S., Murray-Smith, R., & Löchtefeld, M. (2012). A user-specific machine learning approach for improving touch accuracy on mobile devices. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST '12)*, 465-475. New York: ACM Press.

Williams, G. (1984). The Apple Macintosh computer. *Byte*, *9*(2), 30–54.

Wobbrock, J. O. (2003). The benefits of physical edges in gesture-making: Empirical support for an edge-based unistroke alphabet. *Extended Abstracts of the ACM Conference on Human Factors in Computing Systems (CHI '03)*, 942-943. New York: ACM Press.

Wobbrock, J. O., Fogarty, J., Liu, S.-Y., Kimuro, S., & Harada, S. (2009). The Angle Mouse: Target-agnostic dynamic gain adjustment based on angular deviation. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '09)*, 1401-1410. New York: ACM Press.

Wobbrock, J. O., & Gajos, K. Z. (2007). A comparison of area pointing and goal crossing for people with and without motor impairments. In *Proceedings of the ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '07)*, 3-10. New York: ACM Press.

Wobbrock, J. O., & Gajos, K. Z. (2008). Goal crossing with mice and trackballs for people with motor impairments: Performance, submovements, and design directions. *ACM Transactions on Accessible Computing*, *1*(1), 4:1-4:37. doi:10.1145/1361203.1361207.

Wobbrock, J. O., Kane, S. K., Gajos, K. Z., Harada, S., & Froehlich, J. (2011a). Ability-based design: Concept, principles, and examples. *ACM Transactions on Accessible Computing*, *3*(3), 9:1-9:27. doi:10.1145/1952383.1952384.

Wobbrock, J. O., & Myers, B. A. (2006). Trackball text entry for people with motor impairments. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '06)*, 479-488. New York: ACM Press.

Wobbrock, J. O., Myers, B. A., & Aung, H. H. (2004). Writing with a joystick: A comparison of date stamp, selection keyboard, and EdgeWrite. In *Proceedings of Graphics Interface (GI '04)*, 1-8. Toronto: Canadian Information Processing Society.

Wobbrock, J. O., Myers, B. A., Aung, H. H., & LoPresti, E. F. (2004). Text entry from power wheelchairs: EdgeWrite for joysticks and touchpads. In *Proceedings of the ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '04)*, 110-117. New York: ACM Press.

Wobbrock, J. O., Myers, B. A., & Kembel, J. A. (2003). EdgeWrite: A stylus-based text entry method designed for high accuracy and stability of motion. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST '03)*, 61-70. New York: ACM Press.

Wobbrock, J. O., Shinohara, K., & Jansen, A. (2011b). The effects of task dimensionality, endpoint deviation, throughput calculation, and experiment design on pointing measures and models. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '11)*, 1639-1648. New York: ACM Press.

Worden, A., Walker, N., Bharat, K., & Hudson, S. E. (1997). Making computers easier for older adults to use: Area cursors and sticky icons. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '97)*, 266-271. New York: ACM Press.

Yamato, M., Monden, A., Matsumoto, K.-i., Inoue, K., & Torii, K. (2000). Button selection for general GUIs using eye and hand together. In *Proceedings of the ACM Working Conference on Advanced Visual Interfaces (AVI '00)*, 270-273. New York: ACM Press.

Zhai, S. (2004). Characterizing computer input with Fitts' law parameters—the information and non-information aspects of pointing. *International Journal of Human-Computer Studies*, *61*(6), 791–809. doi:10.1016/j.ijhcs.2004.09.006.

Zhai, S., Conversy, S., Beaudouin-Lafon, M., & Guiard, Y. (2003). Human on-line response to target expansion. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '03)*, 177-184. New York: ACM Press.

Zhai, S., Morimoto, C., & Ihde, S. (1999). Manual and gaze input cascaded (MAGIC) pointing. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '99)*, 246-253. New York: ACM Press.

Ziebart, B. D., Dey, A. K., & Bagnell, J. A. (2012). Probabilistic pointing target prediction via inverse optimal control. In *Proceedings of the ACM Conference on Intelligent User Interfaces (IUI '12)*, 1-10. New York: ACM Press.

ADDITIONAL READING

Balakrishnan, R. (2004). "Beating" Fitts' law: Virtual enhancements for pointing facilitation. *International Journal of Human-Computer Studies*, *61*(6), 857–874. doi:10.1016/j.ijhcs.2004.09.002.

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

Findlater, L., Jansen, A., Shinohara, K., Dixon, M., Kamb, P., Rakita, J., & Wobbrock, J. O. (2010). Enhanced area cursors: Reducing fine-pointing demands for people with motor impairments. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST '10),* 153-162. New York: ACM Press.

Gajos, K. Z., Weld, D. S., & Wobbrock, J. O. (2010). Automatically generating personalized user interfaces with Supple. *Artificial Intelligence*, *174*(12-13), 910–950. doi:10.1016/j.artint.2010.05.005.

Gajos, K. Z., Wobbrock, J. O., & Weld, D. S. (2007). Automatically generating user interfaces adapted to users' motor and vision capabilities. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST '07),* 231-240. New York: ACM Press.

Gajos, K. Z., Wobbrock, J. O., & Weld, D. S. (2008). Improving the performance of motor-impaired users with automatically-generated, ability-based interfaces. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '08),* 1257-1266. New York: ACM Press.

Harper, S. (2007). Is there design for all? *Universal Access in the Information Society*, *6*(1), 111–113. doi:10.1007/s10209-007-0071-2.

Hourcade, J. P., Perry, K. B., & Sharma, A. (2008). PointAssist: Helping four year olds point with ease. In *Proceedings of the ACM Conference on Interaction Design and Children (IDC '08)*, 202-209. New York: ACM Press.

Keates, S., Clarkson, P. J., Harrison, L.-A., & Robinson, P. (2000). Towards a practical inclusive design approach. In *Proceedings of the ACM Conference on Universal Usability (CUU '00),* 45-52. New York: ACM Press.

Keates, S., Langdon, P., Clarkson, P. J., & Robinson, P. (2002). User models and user physical capability. *User Modeling and User-Adapted Interaction*, *12*(2-3), 139–169. doi:10.1023/A:1015047002796.

Kondraske, G. V. (1988). Rehabilitation engineering: Towards a systematic process. *IEEE Engineering in Medicine and Biology Magazine*, *7*(3), 11–15. doi:10.1109/51.7928 PMID:18244068.

Kondraske, G. V. (1990). A PC-based performance measurement laboratory system. *Journal of Clinical Engineering*, *15*(6), 467–478.

Lin, M., Goldman, R., Price, K. J., Sears, A., & Jacko, J. (2007). How do people tap when walking? An empirical investigation of nomadic data entry. *International Journal of Human-Computer Studies*, *65*(9), 759–769. doi:10.1016/j.ijhcs.2007.04.001.

Mace, R. L., Hardie, G. J., & Place, J. P. (1991). Accesible environments: Toward universal design. In. W. E. Preiser, J. C. Vischer, and E. T. White (eds), Design Intervention: Toward a More Humane Architecture (155-176). New York: Van Nostrand Reinhold.

Newell, A. F. (1995). Extra-ordinary human-computer interaction. In A. D. N. Edwards (ed) Extra-Ordinary Human-Computer Interaction: Interfaces for Users with Disabilities (3-18). Cambridge, UK: Cambridge University Press.

Ringbauer, B., Peissner, M., & Gemou, M. (2007). From "design for all" towards "design for one" – A modular user interface approach. In *Proceedings of the 4th Int'l Conference on Universal Access in Human-Computer Interaction (UAHCI '07)*, 517-526. Berlin: Springer-Verlag.

Sears, A., Lin, M., Jacko, J., & Xiao, Y. (2003). When computers fade… Pervasive computing and situationally-induced impairments and disabilities. In *Proceedings of the 10th Int'l Conference on Human-Computer Interaction (HCI Int'l '03)*, 1298-1302. Mahwah, NJ: Lawrence Erlbaum Associates.

Shneiderman, B. (2000). Universal usability. *Communications of the ACM*, *43*(5), 84–91. doi:10.1145/332833.332843.

Soukoreff, R. W., & MacKenzie, I. S. (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. doi:10.1016/j.ijhcs.2004.09.001.

Stephanidis, C. (2001). Adaptive techniques for universal access. *User Modeling and User-Adapted Interaction*, *11*(1-2), 159–179. doi:10.1023/A:1011144232235.

Story, M. F. (1998). Maximizing usability: The principles of universal design. *Assistive Technology*, *10*(1), 4–12. doi:10.1080/10400435.1998.10131955 PMID:10181150.

Vanderheiden, G. (1998). Universal design and assistive technology in communication and information technologies: Alternatives or complements? *Assistive Technology*, *10*(1), 29–36. doi:10.1080/10400435.1998.10131958 PMID:10181148.

Wobbrock, J. O., Fogarty, J., Liu, S.-Y., Kimuro, S., & Harada, S. (2009). The Angle Mouse: Target-agnostic dynamic gain adjustment based on angular deviation. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '09),* 1401-1410. New York: ACM Press.

Wobbrock, J. O., Kane, S. K., Gajos, K. Z., Harada, S., & Froehlich, J. (2011). Ability-based design: Concept, principles, and examples. *ACM Transactions on Accessible Computing*, *3*(3), 1–27. doi:10.1145/1952383.1952384.

Worden, A., Walker, N., Bharat, K., & Hudson, S. E. (1997). Making computers easier for older adults to use: Area cursors and sticky icons. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '97),* 266-271. New York: ACM Press.

ENDNOTES

1. The projects described herein resulted from extensive collaborations with current and former doctoral students and postdoctoral fellows: Krzysztof Z. Gajos, Leah Findlater, Shaun K. Kane, Susumu Harada, Alex Jansen, Jon Froehlich, and Mayank Goel. These researchers deserve much of the credit for the projects described.
2. Available at http://www.w3.org/TR/UNDERSTANDING-WCAG20/intro.html#introduction-fourprincs-head. Accessed December 14, 2012.
3. For example, the Windows operating system exposes this setting on the mouse control panel, where users have a slider that produces values of 1 (low) to 20 (high). The default setting is 10. This slider value is *not* the same thing as a control-display gain of 10, but corresponds to a gain of about 5 (Wobbrock, Fogarty, Liu, Kimuro & Harada, 2009).

[4] If targets were to expand in visual-space *only*, they would pose no benefit whatsoever to pointing performance because motor-space would remain unchanged (Gutwin, 2002). Conventional "split-screen" screen magnifiers are visual-space-only magnifiers, and while they are useful to people with low vision, they provide no benefit to pointing performance.

[5] Throughput is a combined speed-accuracy measure of pointing efficiency and is the primary descriptive measure produced by Fitts' law (Fitts, 1954; MacKenzie, 1992; MacKenzie & Isokoski, 2008; Soukoreff & MacKenzie, 2004; Wobbrock, Shinohara & Jansen, 2011b; Zhai, 2004).

[6] The Angle Mouse software is available on its project page at http://depts.washington.edu/aimgroup/proj/angle/. Accessed December 14, 2012.

[7] Technically, the study involved the use of the Visual-Motor-Magnifier, the laboratory design preceding the released Pointing Magnifier. The designs are the same but for an inset bubble cursor within the lens of the former, a mechanism requiring target-awareness and omitted from the final Pointing Magnifier, which was target-agnostic. While the inset bubble cursor undoubtedly aids the technique somewhat, it is not expected that the findings would change for the Pointing Magnifier given the considerable benefits of magnification.

[8] The Pointing Magnifier software is available on its project page at http://depts.washington.edu/aimgroup/proj/ptgmag/. Accessed December 14, 2012.

[9] The Vocal Joystick software is available on its project page at http://ssli.ee.washington.edu/vj/. Accessed December14, 2012.