Input Technologies and Techniques

Ken Hinckley

Microsoft Research

One Microsoft Way

Redmond, WA 98052

kenh@microsoft.com

Tel: (425) 703-9065

DRAFT – subject to editorial changes.

Revision of 2002 chapter with lots of new material.

K. Hinckley, *Input Technologies and Techniques*, in Handbook of Human-Computer Interaction, ed. by A. Sears & J. Jacko.

CONTENTS

10 References

1 Introduction: What's an input device anyway?

Input devices sense physical properties of people, places, or things. Yet any treatment of input devices without regard to the corresponding visual feedback is like trying to use a pen without paper. Small-screen devices with integrated sensors underscore the indivisibility of input and output. This chapter treats input technologies at the level of *interaction techniques*, which provide a way for users to accomplish tasks by combining input with appropriate feedback. An interaction designer must consider the physical sensor, the feedback presented to the user, the ergonomic and industrial design of the device, and the interplay between all of the interaction techniques supported by a system.

This chapter enumerates properties of input devices and provides examples of how these properties apply to common pointing devices as well as mobile devices with touch or pen input. We will discuss how to use input signals in applications, and cover models and theories that help to evaluate interaction techniques and reason about design options. We will also discuss discrete symbolic entry, including mobile and keyboard-based text entry. The chapter concludes with some thoughts about future trends.

2 Understanding Input Technologies

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

2.1 Input Device Properties

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

Property Sensed: Most devices sense linear position, motion, or force; rotary devices sense angle, change in angle, and torque (Buxton, 1995c; Card, Mackinlay & Robertson, 1991). For example, tablets sense position of a pen, mice sense motion (change in position), and isometric joysticks sense force. The property sensed determines the mapping from input to output, or *transfer function*, that is most appropriate for the device (see Section 5). Position sensing devices are *absolute input devices*, whereas motion sensing devices are *relative input devices*. A relative device, such as the mouse, requires visual feedback in the form of a cursor to indicate a screen location. With absolute devices, the *nulling problem* (Buxton, 1983) arises if the position of a physical intermediary, such as a slider on a mixing console, is not in agreement with a value set in software. This problem cannot occur with relative devices, but users may waste time *clutching*: the user must occasionally lift a mouse to reposition it.

Number of Dimensions: Devices sense one or more input dimensions. For example, a mouse senses two linear dimensions of motion, a knob senses one angular dimension, and a six degree-of-freedom magnetic tracker measures three position dimensions and three orientation dimensions. A pair of knobs or a mouse with a scroll wheel sense separate input dimensions and thus form a "1D+1D" device, or a "2D+1D" *multi-channel device*, respectively (Zhai, Smith & Selker, 1997). *Multi-degree-of-freedom devices (3D input devices)* sense three or more simultaneous dimensions of spatial position or orientation (Bowman, Kruijff, LaViola & Poupyrev, 2004; Hinckley, Pausch, Goble & Kassell, 1994; Hinckley, Sinclair, Hanson, Szeliski & Conway, 1999; Zhai, 1998).

Indirect vs. Direct: A mouse is an indirect input device because the user must move the mouse to indicate a point on the screen, whereas a direct input device has a unified input and display surface. Direct devices such as touchscreens, or display tablets operated with a pen, are not necessarily easier to use than indirect devices. Direct devices lack buttons for state transitions. Occlusion is also a major design challenge. The finger or pen occludes the area a user is pointing at, so a user may not realize that they have activated a control; occlusion by the hand and arm also may cause the user to overlook pop-up menus, dialogs, or status indicators.

Device acquisition time: The average time to move one's hand to a device is known as *acquisition time*. *Homing time* is the time to return from a device to a "home" position (e.g. return from mouse to keyboard). For common desktop workflows that involve switching between text entry and pointing, the effectiveness of a device for pointing tends to dominate acquisition and homing time costs (Douglas & Mithal, 1994). Thus, integration of a pointing device with the keyboard may not improve overall performance, but evaluations still must assess any influence of acquisition times (Dillon, Eday & Tombaugh, 1990; Hinckley, Guimbretiere, Baudisch, Sarin, Agrawala & Cutrell, 2006).

Gain: Also known as *control-to-display (C:D) gain* or *C:D ratio*, gain is defined as the distance moved by an input device divided by the distance moved on the display. Gain confounds what should be two measurements, device size and display size, with one arbitrary metric (Accot & Zhai, 2001; MacKenzie, 1995), and is therefore suspect as a factor for experimental study. In experiments, gain typically has little or no effect on the time to perform pointing movements, but variable gain functions may provide some benefit by reducing the required *footprint* (physical movement space) of a device (Hinckley, Cutrell, Bathiche & Muss, 2002; Jellinek & Card, 1990). See also Section 5.

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

2.2 A Brief Tour of Pointing Devices

Most operating systems treat all input devices as *virtual devices*, which tempts one to believe that pointing devices are interchangeable. However, the details of what the input device senses, how it is held, the presence or absence of buttons, and many other properties can significantly impact the interaction techniques, and hence the end-user tasks, that a device can support effectively. The following tour discusses properties to keep in mind for several common pointing devices.

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

Trackballs: A trackball senses the relative motion of a partially exposed ball in two degrees of freedom. Trackballs have a small working space (*footprint*), and afford use on an angled surface. Trackballs may require frequent clutching movements because users must lift and reposition their hand after rolling the ball a short distance. The buttons are located to the side of the ball, which can make them awkward to hold while rolling the ball (MacKenzie, Sellen & Buxton, 1991). Trackballs engage different muscle groups than a mouse, offering an alternative for users who experience discomfort.

Isometric Joysticks: An isometric joystick (e.g. the IBM Trackpoint) is a force-sensing joystick that returns to center when released. Most isometric joysticks are stiff, offering little feedback of the joystick's displacement. The rate of cursor movement is proportional to the force exerted on the stick; as a result users require practice to achieve good cursor control. Isometric joysticks may offer the only pointing option when space is at a premium (Douglas & Mithal, 1994; Rutledge & Selker, 1990; Zhai, et al., 1997).

Isotonic Joysticks: Isotonic joysticks sense angle of deflection. Some hybrid designs blur the distinctions isometric and isotonic joysticks, but the main questions are: does the joystick sense force or angular deflection; does the stick return to center when released; and does the stick move from the starting position. For a discussion of the complex design space of joysticks, see (Lipscomb & Pique, 1993).

Indirect Tablets: Indirect tablets report the absolute position of a pointer on a sensing surface. *Touch tablets* sense a bare finger, whereas *graphics tablets* or *digitizing tablets* typically sense a stylus or other physical intermediary. Tablets can operate in *absolute mode*, where there is a fixed C:D gain between the tablet surface and the display, or in *relative mode*, where the tablet responds only to motion of the stylus. If the user touches the stylus to the tablet in relative mode, the cursor resumes motion from its previous position; in absolute mode, it would jump to the new position. Absolute mode is generally preferable for tasks such as drawing, handwriting, tracing, or digitizing, but relative mode may be preferable for typical desktop interaction tasks such as selecting graphical icons or navigating through menus. Tablets thus allow coverage of many tasks (Buxton, Hill & Rowley, 1985), whereas mice *only* operate in relative mode.

Touchpads: Touchpads are small touch-sensitive tablets often found on laptop computers. Touchpads use relative mode for cursor control because they are too small to map to an entire screen, but most touchpads also have an absolute mode to allow features such as sliding along the edge of the pad to scroll. Touchpads support clicking by recognizing tapping or double-tapping gestures, but accidental contact (or loss of contact) can erroneously trigger such gestures (MacKenzie & Oniszczak, 1998). Like trackballs, the small size of touchpads necessitates frequent clutching, and touchpads can be awkward to use while holding down a button, unless the user employs their other hand.

Touchscreens and Pen-operated Devices: Touchscreens are transparent touch-sensitive tablets mounted on a display. Some touchscreens can only sense a bare finger; others can sense either a plastic stylus or a bare finger. Transparent electromagnetic digitizers, such as those found on the Tablet PC, cannot sense touch and require the use of a special pen. *Parallax error* is a mismatch between the sensed input position and the apparent input position due to viewing angle; look to minimize the displacement between the sensing and display surfaces to avoid this problem. Depending on the mounting angle, a touch or pen-operated display may result in arm or neck fatigue (Sears, Plaisant & Shneiderman, 1992). Touchscreens that are integrated with a mobile device can be prone to accidental contact when the user picks up the device. Yet mobile devices that only sense a pen require the user to unsheathe the stylus to perform any interaction; the user cannot quickly poke at the screen with a finger. The limited states and events sensed by pen or touch-operated devices raise additional design challenges, as discussed below.

2.3 Input Device States

There is a fundamental mismatch between the demands of graphical interfaces and the states and events that can be sensed by devices such as touchscreens and pen-operated devices, which makes it difficult to support the full set of graphical interface primitives, including click, drag, double-click, and right-click. There is no easy solution for nonstandard pointing devices that does not involve design compromises. When considering such devices, to make device limitations and differences concrete, one of the first things a designer should do is diagram all of these states and transitions.

Input devices taken in general support three possible states (Fig. 1): out-of-range, tracking, and dragging; practitioners refer to these as states 0, 1 and 2, respectively, of the 3-state model (Buxton, 1990). This model is useful for reasoning about the relationship between the events sensed by an input device and the demands of interaction techniques.

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

Fig. 1 Summary of states in Buxton's 3-state model. Adapted from (Buxton, 1990).

The 3-state model describes the mouse as a two-state device, supporting state 1, the cursor tracking state, as well as state 2, the dragging state. State 1 provides cursor feedback of the screen position that the device will act upon, while state 2 allows the user to drag an object by holding down the primary mouse button while moving the mouse. The mouse senses movement in both the tracking and dragging states, as represented by the *dx, dy* in each state (Fig. 2, left), indicating relative motion tracking capability.

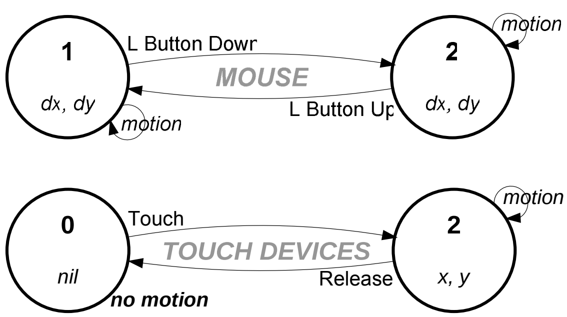

Fig. 2 States sensed by a mouse (left) versus states sensed by touch-operated devices such as touchpads (right). Adapted from (Hinckley, Czerwinski & Sinclair, 1998a).

Many touch-activated devices such as touchscreens, touchpads, and PDA screens are also two-state devices, but *do not sense the same two states as the mouse* (Fig. 2, right). For example, a PDA can sense a finger when it is in contact with the screen; this is the equivalent of the mouse dragging state (state 2). The PDA can also sense when the finger is removed from the screen, but once the finger breaks contact, this enters state 0 (out-of-range), where no motion can be detected (emphasized by the *nil* in state 0 of Fig. 2, right). Thus, although the mouse and PDA screen both sense two states, the lack of a second motion sensing state on the PDA means that it will be difficult to support the same interaction techniques as a mouse. For example, should sliding one's finger on the screen move a cursor, or drag an object? The designer must choose one; the PDA screen cannot support both behaviors at the same time.

The Tablet PC is an example of a pen-operated device that senses all three states of the 3-state model (Fig. 3). The Tablet PC senses the location of the stylus when it is proximate to the screen. The pen triggers an event when it makes contact with the screen as well as when it enters or leaves proximity.

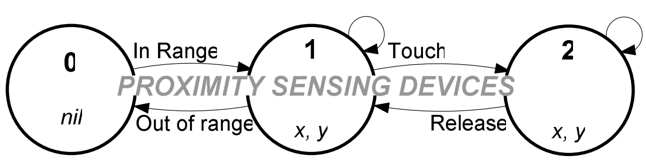

Fig. 3 States sensed by a Tablet PC pen. Adapted from (Hinckley, et al., 1998a).

Unfortunately, even with all three states, it is still difficult for a pen to support all the interaction techniques offered by a mouse. To help illustrate why this is the case, we can extend the 3-state model to more fully characterize the interaction techniques at the core of graphical user interfaces (Fig. 4). The resulting five states of graphical user interfaces are Tracking (1), Hover (1_H), Left Click (2_L), Dragging (2_D), and Right Click (2_R).

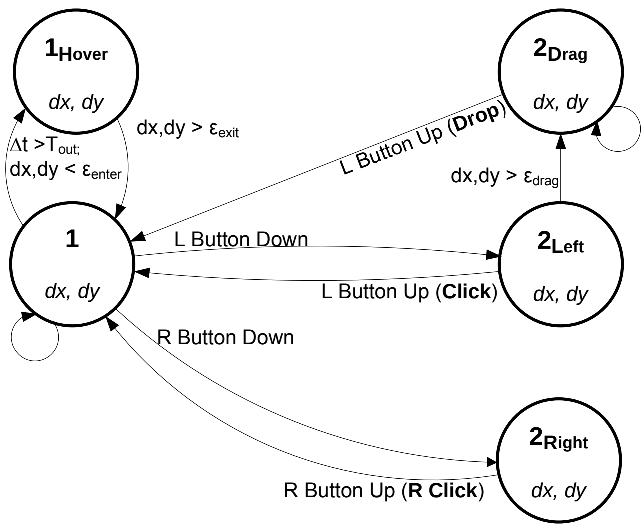

Fig. 4 The five states and transition rules at the core of graphical user interfaces.

If one wants to support all of these interactions on a pen or touch-based device, this diagram suggests that this will be difficult; there is no elegant solution in the literature. In this five-state model, a click is the series of transitions $1\text{-}2_L\text{-}1$ with no motion in state 2_L and a double click is $1\text{-}2_L\text{-}1\text{-}2_L\text{-}1$ with no motion between the two clicks. Even if the device can sense state 1, hovering (1_H) for help text requires holding the pointer motionless above the display. These gestures are all difficult to perform with a pen or finger because touching or lifting disturb the cursor position (Buxton, et al., 1985). Furthermore, because the pen may be out of range, and because the user must move the pen through the tracking zone to enter state 2, pen operated input devices lack a well defined current cursor position.

Pen and touch devices also lack a second button for right-click. A finger obviously has no buttons, but even a barrel button on a pen is slow to access and is prone to inadvertent activation (Li, Hinckley, Guan & Landay, 2005). Some mobile devices use dwelling with the pen or finger as a way to simulate right-click, but the timeout introduces an unavoidable delay; for rapid activation, it should be short as possible, but to avoid inadvertent activation (e.g. if the user is thinking about the next action to take while resting the pen on the screen) the timeout must be as long as possible. A 500 millisecond timeout offers a reasonable compromise (Hinckley, Baudisch, Ramos & Guimbretiere, 2005a). Even techniques designed for pen-operated devices (Apitz & Guimbretiere, 2004; Kurtenbach & Buxton, 1991; Moran, Chiu & van Melle, 1997) require rapid and unambiguous activation of one of several possible actions as fundamental building blocks; otherwise, inefficient or highly modal interactions become necessary and may reduce the appeal of such devices (Hinckley, et al., 2006).

Similar issues plague other interaction modalities, such as motion-sensing mobile devices (Hinckley, Pierce, Horvitz & Sinclair, 2005b), camera-based tracking of the hands (Wilson & Oliver, 2003), and 3D input devices (Hinckley, et al., 1994). All of these techniques require a method for users to move the device or their hands without accidentally performing an action. Thus state transitions form fundamental indications of intent that are essential for rapid and dependable interaction.

3 What's an Input Device For? The Composition of User Tasks

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

3.1 Elemental tasks

While computers can support many activities, at the input level some sub-tasks appear repeatedly in graphical user interfaces, such as pointing at a target on the screen, or typing a character. Foley et al. (Foley, Wallace & Chan, 1984) identified elemental tasks including *text* (entering symbolic data), *select* (indicating objects from a set of alternatives), *position* (pointing to a screen coordinate), and *quantify* (specifying an exact numeric value). But if these are elemental tasks, then where do devices like global positioning system (GPS) readers, cameras, or fingerprint scanners fit in? These offer new 'elemental' data types (e.g. location, images, and identity). Advances in technology will continue to yield data types that enable new tasks and scenarios of use.

3.2 Compound Tasks and Chunking

Another problem with the elemental task approach is that the level of analysis for 'elemental' tasks is not well defined. For example, a mouse indicates an *(x, y)* position on the screen, but an Etch-a-Sketch separates positioning into two sub-tasks by providing a single knob for *x* and a single knob for *y* (Buxton, 1986b). If *position* is an elemental task, why must we subdivide this task for some devices but not others? One way to resolve this puzzle is to view all tasks as hierarchies of sub-tasks (Fig. 5). Whether or not a task is "elemental" depends on the input device being used: the Etch-a-Sketch supports separate *QuantifyX* and *QuantifyY* tasks, whereas the mouse supports a compound *2D Position* task (Buxton, 1986a).

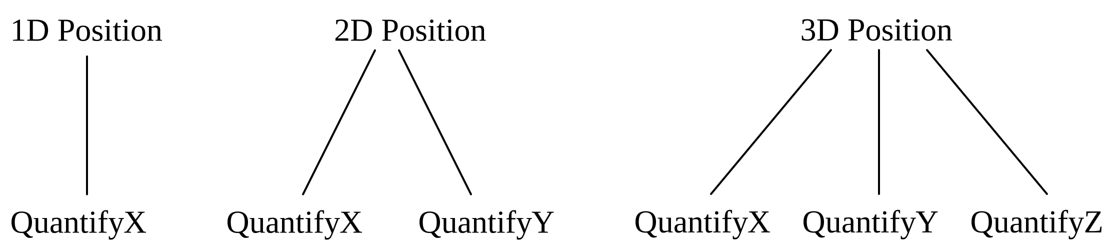

Fig. 5 Task hierarchies for 1D, 2D, and 3D position tasks.

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

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

4 Evaluation and Analysis of Input Devices

Beyond standard usability engineering techniques (see Chapters ??-??), there are a number of techniques tailored to the study of input devices. Representative tasks

(Buxton, 1995c), such as target acquisition, pursuit tracking, freehand drawing, and dragging versus tracking performance (MacKenzie, et al., 1991), can be used to formally or informally evaluate devices. Here, we focus on formal analysis using Fitts' Law, the Steering Law, and the Keystroke-Level Model.

4.1 Fitts' Law and Hick's Law

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

The standard Fitts task paradigm measures the movement time MT between two targets separated by amplitude A, with a width W of error tolerance (Fig. 6).

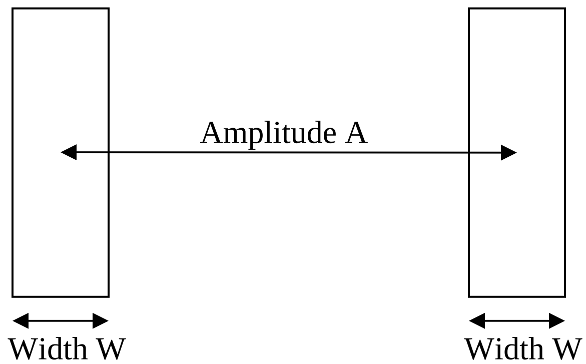

Fig. 6 Fitts' task paradigm (Fitts, 1954).

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

$$MT = a + b \log_2(A/W + 1) \qquad \text{(Equation 1)}$$

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

Psychomotor interpretations for Fitts' Law have been proposed (Douglas & Mithal, 1997). However, since the law characterizes the central tendency of a large number of pointing movements, the law may simply reflect information-theoretic entropy (MacKenzie, 1989). For example, Hick's Law, a model of decision time for a set of choices (e.g. in a menu), has a general form almost identical to Fitts Law:

$$H = \log_2(n + 1) \qquad \text{(Equation 2)}$$

Here n is the number of equally probably alternatives, and H is the entropy of the decision. If we view Fitts' task (Fig. 6) as a 'decision' along the amplitude A between n discrete targets of width W, this raises the possibility that Fitts' Law and Hick's Law are fundamentally the same law where $n = A/W$.

4.2 The Steering Law and Minimum Jerk Law

Steering a cursor through a narrow tunnel, as required to navigate a pull-down menu, is not a Fitts task because steering requires a continuous accuracy constraint: the cursor must stay within the tunnel at all times. For a straight line tunnel (Fig. 7) of width W and length A, for example, the Steering law predicts that movement time is a linear function of A and W:

$$MT = a + b\ A/W \qquad \text{(Equation 3)}$$

The Steering Law can also model arbitrary curved paths, as well as instantaneous velocity (Accot & Zhai, 1997). A limitation of the Steering Law is that it only models successful completion of the task; errors are not considered.

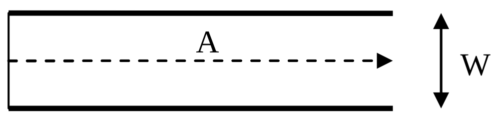

Fig. 7 The Steering Law for a straight tunnel. The user must follow the dotted line without moving beyond the borders. Adapted from (Accot & Zhai, 1999).

The Minimum Jerk Law (Viviani & Flash, 1995) characterizes the dynamics of motions that may follow a curved path but do not have a continuous accuracy constraint. The law states that unconstrained human movement trajectories tend to minimize the derivative of acceleration (jerk); one of its implications is that there is a two-thirds power law linking tangential velocity and path curvature. However, no one has yet formulated a universal law that handles varying accuracy constraints and curvature (Lank & Saund, 2005).

4.3 The Keystroke-Level Model (KLM) and GOMS Analysis

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

GOMS (Goals, Objects, Methods, and Selection Rules) models extend the keystroke-level model (John & Kieras, 1996). Some GOMS models can account for user knowledge and interleaving of tasks, but are more difficult to apply than the KLM. Both GOMS and KLM models are engineering tools that produce *estimates* for expert completion time of routine tasks. These models to not replace the need for usability testing and evaluation, but do offer a means to assess a design without implementing software, training end users, and evaluating their performance (Olson & Olson, 1990). Physical articulation times derived from KLM or GOMS analyses can also be used to help interpret results of empirical studies (Hinckley, et al., 2006). See also Chapter 58, Model-based Evaluations; Chapter 5, Cognitive Architectures; and Chapter 6, Modeling Humans in HCI.

5 Transfer Functions: How to transform an input signal

A transfer function is a mathematical transformation that scales the data from an input device. Typically the goal is to provide more stable and more intuitive control, but one can easily design a poor transfer function that hinders performance. A transfer function that matches the properties of an input device is known as an *appropriate mapping*. For force sensing input devices, the transfer function should be a *force-to-velocity* function: for example, the force one exerts on the IBM Trackpoint isometric joystick controls the speed at which the cursor moves. Other appropriate mappings include *position-to-position* or *velocity-to-velocity* functions, used with tablets and mice, respectively.

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

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

$$dx = K\ x^{\alpha} \qquad \text{(Equation 3)}$$

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

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

Absolute devices: It is possible to temporarily violate the 1:1 control-to-display mapping of absolute devices such as touchscreens by damping small motions to provide fine adjustments; large motions revert to an absolute 1:1 mapping (Sears & Shneiderman, 1991). A drawback of this technique is that cursor feedback in the tracking state becomes the default behavior of the device, rather than dragging (Buxton, 1990), but researchers are exploring ways to overcome this (Benko, Wilson & Baudisch, 2006).

6 Feedback: What happens in response to an input?

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

The industrial design suggests how to use a device before a user touches it (Norman, 1990). Mechanical sounds and vibrations produced by a device provide positive feedback for the user's actions (Lewis, Potosnak & Magyar, 1997). The shape of the device and the presence of landmarks can help users acquire a device without having to look at it (Hinckley, Pausch, Proffitt & Kassell, 1998b).

6.1 Proprioceptive and Kinesthetic Feedback

Internal sensations of body posture, motion, and muscle tension (Burdea, 1996; Gibson, 1962) may allow users to feel how they are moving an input device without looking at the device or even without visual feedback on a display. This is important when the user's attention is divided between multiple tasks and devices (Balakrishnan & Hinckley, 1999; Fitzmaurice & Buxton, 1997; Mine, Brooks & Sequin, 1997). Muscular tension can help to phrase together multiple related inputs (Buxton, 1986a) and may make mode transitions more salient to the user (Hinckley, et al., 2006; Raskin, 2000; Sellen, Kurtenbach & Buxton, 1992).

6.2 Kinesthetic Correspondence

Graphical feedback on the screen should correspond to the direction that the user moves the input device (Britton, Lipscomb & Pique, 1978). If the user moves a device to the left, then the object on the screen should likewise move left. However, users can easily adapt to certain kinds of non-correspondences: when the user moves a mouse forward and back, the cursor actually moves up and down on the screen; if the user drags a scrollbar downward, the text on the screen scrolls upwards. Researchers have also found that the dimensions of an input device should match the perceptual structure of a task (Jacob, Sibert, McFarlane & Mullen, 1994).

6.3 Snapping Behaviors and Active Haptic Feedback

Software constraints, such as snapping (Baudisch, Cutrell, Hinckley & Eversole, 2005), often suffice to support a user's tasks. Active force or tactile feedback (Burdea, 1996) can provide attractive forces for a target, or additional feedback for the boundaries of a target, but such feedback typically yields little or no performance advantage even for isolated target selection (Akamatsu & Mackenzie, 1996; MacKenzie, 1995). Such techniques must evaluate selection among multiple targets, because haptic feedback or snapping behavior for one target interferes with the selection of others (Grossman & Balakrishnan, 2005; Oakley, Brewster & Gray, 2001). *Visual dominance* refers to the tendency for vision to dominate other modalities (Wickens, 1992); haptic feedback typically must closely match visual feedback, which limits its utility as an independent modality (Campbell, Zhai, May & Maglio, 1999). One promising use of tactile feedback is to improve state transitions (Poupyrev & Maruyama, 2003; Snibbe & MacLean, 2001). For further discussion of active feedback modalities, see Chapter 10, Haptic Interfaces, and Chapter 11, Non-speech Auditory Output.

7 Keyboards and Text Entry

Typewriters have been in use for over 100 years; the QWERTY key layout dates to 1868 (Yamada, 1980). Despite the antiquity of the design, QWERTY keyboards are extremely well suited to human performance, and are unlikely to be supplanted by new key layouts, speech recognition technologies, or other techniques any time soon. Many factors can influence typing performance, including the size, shape, activation force, key travel distance, and the tactile and auditory feedback provided by striking the keys (Lewis, et al., 1997), but these well-established design details are not our focus here.

7.1 Procedural Memory

Procedural memory allows performance of complex sequences of practiced movements, seemingly without any cognitive effort (Anderson, 1980). Procedural memory enables touch typing on a keyboard with minimal attention while entering commonly used symbols. As a result, users can focus attention on mental composition and verification of the text appearing on the screen. Dedicated or chorded key presses for frequently used commands (hotkeys) likewise allow rapid command invocation (McLoone, Hinckley & Cutrell, 2003). The automation of skills in procedural memory is described by the *power law of practice*:

$$T = aP^b \qquad \text{(Equation 4)}$$

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

Alternative keyboard layouts such as Dvorak offer about a 5% performance gain (Lewis, et al., 1997), but the power law of practice suggests this small gain comes at a substantial cost for retraining time. However, ergonomic QWERTY keyboards do preserve much of a user's skill for typing. These split-angle keyboards are not faster, but some can help maintain neutral posture of the wrist, and thus avoid ulnar deviation (Honan, Serina, Tal & Rempel, 1995; Marklin, Simoneau & Monroe, 1997; Smutz, Serina, Bloom & Rempel, 1994), which has been associated with increased pressure in the carpal tunnel (Putz-Anderson, 1988; Rempel, Bach, Gordon & Tal, 1998).

7.2 Mobile Text Entry, Character Recognition, and Handwriting Recognition

The difficulty of entering text on handheld devices and cell phones has led to many new text entry techniques (MacKenzie, 2001), but most offer only 10-20 words-per-minute (wpm) typing rates, compared to approximately 60 wpm for a touch typist.

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

Soft keyboards and character recognition techniques are popular for pen-operated devices, but likewise are limited to serial entry. Soft keyboards depict keys in a graphical user interface to allow typing with a touchscreen or stylus. Design issues for soft keyboards differ from mechanical keyboards. Soft keyboards demand visual attention because the user *must look at the keyboard* to aim the pointing device. Only one key at a time can be touched, so much of the time is spent moving back and forth between keys (Zhai, Hunter & Smith, 2000). A soft keyboard can allow the user to draw gestures across multiple keys; in combination with a language model, this can allow entire words to be entered with a single pen gesture (Kristensson & Zhai, 2004).

Handwriting (even on paper, with no "recognition" involved) proceeds at about 15 wpm. Ink has significant value as a natural data type without recognition: it offers an expressive mix of writing, sketches, and diagrams. Although recognition technology continues to improve, recognizing natural handwriting remains difficult and error-prone for computers, and demands error correction input from the user. To make performance more predictable for the user, some devices instead rely on single-stroke gestures, known as *unistrokes* (Goldberg & Richardson, 1993), including *graffiti* for the PalmPilot. Unistroke alphabets attempt to strike a design balance such that each letter is easy for a computer to distinguish, yet also straightforward for users to learn (MacKenzie & Zhang, 1997).

8 Modalities of Interaction

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

8.1 Speech and Multimodal Input

Speech has substantial value without recognition. Computers can augment human-human communication across both time and space by allowing users to record, edit, replay, or transmit digitized speech and sounds (Arons, 1993; Buxton, 1995b; Stifelman, 1996). Systems can also use microphone input to detect ambient speech and employ this as a cue to help prioritize notifications (Horvitz, Jacobs & Hovel, 1999; Sawhney & Schmandt, 2000; Schmandt, Marmasse, Marti, Sawhney & Wheeler, 2000).

Computer understanding of human speech does not enable users to talk to a computer as one would converse with another person (but see also Chapter 8, Conversational Interface Technologies). Speech recognition can succeed for a limited vocabulary, such as speaking the name of a person from one's contact list to place a cell phone call. However, error rates increase as the vocabulary and complexity of the grammar grows, if the microphone input is poor, or if users employ "out-of-vocabulary" words. It is difficult to use speech to refer to spatial locations, so it cannot eliminate the need for pointing (Cohen & Sullivan, 1989; Oviatt, DeAngeli & Kuhn, 1997); see also Chapter 14, Multimodal Interfaces. Currently, keyboard-mouse text entry for the English language is about twice as fast as automatic speech recognition (Karat, Halverson, Horn & Karat, 1999); furthermore, speaking can interfere with one's ability to compose text and remember words (Karl, Pettey & Shneiderman, 1993). Finally, speech is inherently non-private in public situations. Thus, speech has an important role to play, but claims that speech will soon supplant manual input devices should be considered with skepticism.

8.2 Bimanual Input

People use both hands to accomplish most real-world tasks (Guiard, 1987), but computer interfaces make little use of the nonpreferred hand for tasks other than typing. Bimanual input enables compound input tasks such as navigation/selection tasks where the user can scroll with the nonpreferred hand while using the mouse in the preferred hand (Buxton & Myers, 1986). This assignment of roles to the hands corresponds to Guiard's kinematic chain theory (Guiard, 1987): the nonpreferred hand sets a frame of reference (scrolling to a location in the document) for the action of the preferred hand (selecting an item within the page using the mouse). Other applications for bimanual input include command selection (Bier, Stone, Pier, Buxton & DeRose, 1993; Kabbash, Buxton & Sellen, 1994), drawing tools (Kurtenbach, Fitzmaurice, Baudel & Buxton, 1997), and virtual camera control and manipulation (Balakrishnan & Kurtenbach, 1999; Hinckley, et al., 1998b). Integrating additional buttons and controls with keyboards to encourage bimanual interaction can also improve the efficiency of some common tasks (MacKenzie & Guiard, 2001; McLoone, et al., 2003).

8.3 Pen and Gesture Input

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

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

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

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

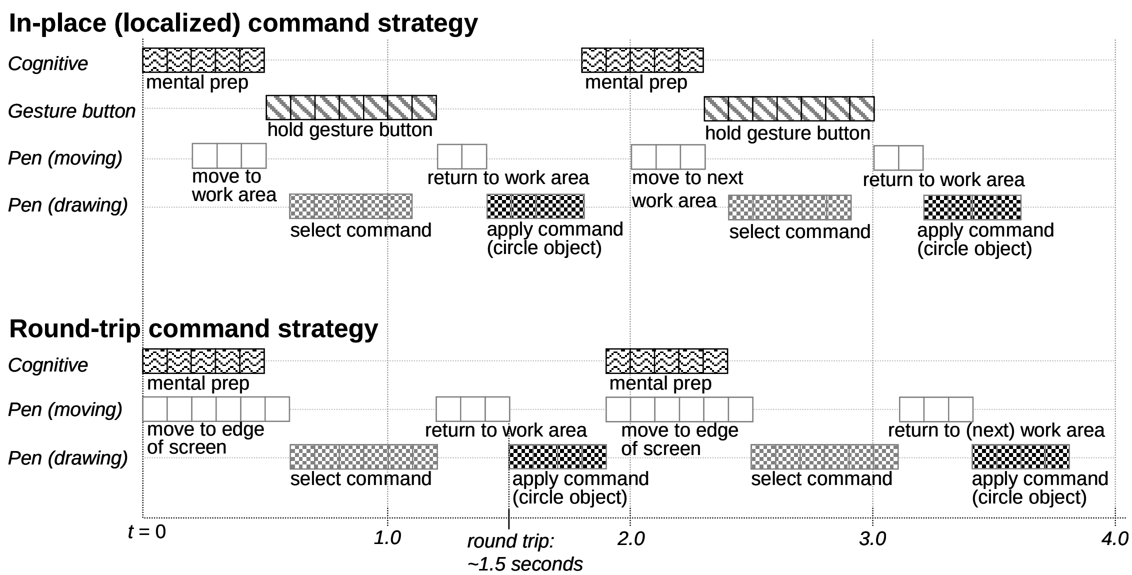

Fig. 8 Chart comparing task times for performing two successive commands on a tablet, with either in-place command selection (top) or round-trip command selection (bottom). The individual boxes in each subtask represent 100 millisecond intervals.

Thus, on a tablet-sized device it is difficult to realize a substantial time savings just by reducing round trips. For our hypothetical task of tagging keywords in a document, Fig X illustrates this predicament for average task times drawn from recent studies (Hinckley, et al., 2006; Li, et al., 2005). The chart shows two successive command selections, and assumes some mental preparation is required before issuing each command. Thus the potential benefit of pen gestures depends on the sequence of operations as well as the elimination of *multiple* round trips, as may be possible with techniques that integrate selection of verb, object, and indirect object (Hinckley, et al., 2005a; Kurtenbach & Buxton, 1991). Localized interaction also may offer indirect benefits by reducing physical effort and by keeping the user's visual attention focused on their work (Grossman, Hinckley, Baudisch, Agrawala & Balakrishnan, 2006; Kabbash, et al., 1994).

8.4 Whole Hand Input

Humans naturally gesture and point using their hands during verbal communication, which has motivated research into freehand gestures, often in combination with speech recognition (Bolt, 1980; Hauptmann, 1989; Wilson & Oliver, 2003). Cadoz categorizes hand gestures as semiotic, ergotic, or epistemic. Semiotic gestures, such as 'thumbs up,' communicate information (Rime & Schiaratura, 1991). Ergotic gestures manipulate physical objects. Epistemic gestures are exploratory motions that gather information (Kirsh, 1995; Kirsh & Maglio, 1994). The interaction literature focuses on empty-handed semiotic gestures (Freeman & Weissman, 1995; Jojic, Brumitt, Meyers & Harris, 2000; Maes, Darrell, Blumberg & Pentland, 1996). A major challenge is to correctly identify when a gesture, as opposed to an incidental hand movement, starts and stops (Baudel & Beaudouin-Lafon, 1993; Wilson & Oliver, 2003). The lack of deterministic state transitions (Buxton, 1990; Vogel & Balakrishnan, 2005) can lead to errors of user intent or errors of computer interpretation (Bellotti, Back, Edwards, Grinter, Lopes & Henderson, 2002). Other problems include fatigue from extending one's arms for long periods, and the imprecision of pointing at a distance. By contrast tangible interaction techniques (Ishii & Ullmer, 1997) and augmented devices (Harrison, Fishkin, Gujar, Mochon & Want, 1998) sense ergotic gestures via a physical intermediary (Hinckley, et al., 1998b; Zhai, et al., 1996). The emergence of cameras, cell phones, and tablets augmented with accelerometers and other sensors suggest this is a promising design space.

8.5 Background Sensing Techniques

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

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

8.6 Multi-Touch Tables and Screens

Technical advances have led to much recent research in touch-sensitive tables and projection screens. These technologies blur the distinction between whole-hand gestural input and traditional single-point touchscreens and touch tablets. Recent prototype systems demonstrate capture of the shape formed by the hand(s) in contact with a surface (Wilson, 2005), multiple points of contact (Han, 2005), or even images of objects placed on or near a surface (Matsushita & Rekimoto, 1997; Wilson, 2004). The DiamondTouch (Dietz & Leigh, 2001) is unique in its ability to determine which user produces each point of contact, which has led to a number of innovative applications and techniques (Shen, Everitt & Ryall, 2003; Wu & Balakrishnan, 2003). For an overview of design issues for tabletop systems, see (Scott, Grant & Mandryk, 2003).

8.7 A Society of Devices

Wireless networking is the technology that will most disrupt traditional approaches to human-computer interaction in the coming years, because it breaks down barriers between devices and enables a new society of devices that fill specific roles yet can still coordinate their activities (Fitzmaurice, Khan, Buxton, Kurtenbach & Balakrishnan, 2003b; Want & Borriello, 2000). The interaction designer thus must consider the full range of scale for display sizes and form-factors that may embody an interaction task, as well as the interactions between different types of devices. How can interaction migrate from watches, cell phones, handheld devices, and tablets, all the way up to desktop monitors, digital whiteboards, and interactive wall-sized displays?

As digital devices become physically smaller, the displays and input mechanisms they offer are shrinking. Considerable effort has been devoted to supporting web browsing in limited screen space (Buyukkokten, Garcia-Molina & Paepcke, 2001; Jones, Marsden, Mohd-Nasir, Boone & Buchanan, 1999; Trevor, Hilbert, Schilit & Koh, 2001). Techniques to make small displays virtually larger include peephole displays (Fitzmaurice, 1993; Yee, 2003), transparent overlays (Harrison, Ishii, Vicente & Buxton, 1995; Kamba, Elson, Harpold, Stamper & Sukaviriya, 1996), and using on-screen visuals to suggest the locations of off-screen objects (Baudisch & Rosenholtz, 2003). Touchscreens and touch-sensitive controls minimize the vertical profile of devices (Wherry, 2003), but may suffer from inadvertent activation. Physical manipulations such as tilting that use the device itself as an interface seem particularly well suited to small devices (Harrison, et al., 1998; Hinckley, et al., 2000; Rekimoto, 1996). Tiny, bright, and inexpensive laser or LED projectors are just around the corner; progress on computer vision techniques suggests that interactive projection may allow small devices to project large displays and sensing surfaces (Raskar, Beardsley, van Baar, Wang, Dietz, Lee, Leigh & Willwacher, 2004; Wilson, 2005).

At the other end of the spectrum, large-format displays are now affordable and common. Large displays lend themselves to collaboration and sharing of information with groups (Funkhouser & Li, 2000; Swaminathan & Sato, 1997) as well as giving a substantial physical presence to virtual activities (Buxton, Fitzmaurice, Balakrishnan & Kurtenbach, 2000; Trimble, Wales & Gossweiler, 2003). Researchers have explored pen and touchscreen interaction techniques for large displays (Guimbretiere, Stone & Winograd, 2001; Moran, et al., 1997). Unfortunately many technologies sense only a single point of contact. For interaction at a distance with large displays, it remains unclear what interaction techniques work best (Olsen & Nielsen, 2001; Vogel & Balakrishnan, 2004; Vogel & Balakrishnan, 2005); even when a user is close to a large display, interacting with portions of the display that are out view or beyond arm's length raise challenges (Bezerianos & Balakrishnan, 2005; Khan, Matejka, Fitzmaurice & Kurtenbach, 2005).

Displays of various sizes support different activities and social conventions; one of the principle challenges of ubiquitous computing (Weiser, 1991) is finding techniques that make it easy for users to work within a digital ecology that supports a range of tasks spanning multiple computers, displays, and interactive surfaces. Even on a single computer, users do not treat multiple monitors as one large display, but instead employ the boundary between displays to partition their tasks (Grudin, 2001). Several projects have probed how to use small displays as adjuncts to large ones (Myers, 2000; Myers, Stiel & Gargiulo, 1998; Rekimoto, 1998), allowing simultaneous interaction with private information on a personal device, and a shared or public context on a larger display. Users need techniques that allow them to access and share information across the boundaries of individual devices, as well as to dynamically bind together multiple devices and displays to accomplish their tasks (Hinckley, Ramos, Guimbretiere, Baudisch & Smith, 2004; Rekimoto, Ayatsuka & Kohno, 2003). Such interaction techniques inherently involve multiple persons and thus must consider how people use physical proximity and relative body orientation, a field known as *proxemics* (Deasy & Lasswell, 1985; Hinckley, et al., 2004; Sommer, 1965).

9 Current and Future Trends for Input

The designer of an interactive system should take a broad view of input, and consider not only traditional pointing techniques and graphical user interface widgets, but also issues such as search strategies to access information in the first place, sensor inputs that enable entirely new data types, synthesis techniques to extract meaningful structure from data, and integration of traditional technologies such as paper that offer fundamental strengths.

Good search tools may reduce the many inputs needed to manually search and navigate file systems. Knowledge work requires integration of external information from web pages or databases (Yee, Swearingen, Li & Hearst, 2003) as well as re-use of personal information from documents, electronic mail messages, and other content that a user has authored or viewed (Dumais, Cutrell, Cadiz, Jancke, Sarin & Robbins, 2003; Lansdale & Edmonds, 1992). Unified full-text indexing allows users to quickly query their personal information across multiple information silos, and can present information in the context of memory landmarks such as the date a message was sent, who authored the text, or the application used to create a document (Cutrell, Robbins, Dumais & Sarin, 2006).

New sensor inputs such as location and tagging technologies are coming to fruition. Radio-frequency identification (RFID) tags (Want, Fishkin, Gujar & Harrison, 1999) allow computers to identify tagged physical objects, thus enabling manipulation of ordinary objects as an input for computers. A mobile tag reader can identify tagged physical locations (Smith, Davenport & Hwa, 2003). Wireless communication technologies are poised to deliver ubiquitous location-based services (Schilit, et al., 1994; Want, Hopper, Falcao & Gibbons, 1992). Cell phones and low-power radios for wireless networking can sense their location or proximity to other devices via analysis of signal strengths (Bahl & Padmanabhan, 2000; Krumm & Hinckley, 2004). As another example, attempting to type a secure password on a mobile phone keypad quickly convinces one that biometric sensors or some other convenient means for establishing identity is essential for these devices to succeed. Such sensors could also make services such as personalization of interfaces much simpler.

The need to extract models and synthesize structure from large quantities of low-level inputs suggests that data mining and machine learning techniques will become important adjuncts to interaction (Fails & Olsen, 2003; Fitzmaurice, Balakrisnan & Kurtenbach, 1999; Horvitz, Breese, Heckerman, Hovel & Rommelse, 1998). Whenever a system considers automatic actions on behalf of the user, however, an important design principle in the face of uncertainty is to "do less, but do it well" (Horvitz, 1999).

With so many new technologies, it is easy to forget that paper remains an important input and display medium (Sellen & Harper, 2002). Paper is inexpensive, easy to annotate, rapid to access, comfortable to read, light to carry, and can be accessed after tens or even hundreds of years. Because technology will not replace paper any time soon, researchers are studying how to bring paper-like interfaces to digital devices (Schilit, Golovchinsky & Price, 1998; Wolf, Rhyne & Ellozy, 1989), as well as how to augment paper with digital capabilities (Guimbretiere, 2003; Liao, Guimbretiere & Hinckley, 2005; Stifelman, 1996).

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

10 References

G. Abowd and E. Mynatt (2000). Charting Past, Present, and Future Research in Ubiquitous Computing. ACM Transactions on Computer-Human Interaction 7 (1): 29-58.

J. Accot and S. Zhai (1997). Beyond Fitts' Law: Models for Trajectory-Based HCI Tasks. Proc. CHI'97: ACM Conference on Human Factors in Computing Systems, 295-302.

J. Accot and S. Zhai (1999). Performance Evaluation of Input Devices in Trajectory-based Tasks: An Application of the Steering Law. Proc. CHI'99, 466-472.

J. Accot and S. Zhai (2001). Scale Effects in Steering Law Tasks. Proc. CHI'2001 ACM Conference on Human Factors in Computing Systems, 1-8.

M. Akamatsu and I. S. Mackenzie (1996). Movement Characteristics Using a Mouse with Tactile and Force Feedback. International Journal of Human-Computer Studies 45: 483-493.

J. R. Anderson (1980). Chapter 8: Cognitive Skills. Cognitive Psychology and Its Implications. San Francisco, W. H. Freeman: 222-254.

G. Apitz and F. Guimbretiere (2004). CrossY: A crossing based drawing application. UIST 2004, 3-12.

B. Arons (1993). SpeechSkimmer: Interactively Skimming Recorded Speech. UIST'93 Symp. on User Interface Software & Technology, 187-195.

P. Bahl and V. Padmanabhan (2000). RADAR: An In-Building RF-Based User Location and Tracking System. IEEE 19th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2000), 775-784.

R. Balakrishnan, T. Baudel, G. Kurtenbach and G. Fitzmaurice (1997). The Rockin'Mouse: Integral 3D Manipulation on a Plane. CHI'97 Conf. on Human Factors in Computing Systems, 311-318.

R. Balakrishnan and K. Hinckley (1999). The Role of Kinesthetic Reference Frames in Two-Handed Input Performance. Proc. ACM UIST'99 Symp. on User Interface Software and Technology, 171-178.

R. Balakrishnan and G. Kurtenbach (1999). Exploring Bimanual Camera Control and Object Manipulation in 3D Graphics Interfaces. Proc. CHI'99 ACM Conf. on Human Factors in Computing Systems, 56-63.

T. Baudel and M. Beaudouin-Lafon (1993). Charade: Remote Control of Objects Using Hand Gestures. Communications of the ACM 36 (7): 28-35.

P. Baudisch, E. Cutrell, K. Hinckley and A. Eversole (2005). Snap-and-go: Helping Users Align Objects Without the Modality of Traditional Snapping. CHI 2005, 301-310.

P. Baudisch and R. Rosenholtz (2003). Halo: a technique for visualizing off-screen objects. CHI '03, 481-488.

V. Bellotti, M. Back, W. K. Edwards, R. Grinter, C. Lopes and A. Henderson (2002). Making sense of sensing systems: Five questions for designers and researchers. Proc. ACM CHI 2002 Conference on Human Factors in Computing Systems, 415-422.

H. Benko, A. Wilson and P. Baudisch (2006). Precise Selection Techniques for Multi-Touch Screens. CHI 2006, to appear.

A. Bezerianos and R. Balakrishnan (2005). The vacuum: facilitating the manipulation of distant objects. CHI '05, 361-370.

E. Bier, M. Stone, K. Pier, W. Buxton and T. DeRose (1993). Toolglass and Magic Lenses: The See-Through Interface. Proceedings of SIGGRAPH 93, 73-80.

R. Bolt (1980). Put-That-There: Voice and Gesture at the Graphics Interface. Computer Graphics (Aug.): 262-270.

D. Bowman, E. Kruijff, J. LaViola and I. Poupyrev (2004). 3D User Interfaces: Theory and Practice, Addison-Wesley.

E. Britton, J. Lipscomb and M. Pique (1978). Making Nested Rotations Convenient for the User. Computer Graphics 12 (3): 222-227.

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

W. Buxton (1983). Lexical and Pragmatic Considerations of Input Structure. Computer Graphics 17 (1): 31-37.

W. Buxton (1986a). Chunking and Phrasing and the Design of Human-Computer Dialogues. Information Processing `86, Proc. of the IFIP 10th World Computer Congress, 475-480.

W. Buxton (1986b). There's More to Interaction Than Meets the Eye. User Centered System Design: New Perspectives on Human-Computer Interaction. D. Norman and S. Draper. Hillsdale, N.J.: 319-337.

W. Buxton (1990). A three-state model of graphical input. Proc. INTERACT'90, 449-456.

W. Buxton (1995a). Integrating the Periphery and Context: A New Taxonomy of Telematics. Proceedings of Graphics Interface '95, 239-246.

W. Buxton (1995b). Speech, Language and Audition. Readings in Human-Computer Interaction: Toward the Year 2000. R. Baecker, J. Grudin, W. Buxton and S. Greenberg, Morgan Kaufmann Publishers: 525-537.

W. Buxton (1995c). Touch, Gesture, and Marking. Readings in Human-Computer Interaction: Toward the Year 2000. R. Baecker, J. Grudin, W. Buxton and S. Greenberg, Morgan Kaufmann Publishers: 469-482.

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

W. Buxton, R. Hill and P. Rowley (1985). Issues and Techniques in Touch-Sensitive Tablet Input. Computer Graphics 19 (3): 215-224.

W. Buxton and B. Myers (1986). A Study in Two-Handed Input. Proceedings of CHI'86: ACM Conference on Human Factors in Computing Systems, 321-326.

O. Buyukkokten, H. Garcia-Molina and A. Paepcke (2001). Accordion Summarization for End-Game Browsing on PDAs and Cellular Phones. ACM CHI 2001 Conf. on Human Factors in Computing Systems.

C. Campbell, S. Zhai, K. May and P. Maglio (1999). What You Feel Must Be What You See: Adding Tactile Feedback to the Trackpoint. Proceedings of INTERACT'99: 7th IFIP conference on Human Computer Interaction, 383-390.

S. Card, W. English and B. Burr (1978). Evaluation of mouse, rate-controlled isometric joystick, step keys, and text keys for text selection on a CRT. Ergonomics 21: 601-613.

S. Card, J. Mackinlay and G. Robertson (1990). The Design Space of Input Devices. Proc. ACM CHI'90 Conference on Human Factors in Computing Systems, 117-124.

S. Card, J. Mackinlay and G. Robertson (1991). A Morphological Analysis of the Design Space of Input Devices. ACM Transactions on Information Systems 9 (2): 99-122.

S. Card, T. Moran and A. Newell (1980). The Keystroke-Level Model for User Performance Time with Interactive Systems. Communications of the ACM 23 (7): 396-410.

E. Clarkson, J. Clawson, K. Lyons and T. Starner (2005). An empirical study of typing rates on mini-QWERTY keyboards. CHI Extended Abstracts, 1288-1291.

P. R. Cohen and J. W. Sullivan (1989). Synergistic Use of Direct Manipulation and Natural Language. Proc. ACM CHI'89 Conference on Human Factors in Computing Systems, 227-233.

E. Cutrell, D. Robbins, S. Dumais and R. Sarin (2006). Fast, Flexible Filtering with Phlat: Personal Search and Organization Made Easy. CHI 2006, to appear.

C. M. Deasy and T. E. Lasswell (1985). Designing Places for People: A Handbook on Human Behavior for Architects, Designers, and Facility Managers. New York, Whitney Library of Design an imprint of Watson-Guptill Publications.

P. Dietz and D. Leigh (2001). DiamondTouch: A Multi-User Touch Technology. CHI Letters 3 ((2)): 219-226.

R. F. Dillon, J. D. Eday and J. W. Tombaugh (1990). Measuring the True Cost of Command Selection: Techniques and Results. Proc. of ACM CHI'90 Conference on Human Factors in Computing Systems, 19-25.

S. Douglas, A. Kirkpatrick and I. S. MacKenzie (1999). Testing Pointing Device Performance and User Assessment with the ISO 9241, Part 9 Standard. Proc. ACM CHI'99 Conf. on Human Factors in Computing Systems, 215-222.

S. Douglas and A. Mithal (1994). The Effect of Reducing Homing Time on the Speed of a Finger-Controlled Isometric Pointing Device. Proc. ACM CHI'94 Conf. on Human Factors in Computing Systems, 411-416.

S. A. Douglas and A. K. Mithal (1997). Ergonomics of Computer Pointing Devices, Springer-Verlag.

S. Dumais, E. Cutrell, J. Cadiz, G. Jancke, R. Sarin and D. Robbins (2003). Stuff I've Seen: A system for personal information retrieval and re-use. SIGIR 2003, 72-79.

W. K. English, D. C. Englebart and M. L. Berman (1967). Display-selection Techniques for Text Manipulation. Transactions on Human Factors in Electronics 8 (1): 5-15.

J. A. Fails and D. R. Olsen (2003). Interactive machine learning. ACM Intelligent User Interfaces (IUI'03), 39-45.

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

G. Fitzmaurice and W. Buxton (1997). An Empirical Evaluation of Graspable User Interfaces: towards specialized, space-multiplexed input. Proceedings of CHI'97: ACM Conference on Human Factors in Computing Systems, 43-50.

G. Fitzmaurice, A. Khan, R. Pieke, B. Buxton and G. Kurtenbach (2003a). Tracking Menus. UIST 2003, 71-79.

G. W. Fitzmaurice (1993). Situated information spaces and spatially aware palmtop computers. Commun. ACM 36 (7 (Jul.)): 39 - 49.

G. W. Fitzmaurice, R. Balakrisnan and G. Kurtenbach (1999). Sampling, synthesis, and input devices. Commun. ACM 42 (8): 54 - 63.

G. W. Fitzmaurice, A. Khan, W. Buxton, G. Kurtenbach and R. Balakrishnan (2003b). Sentient Data Access via a Diverse Society of Devices. ACM Queue 1 (8 (Nov)).

J. D. Foley, V. Wallace and P. Chan (1984). The Human Factors of Computer Graphics Interaction Techniques. IEEE Computer Graphics and Applications (Nov.): 13-48.

W. T. Freeman and C. Weissman (1995). Television control by hand gestures. Intl. Workshop on Automatic Face and Gesture Recognition, 179-183.

T. Funkhouser and K. Li (2000). Large Format Displays. IEEE Comput. Graphics Appl. (July-Aug. special issue): 20-75.

J. Gibson (1962). Observations on Active Touch. Psychological Review 69 (6): 477-491.

D. Goldberg and C. Richardson (1993). Touch-Typing with a Stylus. Proc. INTERCHI'93 Conf. on Human Factors in Computing Systems, 80-87.

T. Grossman and R. Balakrishnan (2005). The bubble cursor: enhancing target acquisition by dynamic resizing of the cursor's activation area. CHI '05, 281-290.

T. Grossman, K. Hinckley, P. Baudisch, M. Agrawala and R. Balakrishnan (2006). Hover Widgets: Using the Tracking State to Extend the Capabilities of Pen-Operated Devices. CHI 2006, to appear.

J. Grudin (2001). Partitioning Digital Worlds: Focal and Peripheral Awareness in Multiple Monitor Use. CHI 2001, 458-465.

Y. Guiard (1987). Asymmetric Division of Labor in Human Skilled Bimanual Action: The Kinematic Chain as a Model. The Journal of Motor Behavior 19 (4): 486-517.

Y. Guiard, F. Buourgeois, D. Mottet and M. Beaudouin-Lafon (2001). Beyond the 10-bit Barrier: Fitts' Law in Multi-Scale Electronic Worlds. IHM-HCI 2001.

F. Guimbretiere (2003). Paper Augmented Digital Documents. UIST 2003.

F. Guimbretiere, M. C. Stone and T. Winograd (2001). Fluid Interaction with High-resolution Wall-size Displays. Proc. UIST 2001 Symp. on User Interface Software and Technology, 21-30.

J. Y. Han (2005). Low-Cost Multi-Touch Sensing through Frustrated Total Internal Reflection. UIST 2005, 115-118.

B. Harrison, K. Fishkin, A. Gujar, C. Mochon and R. Want (1998). Squeeze Me, Hold Me, Tilt Me! An Exploration of Manipulative User Interfaces. Proc. ACM CHI'98 Conf. on Human Factors in Computing Systems, 17-24.

B. Harrison, H. Ishii, K. Vicente and W. Buxton (1995). Transparent Layered User Interfaces: An Evaluation of a Display Design to Enhance Focused and Divided Attention. Proceedings of CHI'95: ACM Conference on Human Factors in Computing Systems, 317-324.

A. Hauptmann (1989). Speech and Gestures for Graphic Image Manipulation. Proceedings of CHI'89: ACM Conference on Human Factors in Computing Systems, 241-245.

K. Hinckley, P. Baudisch, G. Ramos and F. Guimbretiere (2005a). Design and Analysis of Delimiters for Selection-Action Pen Gesture Phrases in *Scriboli*. CHI 2005, 451-460.

K. Hinckley, E. Cutrell, S. Bathiche and T. Muss (2002). Quantitative Analysis of Scrolling Techniques. CHI 2002.

K. Hinckley, M. Czerwinski and M. Sinclair (1998a). Interaction and Modeling Techniques for Desktop Two-Handed Input. Proceedings of the ACM UIST'98 Symposium on User Interface Software and Technology, 49-58.

K. Hinckley, F. Guimbretiere, P. Baudisch, R. Sarin, M. Agrawala and E. Cutrell (2006). The Springboard: Multiple Modes in One Spring-Loaded Control. CHI 2006, to appear.

K. Hinckley, R. Pausch, J. C. Goble and N. F. Kassell (1994). A Survey of Design Issues in Spatial Input. Proceedings of the ACM UIST'94 Symposium on User Interface Software and Technology, 213-222.

K. Hinckley, R. Pausch, D. Proffitt and N. Kassell (1998b). Two-Handed Virtual Manipulation. ACM Transactions on Computer-Human Interaction 5 (3): 260-302.

K. Hinckley, J. Pierce, E. Horvitz and M. Sinclair (2005b). Foreground and Background Interaction with Sensor-Enhanced Mobile Devices. ACM TOCHI 12 (1 (Special Issue on Sensor-Based Interaction)): 31-52.

K. Hinckley, J. Pierce, M. Sinclair and E. Horvitz (2000). Sensing Techniques for Mobile Interaction. ACM UIST 2000 Symp. on User Interface Software & Technology, 91-100.

K. Hinckley, G. Ramos, F. Guimbretiere, P. Baudisch and M. Smith (2004). Stitching: Pen Gestures that Span Multiple Displays. ACM 7th International Working Conference on Advanced Visual Interfaces (AVI 2004), 23-31.

K. Hinckley, M. Sinclair, E. Hanson, R. Szeliski and M. Conway (1999). The VideoMouse: A Camera-Based Multi-Degree-of-Freedom Input Device. ACM UIST'99 Symp. on User Interface Software & Technology, 103-112.

M. Honan, E. Serina, R. Tal and D. Rempel (1995). Wrist Postures While Typing on a Standard and Split Keyboard. Proc. HFES Human Factors and Ergonomics Society 39th Annual Meeting, 366-368.

E. Horvitz (1999). Principles of Mixed-Initiative User Interfaces. Proc. ACM CHI'99 Conf. on Human Factors in Computing Systems, 159-166.

E. Horvitz, J. Breese, D. Heckerman, D. Hovel and K. Rommelse (1998). The Lumiere Project: Bayesian User Modeling for Inferring the Goals and Needs of Software Users. Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, Madison, WI, July 1998, pages 256-265. Morgan Kaufmann: San Francisco.

E. Horvitz, A. Jacobs and D. Hovel (1999). Attention-Sensitive Alerting. Proceedings of UAI '99, Conference on Uncertainty and Artificial Intelligence, 305-313.

H. Ishii and B. Ullmer (1997). Tangible Bits: Towards Seamless Interfaces between People, Bits, and Atoms. Proceedings of CHI'97: ACM Conference on Human Factors in Computing Systems, 234-241.

R. Jacob, L. Sibert, D. McFarlane and M. Mullen, Jr. (1994). Integrality and Separability of Input Devices. ACM Transactions on Computer-Human Interaction 1 (1): 3-26.

H. Jellinek and S. Card (1990). Powermice and User Performance. Proc. ACM CHI'90 Conf. on Human Factors in Computing Systems, 213-220.

B. E. John and D. Kieras (1996). Using GOMS for User Interface Design and Evaluation: Which Technique? ACM Trans. Comput.-Hum. Interact. 3 (4): 287-319.

N. Jojic, B. Brumitt, B. Meyers and S. Harris (2000). Detecting and estimating pointing gestures in dense disparity maps. Proceed. of IEEE Intl. Conf. on Automatic Face and Gesture Recognition.

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

P. Kabbash and W. Butxon (1995). The "Prince" Technique: Fitts' Law and Selection Using Area Cursors. Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems, 273-279.

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

T. Kamba, S. A. Elson, T. Harpold, T. Stamper and P. Sukaviriya (1996). Using small screen space more efficiently. Conference proceedings on Human factors in computing systems, 383.

C. Karat, C. Halverson, D. Horn and J. Karat (1999). Patterns of Entry and Correction in Large Vocabulary Continuous Speech Recognition Systems. Proc. ACM CHI'99 Conf. on Human Factors in Computing Systems, 568-575.

L. Karl, M. Pettey and B. Shneiderman (1993). Speech-Activated versus Mouse-Activated Commands for Word Processing Applications: An Empirical Evaluation. International Journal of Man-Machine Studies 39 (4): 667-687.

A. Khan, J. Matejka, G. Fitzmaurice and G. Kurtenbach (2005). Spotlight: directing users' attention on large displays. CHI '05, 791-798.

D. Kirsh (1995). Complementary strategies: why we use our hands when we think. Proceedings of 7th Annual Conference of the Cognitive Science Society., 212-217.

D. Kirsh and P. Maglio (1994). On Distinguishing Epistemic from Pragmatic Action. Cognitive Science 18 (4): 513-549.

P. Kristensson and S. Zhai (2004). SHARK2: a large vocabulary shorthand writing system for pen-based computers. Proc. UIST 2004 Symposium on User interface Software and Technology, 43-52.

J. Krumm and K. Hinckley (2004). The NearMe Wireless Proximity Server. Ubicomp 2004.

G. Kurtenbach and W. Buxton (1991). Issues in Combining Marking and Direct Manipulation Techniques. UIST'91, 137-144.

G. Kurtenbach, G. Fitzmaurice, T. Baudel and B. Buxton (1997). The Design of a GUI Paradigm based on Tablets, Two-hands, and Transparency. Proceedings of CHI'97: ACM Conference on Human Factors in Computing Systems, 35-42.

G. Kurtenbach, A. Sellen and W. Buxton (1993). An emprical evaluation of some articulatory and cognitive aspects of 'marking menus'. J. Human Computer Interaction 8 (1).

E. Lank and E. Saund (2005). Sloppy Selection: Providing an Accurate Interpretation of Imprecise Selection Gestures. Computers and Graphics 29 (4): 490 - 500.

M. Lansdale and E. Edmonds (1992). Using memory for events in the design of personal filing systems. Int. J. Man-Mach. Studies 36 (1): 97-126.

J. a. Z. LaViola, R. (2004). MathPad2: A System for the Creation and Exploration of Mathematical Sketches. ACM Transactions on Graphics 23 (3): 432-440.

J. Lewis, K. Potosnak and R. Magyar (1997). Keys and Keyboards. Handbook of Human-Computer Interaction. M. Helander, T. Landauer and P. Prabhu. Amsterdam, North-Holland: 1285-1316.

Y. Li, K. Hinckley, Z. Guan and J. A. Landay (2005). Experimental Analysis of Mode Switching Techniques in Pen-based User Interfaces. CHI 2005, 461-470.

C. Liao, F. Guimbretiere and K. Hinckley (2005). PapierCraft: A Command System for Interactive Paper. UIST 2005, 241-244.

J. Lipscomb and M. Pique (1993). Analog Input Device Physical Characteristics. SIGCHI Bulletin 25 (3): 40-45.

I. S. MacKenzie (1989). A note on the information-theoretic basis for Fitts' Law. Journal of Motor Behavior 21: 323-330.

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

I. S. MacKenzie (1992b). Movement Time Prediction in Human-Computer Interfaces. Graphics Interface '92, 140-150.

I. S. MacKenzie (1995). Input Devices and Interaction Techniques for Advanced Computing. Virtual environments and advanced interface design. W. Barfield and T. Furness. Oxford, UK, Oxford University Press: 437-470.

I. S. MacKenzie, Ed. (2001). special issue of Human-Computer Interaction journal on "Text Entry for Mobile Computing".

I. S. MacKenzie and Y. Guiard (2001). The Two-Handed Desktop Interface: Are We There Yet? Proc. ACM CHI 2001 Conf. on Human Factors in Computing Systems: Extended Abstracts, 351-352.

I. S. MacKenzie, H. Kober, D. Smith, T. Jones and E. Skepner (2001). LetterWise: Prefix-based Disambiguation for Mobile Text Input. ACM UIST 2001 Symp. on User Interface Software & Technology, 111-120.

I. S. MacKenzie and A. Oniszczak (1998). A Comparison of Three Selection Techniques for Touchpads. Proc. ACM CHI'98 Conf. on Human Factors in Computing Systems, 336-343.

I. S. MacKenzie, A. Sellen and W. Buxton (1991). A Comparison of Input Devices in Elemental Pointing and Dragging Tasks. Proc. ACM CHI '91 Conf. on Human Factors in Computing Systems, 161-166.

I. S. MacKenzie and R. W. Soukoreff (2002). A model of two-thumb text entry. Proceedings of Graphics Interface, 117-124.

I. S. MacKenzie and S. Zhang (1997). The Immediate Usability of Graffiti. Proc. Graphics Interface '97, 129-137.

P. Maes, T. Darrell, B. Blumberg and A. Pentland (1996). The ALIVE system: wireless, full-body interaction with autonomous agents. ACM Multimedia Systems (Special Issue on Multimedia and Multisensory Virutal Worlds).

R. Marklin, G. Simoneau and J. Monroe (1997). The Effect of Split and Vertically-Inclined Computer Keyboards on Wrist and Forearm Posture. Proc. HFES Human Factors and Ergonomics Society 41st Annual Meeting, 642-646.

N. Matsushita and J. Rekimoto (1997). Holo Wall: Designing a Finger, Hand, Body, and Object Sensitive Wall. Proceedings of the ACM UIST'97 Symposium on User Interface Software and Technology, 209-210.

H. McLoone, K. Hinckley and E. Cutrell (2003). Bimanual Interaction on the Microsoft Office Keyboard. INTERACT 2003.

Microsoft Corp. (2002). Windows XP Pointer Ballistics. http://www.microsoft.com/whdc/device/input/pointer-bal.mspx

M. Mine, F. Brooks and C. Sequin (1997). Moving Objects in Space: Expoiting Proprioception in Virtual-Environment Interaction. Computer Graphics 31 (Proc. SIGGRAPH'97): 19-26.

T. Moran, P. Chiu and W. van Melle (1997). Pen-Based Interaction Techniques for Organizing Material on an Electronic Whiteboard. UIST'97, 45-54.

B. Myers, Lie, K., Yang, B. (2000). Two-Handed Input using a PDA and a Mouse. CHI 2000, 41-48.

B. Myers, H. Stiel and R. Gargiulo (1998). Collaboration Using Multiple PDAs Connected to a PC. Proc. ACM CSCW'98 Conf. on Computer Supported Cooperative Work, 285-294.

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

I. Oakley, S. Brewster and P. Gray (2001). Solving Multi-Target Haptic Problems in Menu Interaction. Proc. ACM CHI 2001 Conf. on Human Factors in Computing Systems: Extended Abstracts, 357-358.

D. R. Olsen and T. Nielsen (2001). Laser pointer interaction. Proc. ACM CHI 2001 Conf. on Human Factors in Computing Systems, 17-22.

J. R. Olson and G. M. Olson (1990). The Growth of Cognitive Modeling in Human-Computer Interaction Since GOMS. Human-Computer Interaction 5 (2 and 3): 221-266.

S. Oviatt, A. DeAngeli and K. Kuhn (1997). Integration and Synchronization of Input Modes during Multimodal Human-Computer Interaction. CHI'97, 415-422.

I. Poupyrev and S. Maruyama (2003). Tactile Interfaces for Small Touch Screens. UIST 2003 Symp. on User Interface Software & Technolgy, 217-220.

V. Putz-Anderson (1988). Cumulative trauma disorders: A manual for musculoskeletal diseases of the upper limbs. Bristol, PA, Taylor & Francis.

R. Raskar, P. Beardsley, J. van Baar, Y. Wang, P. Dietz, J. Lee, D. Leigh and T. Willwacher (2004). RFIG lamps: interacting with a self-describing world via photosensing wireless tags and projectors. 23 (3): 406-415.

J. Raskin (2000). The Humane Interface: New Directions for Designing Interactive Systems, ACM Press.

J. Rekimoto (1996). Tilting Operations for Small Screen Interfaces. ACM UIST'96 Symposium on User Interface Sofware & Technology, 167-168.

J. Rekimoto (1998). A Multiple Device Approach for Supporting Whiteboard-based Interactions. CHI'98, 344-351.

J. Rekimoto, Y. Ayatsuka and M. Kohno (2003). SyncTap: An Interaction Technique for Mobile Networking. Mobile HCI 2003, 104-115.

D. Rempel, J. Bach, L. Gordon and R. Tal (1998). Effects of forearm pronation/supination on carpal tunnel pressure. J Hand Surgery 23 (1): 38-42.

B. Rime and L. Schiaratura (1991). Gesture and speech. Fundamentals of Nonverbal Behaviour. New York, Press Syndacate of the University of Cambridge: 239--281.

J. Rutledge and T. Selker (1990). Force-to-Motion Functions for Pointing. Proc. of Interact '90: The IFIP Conf. on Human-Computer Interaction, 701-706.

E. Saund and E. Lank (2003). Stylus Input and Editing Without Prior Selection of Mode. UIST'03, 213-216.

N. Sawhney and C. M. Schmandt (2000). Nomadic Radio: Speech and Audio Interaction for Contextual Messaging in Nomadic Environments. ACM Transactions on Computer-Human Interaction 7 (3): 353-383.

B. N. Schilit, N. I. Adams and R. Want (1994). Context-Aware Computing Applications. Proc. IEEE Workshop on Mobile Computing Systems and Applications, 85-90.

B. N. Schilit, G. Golovchinsky and M. N. Price (1998). Beyond paper: supporting active reading with free form digital ink annotations. CHI'98, 249-256.

C. M. Schmandt, N. Marmasse, S. Marti, N. Sawhney and S. Wheeler (2000). Everywhere messaging. IBM Systems Journal 39 (3&4).

S. D. Scott, K. D. Grant and R. L. Mandryk (2003). System Guidelines for Co-located, Collaborative Work on a Tabletop Display. Proc. ECSCW'03, European Conference Computer-Supported Cooperative Work, 159-178.

A. Sears, C. Plaisant and B. Shneiderman (1992). A New Era for High Precision Touchscreens. Advances in Human-Computer Interaction. Hartson and Hix, Ablex Publishers. 3: 1-33.

A. Sears and B. Shneiderman (1991). High Precision Touchscreens: Design Strategies and Comparisons with a Mouse. International Journal of Man-Machine Studies 34 (4): 593-613.

A. Sellen, G. Kurtenbach and W. Buxton (1992). The Prevention of Mode Errors through Sensory Feedback. J. Human Computer Interaction 7 (2): 141-164.

A. J. Sellen and H. R. Harper (2002). The myth of the paperless office. Cambridge, MA, MIT Press.

C. Shen, K. Everitt and K. Ryall (2003). UbiTable: Impromptu Face-to-Face Collaboration on Horizontal Interactive Surfaces. UbiComp 2003.

M. A. Smith, D. Davenport and H. Hwa (2003). AURA: A mobile platform for object and location annotation. Ubicomp 2003.

W. Smutz, E. Serina, T. Bloom and D. Rempel (1994). A System for Evaluating the Effect of Keyboard Design on Force, Posture, Comfort, and Productivity. Ergonomics 37 (10): 1649-1660.

S. Snibbe and K. MacLean (2001). Haptic Techniques for Media Control. CHI Letters (Proc. UIST 2001) 3 (2): 199-208.

R. Sommer (1965). Further studies of small group ecology. Sociometry 28: 337-348.

L. Stifelman (1996). Augmenting Real-World Objects: A Paper-Based Audio Notebook. CHI'96 Conference Companion, 199-200.

K. Swaminathan and S. Sato (1997). Interaction Design for Large Displays. interactions (Jan-Feb).

J. Trevor, D. M. Hilbert, B. N. Schilit and T. K. Koh (2001). From Desktop to Phonetop: A UI for Web Interaction on Very Small Devices. Proc. UIST '01 Symp. on User Interface Software and Technology, 121-130.

J. Trimble, R. Wales and R. Gossweiler (2003). NASA's MERBoard. Public and Situated Displays: Social and interactional aspects of shared display technologies. K. O'Hara, M. Perry, E. Churchill and D. Russell, Kluwer.

P. Viviani and T. Flash (1995). Minimum-jerk, two-thirds power law and isochrony: converging approaches to the study of movement planning. Journal of Experimental Psychology: Perception and Performance 21: 32-53.

D. Vogel and R. Balakrishnan (2004). Interactive public ambient displays: transitioning from implicit to explicit, public to personal, interaction with multiple users. UIST 2004.

D. Vogel and R. Balakrishnan (2005). Distant Freehand Pointing and Clicking on Very Large, High Resolution Displays. UIST 2005, 33-42.

R. Want and G. Borriello (2000). Survey on Information Appliances. IEEE Personal Communications (May/June): 24-31.

R. Want, K. P. Fishkin, A. Gujar and B. L. Harrison (1999). Bridging physical and virtual worlds with electronic tags. Proc. ACM CHI'99 Conf. on Human Factors in Computing Systems, 370-377.

R. Want, A. Hopper, V. Falcao and J. Gibbons (1992). The active bage location system. ACM Transactions on Information Systems 10 (1): 91-102.

M. Weiser (1991). The Computer for the 21st Century. Scientific American (September): 94-104.

E. Wherry (2003). Scroll ring performance evaluation. CHI '03 Extended Abstracts, 758-759.

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

D. Wigdor and R. Balakrishnan (2003). TiltText:Using Tilt for Text Input to Mobile Phones. ACM UIST'03 Symp. on User Interface Software & Technology, 81-90.

A. Wilson (2004). TouchLight: An Imaging Touch Screen and Display for Gesture-Based Interaction. International Conference on Multimodal Interfaces (ICMI).

A. Wilson (2005). PlayAnywhere: A Compact Interactive Tabletop Projection-Vision System.

A. Wilson and N. Oliver (2003). GWindows: Towards Robust Perception-Based UI. Proc. of CVPR 2003 (Workshop on Computer Vision for HCI).

F. R. Wilson (1998). The Hand: How its use shapes the brain, language, and human culture. New York, Pantheon Books.

C. Wolf, J. Rhyne and H. Ellozy (1989). The paper-like interface. Designing and Using Human Computer Interface and Knowledge-Based Systems. G. Salvendy and M. Smith. Amsterdam, Elsevier: 494-501.

M. Wu and R. Balakrishnan (2003). Multi-finger and whole hand gestural interaction techniques for multi-user tabletop displays. UIST 2003, 193-202.

H. Yamada (1980). A historical study of typewriters and typing methods: from the position of planning Japanese parallels. J. Information Processing 24 (4): 175-202.

K.-P. Yee (2003). Peephole Displays: Pen Interaction on Spatially Aware Handheld Computers. CHI 2003.

K.-P. Yee, K. Swearingen, K. Li and M. Hearst (2003). Faceted metadata for image search and browsing. CHI 2003, 401-408.

S. Zhai (1993). Human Performance Evaluation of Manipulation Schemes in Virtual Environments. Proc. IEEE Virtual Reality International Symposium (VRAIS'93), 155-161.

S. Zhai (1998). User Performance in Relation to 3D Input Device Design. Computer Grapics 32 (4, Nov.): 50-54.

S. Zhai, M. Hunter and B. A. Smith (2000). The Metropolis Keyboard- An Exploration of Quantitative Techniques for Virtual Keyboard Design. CHI Letters 2 (2): 119-128.

S. Zhai and P. Milgram (1993). Human Performance Evaluation of Isometric and Elastic Rate Controllers in a 6DoF Tracking Task. Proc. SPIE Telemanipulator Technology.

S. Zhai, P. Milgram and W. Buxton (1996). The Influence of Muscle Groups on Performance of Multiple Degree-of-Freedom Input. Proceedings of CHI'96: ACM Conference on Human Factors in Computing Systems, 308-315.

S. Zhai, B. A. Smith and T. Selker (1997). Improving Browsing Performance: A study of four input devices for scrolling and pointing tasks. Proc. INTERACT97: The Sixth IFIP Conf. on Human-Computer Interaction, 286-292.

S. Zhao and R. Balakrishnan (2004). Simple vs. Compound Mark Hierarchical Marking Menus. UIST 2004, 33-42.