# Input Technologies and Techniques

**Ken Hinckley**

Microsoft Research

One Microsoft Way

Redmond, WA 98052

kenh@microsoft.com

Tel: (425) 703-9065

DRAFT. To appear in:

*Chapter 7*
*Handbook of Human-Computer Interaction*
*Ed. By Andrew Sears & Julie A. Jacko*
*Lawrence Erlbaum & Associates*

## 1 Introduction: What's an input device anyway?

Input to computers consists of sensed information about physical properties (such as position, velocity, temperature, or pressure) of people, places, or things. For example, the computer mouse senses the motion imparted by the user's hand. But using an "input device" is a multifaceted experience that encompasses all of the following:

- *The physical sensor:* On mechanical mice, this typically consists of a rolling ball and optical encoder mechanism.
- *Feedback to the user*: For the mouse, feedback includes the visible cursor on the display, as well as the "clicking" sound and the tactile feel of the mouse buttons.
- *Ergonomic and industrial design:* This may include color, shape, texture, and the number and placement of buttons.
- *Interaction techniques:* Interaction techniques are the hardware and software elements that together provide a way for the user to accomplish a task. This

includes details of how sensor data is filtered and interpreted, the use of buttons to support selection and dragging, and input/output constructs such as scroll bars.

A poor design in any of these areas can lead to usability problems, so it is important to consider all of these aspects when designing or evaluating input to a computer system.

This chapter emphasizes continuous input sensors and their use in applications, but also includes a brief section on text entry. The chapter discusses important questions to ask about input technologies, techniques to effectively use input signals in applications, and models and theories that can be used to evaluate interaction techniques as well as to reason about design options.

## 2 Understanding Input Technologies

A designer who understands input technologies and the task requirements of users can choose input devices with properties that match all of the user's tasks as well as possible. But what are some of the important properties of input devices?

### 2.1 Pointing Device Properties

On the surface, the variety of pointing devices is bewildering. Fortunately there is a limited set of properties shared by many devices. These properties help a designer know what to ask about a device and anticipate potential problems. We will first consider these device properties in general, and then follow up with specific examples of common input

devices and how these properties apply to them. See the reviews of Buxton (1995c), Jacob (1996), MacKenzie (1995), and Greenstein (1997) for further discussion.

*Resolution and Accuracy*: The resolution of a sensor describes how many unique units of measure can be addressed. High resolution does not necessarily imply high accuracy, however, which is why you should always ask about both.

*Sampling Rate and Latency*: The *sampling rate*, measured in Hertz (Hz), indicates how often a sample is collected from a device. *Latency (lag)* is the delay between the onset of sampling and the resulting feedback to the user. It is impossible to completely eliminate latency from a system, and minimizing it can be difficult (Liang, Shaw & Green, 1991). Latency of more than about 75-100 milliseconds harms user performance for many interactive tasks (Robertson, Card & Mackinlay, 1989; MacKenzie & Ware, 1993).

*Noise, Aliasing, and Nonlinearity*: Electrical or mechanical imperfections in a device can add *noise* to a signal. *Aliasing* resulting from an inadequate sampling rate may cause rapid signal changes to be missed, just as a spinning wheel appears to stop or rotate backwards to the naked eye. Due to physical limitations, a sensor may exhibit an unequal response across its sensing range, known as *nonlinearity*.

*Absolute vs. Relative*: Does the device sense only relative changes to its position, or is the absolute value known? With absolute devices, the *nulling problem* arises if the position of a physical intermediary (e.g. a slider on a mixing console) is not in agreement with a

value set in the software, which can occur if the slider is used to control more than one variable. This problem cannot occur with relative devices, but time may be wasted *clutching* the device: if the mouse reaches the edge of the mouse pad, the user must pick up the mouse, move it, and put it back down in a comfortable position on the pad.

*Control-to-Display Ratio*: This is the ratio of the distance moved by an input device to the distance moved on the display, also known as *C:D gain*. Experts have criticized the concept of gain because it confounds what should be two measurements, device size and display size, in one arbitrary metric (Accot & Zhai, 2001; MacKenzie, 1995). The common belief that there is an optimal setting for the C:D gain is also controversial (Accot et al., 2001), since gain often exhibits little or no effect in experiments (Jellinek & Card, 1990), and further because faster performance may be offset by higher error rates (MacKenzie, 1995).

*Physical Property Sensed:* The type of physical property sensed by most pointing devices can be classified as position, motion, or force. This determines the type of transfer function that is most appropriate for the device (see Section 5.1).

*Number of Dimensions*: A device that senses position, for example, might be a one-dimensional slider, a 2D pointer, or even a 3D position tracker.

*Direct vs. Indirect*: On *direct* devices, the display surface is also the input surface. Examples include touchscreens, handheld devices with pen input, and light pens. All

other devices are *indirect*. Indirect devices often involve a mechanical intermediary that can easily become lost or moving parts that are subject to damage, which is one reason that touchscreens are popular for high volume applications such as shopping mall kiosks.

*Metrics of effectiveness*: Various other criteria can distinguish devices, including pointing speed and accuracy, error rates, *device acquisition time* (time to pick up and put down the device), learning time, footprint (how much space the device occupies), user preference, and cost (Card, Mackinlay & Robertson, 1990).

### 2.1.1 Taxonomies of Input Devices

Buxton classifies continuous, manually operated input devices by the property sensed versus the number of dimensions (Buxton, 1995b). Other device properties can be enumerated and organized into a decision tree that shows design choices along classification dimensions (Lipscomb & Pique, 1993). The taxonomy of Card, Mackinlay and Robertson (1991) includes composition operators, allowing description of interaction techniques and combination devices such as a radio with multiple buttons and knobs.

## 2.2 A Brief Tour of Pointing Devices

Often the term “mouse” is applied to any device that is capable of producing cursor motion, but what really is the difference between a mouse and a touchscreen, for example? Indeed, most operating systems treat input devices uniformly as *virtual devices*, so one might be tempted to believe that pointing devices are completely interchangeable.

As suggested above, 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 important properties to keep in mind for several common pointing devices. See also Bill Buxton's *Directory Of Sources For Input Technologies* for a comprehensive list of devices on the market (Buxton, 2001).

*Mice*: Douglas Englebart and colleagues (English, Englebart & Berman, 1967) invented the mouse in 1967 at the Stanford Research Institute. The long endurance of the mouse stands out in an era where technologies have become obsolete like last year's fashions, but the mouse is still in use because its properties match the demands of desktop graphical interfaces (Balakrishnan, Baudel, Kurtenbach & Fitzmaurice, 1997a). For typical pointing tasks on a computer, one can point with the mouse about as well as with the hand itself (Card, English & Burr, 1978). Furthermore, the mouse is stable: unlike a stylus used on a tablet, the mouse does not fall over when released, saving the user the time of picking it up again later. The mouse also doesn't tend to move when you press a button, and the muscle tension required to press the button has minimal interference with cursor motion compared to other devices. Finally, with mice, all of the muscle groups of the hand, wrist, arm, and shoulder contribute to pointing. This combination of muscle groups allows high performance for both rapid, coarse movements and slow, precise movements (Zhai, Milgram & Buxton, 1996; Guiard, 1987).

*Trackballs*: A trackball is essentially a mechanical mouse that has been turned upside-down. Since the trackball rolls in place, it stays at a fixed place and has a small working space (*footprint*). A trackball can also be mounted for use on an angled working surface. The buttons are located to the side of the ball, which can make them awkward to reach and hold while rolling the ball.

*Isometric Joysticks*: An isometric joystick (such as the IBM Trackpoint) is a force-sensing joystick that returns to center when released. Isometric joysticks require significant practice to achieve expert cursor control, but when integrated with a keyboard, it takes less time compared to the mouse for the user to acquire the joystick while typing, or to return to typing after pointing (Rutledge, 1990). However, this reduction in acquisition time is usually not enough to overcome the longer pointing time for the isometric joystick (Douglas & Mithal, 1994). Because isometric joysticks can have a tiny footprint, they are often used when space is at a premium.

*Isotonic Joysticks*: Isotonic joysticks sense the angle of deflection of the joystick, so most isotonic joysticks move from their center position. By contrast many isometric joysticks are stiff, with little or no “give” to provide the user feedback of how he or she is moving the joystick. Some hybrid designs blur the distinctions between the two types of joysticks, but the main questions to ask are: does the joystick sense force or angular deflection; does the stick return to center (zero value) when released; and does the stick move from the starting position. See also Lipscomb et al. (1993).

*Tablets*: Tablets (known variously as touch tablets, graphics tablets, or digitizing tablets) sense the absolute position of a pointing device on the tablet. Tablets might be used with the bare finger, a stylus, or a puck[1]. 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 icons or navigating through menus. Indeed, tablets can operate in either mode, allowing coverage of a wide range of tasks, whereas devices like mice and trackballs can *only* operate in relative mode.

*Touchpads*: Touchpads are small touch-sensitive tablets commonly used on laptop computers. Touchpads typically respond in relative mode, because of the small size of the pad. Most touchpads support an absolute mode to allow Asian language input or signature acquisition, for example, but for these uses make sure the touchpad can sense contact from a stylus (and not just the bare finger). Touchpads often support clicking by recognizing tapping or double-tapping gestures, but accidental contact (or loss of contact) can erroneously trigger such gestures (MacKenzie & Oniszczak, 1998).

---

[1] A puck is a mouse that is used on a tablet – the primary difference is that a puck usually senses absolute position, while a mouse senses motion only.

*Touchscreens*: Touchscreens are transparent touch-sensitive tablets mounted on a display. *Parallax error* is a mismatch between the sensed finger position and the apparent finger position due to viewing angle and displacement between the sensing and display surfaces. Check the *transmissivity* of a touchscreen, as it may reduce the luminance and contrast ratio of the display. Depending on the mounting angle, touchscreens may result in arm fatigue. Also, when the user drags a finger across the screen, this must either move the cursor, or drag on object on the screen; it cannot do both because there is no separation between the two actions (Buxton, 1990b). Finally, offset the selected position to be visible above the fingertip, and present new information in an area of the screen unlikely to be occluded by the user's hand. See also Sears et al. (1991).

*Pen Input Devices*: The issues noted above for touchscreens also apply to pen input on handheld devices. Furthermore, users often want to touch the screen using a bare finger as well as using the stylus, so commonly used commands should be large enough to accommodate this. Keep in mind that there is no true equivalent of the mouse "hover" state for calling up tool tips, nor is there any extra button for context menus. See also Chapter 32, Designing Interfaces for Handheld Computers.

*Alternative Pointing Devices*: For disabled use, or tasks where the user's hands may be occupied, standard pointing devices may be unsatisfactory. Some alternatives, roughly ordered from most the practical solutions with current technology to the most preliminary solutions still in the research labs, include the following:

- *Software Aids*: Because people are so resourceful and adaptable, modification of system software through screen magnification, sticky modifier keys, and other such techniques can often enable access to technology.
- *Feet for input*. Foot-operated devices can provide effective pointing control (Pearson & Weiser, 1988); input using the knee is also possible (English et al., 1967). Foot switches and rocker pedals are useful for specifying modes or controlling secondary values (Sellen, Kurtenbach & Buxton, 1992; Balakrishnan, Fitzmaurice, Kurtenbach & Singh, 1999).
- *Head Tracking*: It is possible to track the position and orientation of the user's head, but unfortunately, the neck muscles offer low bandwidth cursor control compared to the hands. Head tracking is a natural choice for viewpoint control in virtual environments (Sutherland, 1968; Brooks, 1988).
- *Eye Tracking*: Eye tracking has several human factors and technology limitations. The human eye fixates visual targets within the fovea, which fundamentally limits the accuracy of eye gaze tracking to 1 degree of the field of view (Zhai, Morimoto & Ihde, 1999). The eye jumps around constantly, moving rapidly in saccades between brief fixation points, so a high sampling rate and intelligent filtering is necessary to make sense of eye tracking data. If one uses eye gaze to execute commands, the *Midas touch problem* results, because the user cannot glance at a command without activating it (Jacob, 1991). Combining manual input and eye tracking offers another approach (Zhai et al., 1999).
- *Direct Brain Interfacing*. In some cases, direct access to the brain offers the only hope for communication. Electrodes have been surgically implanted in the motor

cortex of human patients; signal processing of the data allows imagined hand movements to move a cursor on the screen (Moore, Kennedy, Mynatt & Mankoff, 2001). Clinical trials of this approach are currently *extremely* limited.

See also Interaction Issues for Diverse Users (Chapters 20-26), speech recognition technologies as discussed in Chapters 8, 14, and 36, and alternative devices listed by Buxton (2001).

## 2.3 Input Device States

The integration of buttons with devices to support clicking and dragging may seem trivial, but failing to recognize these fundamental states of input devices can lead to design problems. The three-state model (Buxton, 1990b) enumerates the states recognized by commonly available input devices: *Tracking* (State 1), *Dragging* (State 2), and *Out of Range* (State 0), as shown in Fig. 1. The 3-state model is useful to specify exactly what an input device can do in relation to the demands of interaction techniques.

| **State** | **Description** |
|---|---|
| 0 | *Out Of Range:* The device is not in its physical tracking range. |
| 1 | *Tracking:* Moving the device causes the tracking symbol to move. |
| 2 | *Dragging:* Allows one to move objects in the interface. |

Fig. 1 Summary of states in Buxton's 3-state model.

For example, Fig. 2 compares the states supported by a mouse to those supported by a touchpad. A mouse operates in the Tracking state, or in the Dragging state (when one

moves the mouse while holding the primary mouse button), but it cannot sense the Out Of Range state: it does not report an event if it is lifted. A touchpad also senses two states, but they are not the same two states as the mouse. A touchpad senses a Tracking state (the cursor moves while one's finger is on the pad) and an Out Of Range state (the pad senses when the finger is removed).

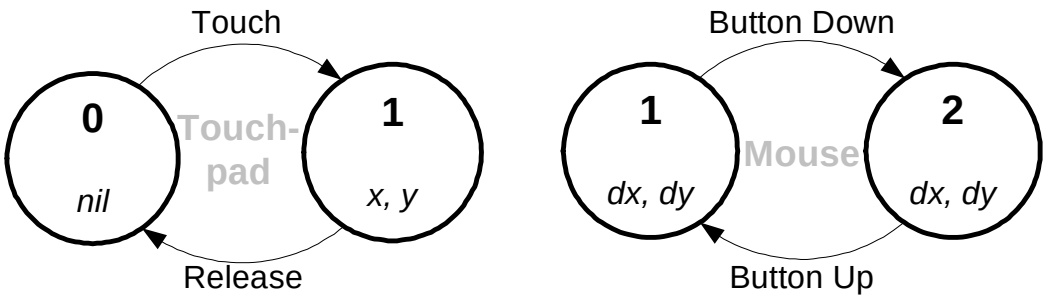


Fig. 2 State models for a touchpad (left) and a standard mouse (right).

The continuous values that an input device senses also may depend on these states (Hinckley, Czerwinski & Sinclair, 1998). For example, the *(x, y)* position of one's finger is sensed while it contacts a touchpad, but nothing is sensed once the finger breaks contact. Thus the touchpad has a single state where it senses a position, whereas the mouse has two states where it senses motion; any interaction technique that requires sensing motion in two different states will require special treatment on a touchpad.

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

Input devices are used to complete elemental tasks on a computer. One way of reasoning about input devices and interaction techniques is to view the 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, Wallace, and Chan (1984) propose that all user interface transactions are composed of the following six elemental tasks:

- *Select*: Indicating object(s) from a set of alternatives.
- *Position*: Specifying a position within a range, such as a screen coordinate.
- *Orient:* Specifying a rotation, such as an angle, or the three-dimensional orientation of an object in a virtual environment.
- *Path*: Specifying a series of positions or orientations over time, such as drawing a freehand curve in a paint program.
- *Quantify*: Specifying a single numeric value.
- *Text*: Specifying symbolic data such as a sequence of characters.

If a computer system allows the user to accomplish all six of these elemental tasks, then in principle the user can use the system to accomplish any computer-based task.

## 3.2 Compound Tasks and Chunking

While elemental tasks are useful to find commonalities between input devices and different ways of accomplishing the same task on different systems, the level of analysis at which one specifies "elemental" tasks is not well defined. For example, a mouse can indicate an integral *(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*.

If positioning is an elemental task, why do we find ourselves subdividing this task for some devices but not others? What if we are working with three-dimensional computer graphics where a *Position* requires an *(x, y, z)* coordinate? One way to resolve this puzzle is to view all tasks as hierarchies of sub-tasks (Fig. 3). Observe that 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 single compound *2D Position* task (Buxton, 1986).

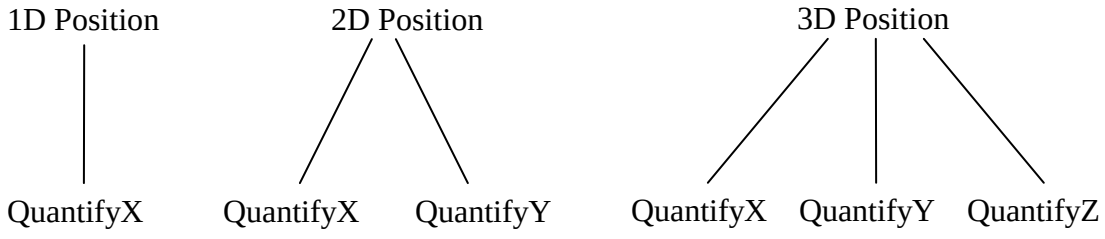


Fig. 3 Task hierarchies for 1D, 2D, and 3D position tasks.

From the user's perspective, a series of elemental tasks often seems 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). One can design the interaction so that it encourages the user to work at the higher level of the compound task, such as 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. This is a very important point. The appropriate choice and 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. In short, the design of the system has a direct influence on the steps of the workflow that must be exposed to the user (Buxton, 1986).

### 3.2.1 Multi-Channel Input Devices

Because many user tasks represent parts of larger compound tasks, it is sometimes desirable to provide a *multi-channel input device*, which provides multiple controls on a single device (Zhai, Smith & Selker, 1997). For example, many mice include dedicated scrolling devices such as wheels, isometric joysticks, or touchpads. One advantage of

such devices is that they eliminate the need to remove visual attention from one's work, as is necessary when moving the cursor to interact with standard scroll bars.

### 3.2.2 Multiple Degree-of-Freedom Input Devices

Multiple degree-of-freedom (Multi-DOF) devices sense multiple dimensions of spatial position or orientation, unlike multi-channel devices, which provide extra input dimensions as separate controls. Examples include mice that can sense when they are rotated or tilted (Kurtenbach, Fitzmaurice, Baudel & Buxton, 1997; Balakrishnan et al., 1997a; Hinckley, Sinclair, Hanson, Szeliski & Conway, 1999) and magnetic trackers that sense six degree-of-freedom motion (3D position and 3D orientation). Numerous interaction techniques have been proposed to allow standard 2D pointing devices to control 3D positioning or orientation tasks (Conner et al., 1992). Such techniques may be less effective for 3D input tasks than well-designed Multi-DOF input techniques (Ware, 1990; Balakrishnan et al., 1997a; Hinckley, Tullio, Pausch, Proffitt & Kassell, 1997). However, 3D input devices can be ineffective for many standard desktop tasks, so overall performance for all tasks must be considered. See (Hinckley, Pausch, Goble & Kassell, 1994b) and (Zhai, 1998) for further discussion of 3D input techniques. See also Chapter 31, Virtual Environments and Augmented Realities.

### 3.2.3 Bimanual Input

People use both hands to accomplish most real-world tasks (Guiard, 1987), but with computers, there is very little use of the nonpreferred hand for tasks other than typing.

Bimanual input enables compound input tasks such as navigation / selection where the user can scroll with the nonpreferred hand while using the mouse in the preferred hand (Buxton et al., 1986). This assignment of roles to the hands corresponds to a theory of bimanual action proposed by Guiard (Guiard, 1987), which suggests that the nonpreferred hand sets a frame of reference (scrolling to a location in the document) for the more precise actions of the preferred hand (selecting an item within the page using the mouse). Other applications for bimanual input include command selection with the nonpreferred hand (Bier, Stone, Pier, Buxton & DeRose, 1993; Kabbash, Buxton & Sellen, 1994), drawing programs (Kurtenbach et al., 1997), and virtual camera control and 3D manipulation (Hinckley, Pausch, Proffitt & Kassell, 1998; Balakrishnan & Kurtenbach, 1999c).

### 3.2.4 Gesture Input Techniques

With mouse input, or preferably pen input, users can draw simple gestures analogous to proofreader's marks to issue commands, such as scribbling over a word to delete it. Note that the gesture integrates the command (delete) and the selection of the object to delete. 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 of the command by selecting the command, selecting the extent of text to move, and selecting the location to move the text [Buxton, 1995 #97]. A design tradeoff is whether to treat a gesture as content ("ink") or recognize it immediately as a command. With careful design, the approaches can be combined (Kramer, 1994; Moran, Chiu & van Melle, 1997). *Marking menus* use simple

straight-line gestures to speed menu selection (Kurtenbach, Sellen & Buxton, 1993). *Multimodal Input* (Chapter 14) combines speech input with pen gestures. Future interfaces may use sensors to recognize whole-hand gestures (Baudel & Beaudouin-Lafon, 1993) or movements of physical tools, such as handheld computers (Hinckley, Pierce, Sinclair & Horvitz, 2000; Bartlett, 2000).

Gesture input has technical challenges and limitations. A pragmatic difficulty is to recognize the gesture, using techniques such as the Rubine classifier (Rubine, 1991). As the number of gestures increases, recognition becomes more difficult, and it is harder for the user to learn, remember, and correctly articulate each gesture, which together limit the number of commands that can be supported (Zeleznik, Herndon & Hughes, 1996).

## 4 Evaluation and Analysis of Input Devices

In addition to general usability engineering techniques (see Chapters 55-59), there are a number of techniques specifically tailored to the study of input devices.

### 4.1 Representative Tasks for Pointing Devices

To test an input device, it is useful to see how the device performs in a wide range of tasks that represent actions users may perform with the device. Some examples are listed below (see also Buxton, 1995c). These tasks are appropriate for quick, informal studies, or they can be formalized for quantitative experiments.

*Target acquisition*: The user clicks back and forth between targets on the screen. Vary the distance between the targets and the size of the targets. This can be formalized using Fitts' Law (Fitts, 1954; MacKenzie, 1992a), as discussed in section 4.3.

*Steering*: The user moves the cursor through a tunnel. Typically circular and straight-line tunnels are used; vary the length, width, and direction of travel. This task can be modeled using the Steering Law (Accot & Zhai, 1997); see section 4.4.

*Pursuit tracking*: The user follows a randomly moving target with the cursor. Average root-mean-squared (RMS) error quantifies the performance. See (Zhai, Buxton & Milgram, 1994) for an example of this type of study.

*Freehand drawing*: Try signing your name in different sizes (Buxton, 1995b) or rapidly drawing a series of XOXO's, for example.

*Drawing lines*: Draw a box and see if it is easy to get the ends of the strokes to line up with one another. Also try a diamond and a box slightly off-axis (rotated 5 degrees).

*Tracing and digitizing*: Some tablets can digitize coordinates from paper drawings or maps. The accuracy and linearity of the device, particularly when the pointer is sensed through different types of materials that the user may need to digitize, should be tested.

*Rapid or slow motion*: Does the device respond quickly and accurately when rapidly moving back and forth or in circles? Can one hold the device still to bring up tool tips? Look for signs of jitter when slowing to select a very tiny target.

*Clicking, double clicking, and dragging*: Does the device support each effectively? One should also consider the placement of the buttons. Are they awkward to reach or hold while moving the device? Does hitting the buttons cause unintended cursor motion? Are any of them easy to hit by mistake?

## 4.2 Ergonomic Issues for Input Devices

Many modern information workers suffer from repetitive strain injury (RSI). Researchers have identified many risk factors for such injuries, such as working under stress or taking inadequate rest breaks (Putz-Anderson, 1988). Some themes that encourage ergonomic device design (Pekelney & Chu, 1995) are listed below. See also section 7.3 for a brief discussion of ergonomic keyboards, and Chapter 19, Workstation Design and HCI, for a full treatment of this important topic. Example ergonomic design themes include:

- *Reduce Repetition* by using one-step instead of multiple-step operations, or by providing alternative means (e.g. keyboard shortcuts) to accomplish tasks.
- *Minimize Force* required for dynamic load to hold and move the device, as well as static load required to hold down a button. Also, avoid sharp edges that might put pressure on the soft tissues of the hand.
- *Natural and Neutral Postures*: The device should accommodate a range of hand sizes, and should not restrict hand motion. Design for the optimal direction of

movement for the finger, hand, or arm used to operate the device. Devices should be held with a natural, slight flexion of fingers, straight or slight extension and ulnar deviation of the wrist, and slight pronation of the wrist (Rempel, Bach, Gordon & Tal, 1998; Keir, Back & Rempel, 1999).

- *Cues for Use*: Devices should communicate a clear orientation for grip and motion to discourage inappropriate grips that may lead to problems.

## 4.3 Fitts' Law: A Design, Engineering, and Research Tool

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, summarized as the *bandwidth* of the input device. Industry practitioners consider even a 5% difference in bandwidth to be important. See MacKenzie (1992a, 1992b), and Douglas, Kirkpatrick & MacKenzie (1999) for details of conducting such studies.

### 4.3.1 What is Fitts' Law?

Fitts' Law was first described by Paul Fitts in 1954. The law as Fitts envisioned it had foundations in information theory, although psychomotor interpretations have also been suggested (Douglas & Mithal, 1997). Fitts conducted an experiment where participants were asked to use a stylus to rapidly tap back and forth between two metal plates, while missing the plates as infrequently as possible. Fitts measured the movement time MT

between the two plates, while varying the amplitude A of the movement, as well as the width W of the error tolerance (Fig. 4).

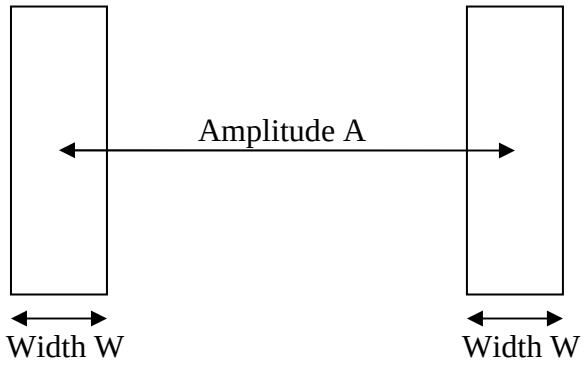


Fig. 4 Fitts' task paradigm

Fitts discovered that the average movement time is predicted by a logarithmic function of the ratio of A to W. The formulation of Fitts' Law typically used in device studies is:

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

Here MT is the movement time, A is the amplitude of the movement, and W is the width of the target that must be selected. The constants a and b are coefficients fit to the average of the observed data for MT for each combination of A and W tested in the experiment. One calculates a and b by the normal techniques of linear regression, using a statistical package or spreadsheet program. In typical rapid, aimed movement studies, it is not unusual to see Fitts' Law model the mean observed movement times with more than 85% of all variance explained by Equation 1. For example experimental data, see MacKenzie (1992a, 1992b).

The Fitts *Index of Difficulty* (ID), measured in units of *bits*, is defined as:

$$ID = \log_2(A/W + 1) \qquad \text{(Equation 2)}$$

The Index of Difficulty is the key concept of Fitts' Law, as it describes moves in the task space using a single abstract metric of difficulty. The ID balances the intuitions that far movements should take more time than short ones, while pointing to small targets should take longer than large ones. Note that the units of A and W cancel out, so the ID is dimensionless. It is arbitrarily assigned the units of *bits* because of the base 2 logarithm. By substitution of the ID, Fitts' Law (Equation 1) often appears in the literature as:

$$MT = a + b\ ID \qquad \text{(Equation 3)}$$

The Index of Performance (IP), measured in *bits/second*, quantifies the bandwidth of a pointing device. In the information-theoretic view of Fitts' Law, the IP captures the rate of information transfer through the human-computer interface, defined as:

$$IP = MT/ID \qquad \text{(Equation 4)}$$

Dropping the constant offset $a$ from Equation 3 and dividing MT by ID, the IP can also be calculated as:

$$IP = 1/b \qquad \text{(Equation 5)}$$

When "bandwidth," "throughput," or IP is mentioned in experimental studies, this quantity is typically the inverse slope as calculated by Equation 5. Comparing the *ratio* of IP's is a good way to make comparisons across Fitts' Law studies (MacKenzie, 1992a). Although Equation 5 is normally reported as the IP, Equation 4 is sometimes used to calculate an IP for an individual experimental condition, or even an individual pointing movement, where a "regression line" is nonsensical.

A final quantity known as the *effective width* is sometimes used to account for both speed and accuracy of performance in Fitts' Law studies. The effective width is the target width that would be required to statistically correct the subject's performance to a constant 4% error rate (MacKenzie, 1992a), using the standard deviation σ of endpoint coordinates around the target. For each combination of A and W tested in an experiment, the effective width $W_e$ for that condition is calculated as:

$$W_e = 4.133\ \sigma \qquad \text{(Equation 6)}$$

One then recalculates the Index of Difficulty by substituting $W_e$ for W in Equation 3, yielding an effective ID ($ID_e$). This results in an effective Index of Performance ($IP_e$) in equation 5. The $IP_e$ thus incorporates both speed and accuracy of performance.

### 4.3.2 Applications of Fitts' Law

Fitts' Law was first applied to the study of input devices by Card, English, and Burr (1978), and since then it has evolved into a precise specification for device evaluations published by the International Standards Organization (ISO). Strictly speaking, Fitts' Law, as specified in Equation 1, only deals with one-dimensional motion. The law can be extended to two-dimensional targets by taking the target width W as the extent of the target along the direction of travel (MacKenzie & Buxton, 1992). The approach angle can also have some effect on target acquisition. The ISO standard uses circular targets and a movement pattern analogous to the spokes of a wheel to keep the width W constant while averaging performance over many different approach angles (Douglas et al., 1999).

Fitts' Law has been found to apply across a remarkably diverse set of task conditions, including rate-controlled input devices (MacKenzie, 1992a), area cursors (Kabbash & Butxon, 1995), and even when pointing under a microscope (Langolf & Chaffin, 1976). A few important results for input devices are summarized below.

*Tracking versus Dragging States:* Experiments have shown that some devices exhibit much worse performance when used in the dragging state than in the tracking state (MacKenzie, Sellen & Buxton, 1991). It is important to evaluate a device in both states.

*Bandwidth of Limb Segments*: Different muscle groups exhibit different peak performance bandwidths. The general ordering is fingers > wrist > upper arm, although much depends on the exact combination of muscle groups used, the direction of movement, and the amplitude of movement (Balakrishnan & MacKenzie, 1997b; Langolf et al., 1976). For example, using the thumb and index finger together has higher bandwidth than the index finger alone (Balakrishnan et al., 1997b). A device that allows fine fingertip manipulations is said to afford a *precision grasp*, like holding a pencil, as opposed to a *power grasp*, such as the grip used to hold a suitcase handle (Mackenzie & Iberall, 1994; Zhai et al., 1996).

*Effects of Lag:* Fitts Law can be adapted to predict performance in the presence of system latency (MacKenzie et al., 1993), yielding a form of the law as follows:

$$MT = a + (b + c\ LAG)ID \qquad \text{(Equation 7)}$$

*C:D gain*: Fitts' Law predicts that gain should have no effect on performance (Jellinek et al., 1990), since for any constant gain $g$, the ratio $gA/gW = A/W$, which does not change the ID (Equation 2). However, this scale-invariant prediction is only valid within limits (Guiard, 2001; Accot et al., 2001).

*Scrolling and Multi-scale navigation*: The effectiveness of scrolling techniques varies depending on how far, and how precisely, one must scroll in a document, a problem which can be formulated as a Fitts' Law study (Hinckley & Cutrell, 2001). See Guiard, Buourgeois, Mottet & Beaudouin-Lafon (2001) for Fitts' Law in multi-scale interfaces.

### 4.4 Other Metrics and Models of Input

We have emphasized Fitts' Law research because it has found such wide influence and application. But Fitts' Law only deals with the elemental action of pointing. To go beyond pointing, other models are needed.

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

$$MT = a + b\ A/W \qquad \text{(Equation 8)}$$

For a circular tunnel with the center of the tunnel at radius R, the Steering Law becomes:

$$MT = a + b\ 2\pi R/W \qquad \text{(Equation 9)}$$

See (Accot et al., 1997) for discussion of the Steering law with arbitrary curved paths, as well as using the Steering Law to predict instantaneous velocity. Also note that the Steering Law only models successful completion of the task; errors are not considered.

*Keystroke-Level Model (KLM)*: The KLM model predicts expert performance of computer tasks (Card, Moran & Newell, 1980). One counts the elemental inputs required to complete the task, including keystrokes, homing times to acquire input devices, pauses for mental preparation, and pointing at targets. For each elemental input, a constant estimate of the average time required is substituted, yielding an overall time estimate. The model is only useful for predicting the performance of experts because it assumes error-free execution; it does not account for the searching and problem-solving behaviors of novices.

Numerous enhancements and extensions to the Keystroke-Level Model (KLM) have been devised, including GOMS (Goals, Objects, Methods, and Selection Rules). The advantage of these kind of models is that they can predict the average expert completion time for short, repetitive tasks without requiring implementation of the tasks, training end users, and evaluating their performance (Olson & Olson, 1990). See also Chapter 5, Cognitive Architectures, and Chapter 6, Modeling Humans in HCI.

## 5 Mappings: How to get the most out of an input signal

Once an input signal has been digitized and passed along to a host computer, the work of the interface designer has only begun. One still must determine how to best make use of that data to provide fast, accurate, comfortable, easy to learn, and satisfying computer interfaces.

## 5.1 Transfer Functions

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 easy design a poor transfer function that hinders performance. The choice of transfer function often depends on the type of input sensor being used. 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 commonly seen 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 often hard to learn how to use and difficult to control; a better solution is to use the position of the cursor to control the resulting position within the document, as occurs when one grabs the handle of a scrollbar. Inappropriate mappings are sometimes the best design compromise given competing demands, but should be avoided if possible. For further discussion, see Zhai (1993) and Zhai et al. (1997).

*Self-centering devices*: Rate mappings are most appropriate for force-sensing devices or other devices which return-to-center when released. This property allows the user to stop quickly by releasing the device. The formula for a nonlinear rate mapping is:

$$dx = K\ x^{\alpha} \qquad \text{(Equation 10)}$$

Where $x$ is the input signal, $dx$ is the resulting rate, $K$ is a gain factor, and $\alpha$ is the nonlinear parameter. The best values for $K$ and $\alpha$ depend on the details of the device and application, and appropriate values must be identified by experimentation or optimal search (Zhai & Milgram, 1993). It may be necessary to ignore sensor values near zero so that noise in the sensor will not cause motion. The mappings used on commercial devices can be quite complex (Rutledge, 1990).

*Motion sensing devices:* Most computer systems available today use an exponential transformation of the mouse velocity, known as an *acceleration* function, to modify the cursor response. Experimental testing of mouse acceleration functions suggest that they may not improve the overall pointing speed. Rather, they help to limit the footprint that is required to use the mouse (Jellinek et al., 1990).

*Absolute devices:* Sometimes one can temporarily break the 1:1 control-to-display mapping of absolute devices. Sears and Shneiderman (Sears et al., 1991) describe a technique for touchscreens that uses cursor feedback to allow high-resolution pointing. If the finger moves slightly, the movement of the cursor is dampened to allow fine adjustments. But if the finger continues to move, the cursor jumps to the absolute position

of the user's finger again. In this way, fine adjustments are possible but the absolute nature of the device is preserved. This technique uses the touchscreen in the cursor tracking state, so it cannot be applied if one uses the dragging state as the default (Buxton, 1990b).

## 5.2 Design Challenges for Real-Time Response

When designing the real-time response to a continuous input from the user, there are often several conflicting design challenges. From the author's experience, three common themes in such challenges are as follows:

(1) *Never throw data from the user's input away* should be the mantra of the interaction designer. The data from the input device is the only direct evidence of the user's will. However, it is sometimes necessary to ignore artifacts of devices, such as noise or nonlinearity. Removing such irregularities without ruining the expressive bandwidth of a device can be challenging. For example, filtering noise by averaging input over several samples *necessarily* adds latency to the interaction (Liang et al., 1991). Another example is using a time-out (of perhaps 40 milliseconds) to debounce switch closures; this introduces some lag, but otherwise the software might sense multiple false contacts.

(2) *Immediate Feedback*. It is usually important to provide a real-time response to the user's input with minimum latency. This is true even if the software is not yet sure what the user is trying to do. The user knows what he or she is trying to accomplish from the very start of a movement or gesture; the computer must provide some kind of

intermediate feedback so that any recognition delay does not annoy the user or interfere with performance of the task. For example, a first-in, first-out queue of recent inputs allows restoration of the cursor position if lifting one's finger from a touch tablet disturbs the cursor position, without delaying feedback of the current cursor position (Buxton, Hill & Rowley, 1985). *Speculative execution* offers another approach to deal with ambiguous inputs (Bartlett, 2000).

(3) *Make the response to the input readily perceptible*. For example, during direct manipulation of a zoom factor, providing an exponential transformation of the input makes the response uniformly controllable (Igarashi & Hinckley, 2000). Likewise, one may need to linearize the response of some sensors (e.g., Hinckley, Pierce, Sinclair & Horvitz, 2000). It is also important to avoid sudden changes that may disorient the user. For example, instead of moving objects on the screen instantaneously, animated transitions can help the user to more readily perceive changes in response to an input (Robertson et al., 1989). Limiting the derivative of movement in response to an input (Igarashi et al., 2000) is another way to avoid sudden changes.

## 6 Feedback: What happens in response to an input?

The literature often treats input to computers separately from the resulting output through visual, tactile, or auditory feedback. But human perception and action are tightly coupled in a single motor-visual feedback loop, so clearly both topics need to be considered in the

design of interaction techniques. For a discussion of audio feedback design and perception issues, refer to Chapter 11, Non-speech Auditory Output.

From the technology perspective, one can consider feedback as passive or active. Passive feedback results from physical properties of the device, such as the shape, color, or feel of buttons when they are depressed. Active feedback is under computer control. For example, the SenseAble Phantom is a stylus connected to motors by an armature; computer control of the motors presents artificial forces to the user.

## 6.1 Passive Feedback

Several types of passive feedback are relevant to the design and use of input devices:

*Passive Visual, Auditory, and Tactile Feedback*: The industrial design of a device suggests the purpose and use of a device even before a user touches it. Mechanical sounds and vibrations that result from using the device provide confirming feedback of the user's action. The shape of the device and the presence of landmarks can help users orient a device without having to look at it (Hinckley, Pausch, Goble & Kassell, 1994a; Hinckley et al., 1997).

*Proprioceptive and Kinesthetic Feedback*: These imprecise terms refer to sensations of body posture, motion, and muscle tension (Gibson, 1962; Burdea, 1996). These senses allow users to feel how they are moving an input device without looking at the device, and indeed without looking at the screen in some situations (Mine, Brooks & Sequin,

1997; Balakrishnan & Hinckley, 1999b). This may be important when the user's attention is divided between multiple tasks and devices (Fitzmaurice & Buxton, 1997). Sellen et al. (1992) report that muscular tension makes modes more salient to the user.

## 6.2 Input/Output Correspondence

Performance can be influenced by correspondences between input and output. For visual feedback, two types of input/out correspondence that should be considered are known as *perceptual structure* and *kinesthetic correspondence*.

*Perceptual Structure*: The input dimensions of a device, as well as the control dimensions of a task, can be classified as *integral* or *separable*. Jacob, Sibert, McFarlane & Mullen (1994) explore two input devices, a 3D position tracker with integral ($x$, $y$, $z$) input dimensions, and a standard 2D mouse, with ($x$, $y$) input separated from ($z$) input by holding down a mouse button. For selecting the position and size of a rectangle, the position tracker is most effective. For selecting the position and grayscale color of a rectangle, the mouse is most effective. The best performance results when the integrality or separability of the input matches that of the output.

*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.

## 6.3 Active Haptic Feedback

Haptic feedback research has sought to provide an additional channel of sensory feedback that might improve user interfaces. Haptic feedback includes force feedback (active presentation of forces to the user) and tactile feedback (active presentation of vibrotactile stimuli to the user). Here, we briefly discuss haptic feedback with an emphasis on pointing devices. Haptic feedback is popular for gaming devices, such as force feedback steering wheels and joysticks, but general-purpose pointing devices with force or tactile feedback are not yet common. See also Chapter 10, Haptic Output, and Burdea (1996).

Adding force to a mouse or stylus may impose constraints on the mechanical design, since a physical linkage is typically needed to reflect the forces. This may prevent a force feedback mouse from functioning like a traditional mouse, as it may limit range of motion or preclude clutching by lifting the device. Some devices simulate forces by increasing resistance between the mouse and the pad. One can also use a vibrotactile stimulus, such as a vibrating pin under the mouse button, or vibrating the shaft of an isometric joystick (Campbell, Zhai, May & Maglio, 1999). Combination devices have also been explored (Akamatsu & Mackenzie, 1996).

Using force or tactile feedback to provide attractive forces that pull the user towards a target, or to provide additional feedback for the boundaries of the target, has been found

to yield modest speed improvements in some target acquisition experiments, although error rates also may also increase (Akamatsu et al., 1996; MacKenzie, 1995). When multiple targets are present, as on a computer screen with many icons and menus, haptic feedback for one target may interfere with the selection of another, unless one uses techniques such as reducing the haptic forces during rapid motion (Oakley, Brewster & Gray, 2001). Finally, one should also consider whether software constraints, such as snap-to grids, are sufficient to support the user's tasks.

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

# 7 Keyboards and Text entry techniques

## 7.1 Do Keyboards Have a Future?

Keyboards and typewriters have been in use for well over 100 years (Yamada, 1980). With the advent of speech recognition technology, it is tempting to believe that this is going to change, and that keyboards will soon become irrelevant. But a recent study revealed that, when error correction is included, keyboard-mouse text entry for the English language is more than twice as fast as automatic recognition of dictated speech (Karat, Halverson, Horn & Karat, 1999). Furthermore, speaking commands can interfere

with short term memory for words (Karl, Pettey & Shneiderman, 1993), so even perfect speech recognition will be problematic for some text entry tasks. Thus keyboards will continue to be widely used for text entry tasks for many years to come.

## 7.2 Procedural Memory

Keyboards rely on the automation of skills in *procedural memory*, which refers to the human ability to perform complex sequences of practiced movements, seemingly without any cognitive effort (Anderson, 1980). Procedural memory enables touch typing on a keyboard with minimal attention when entering commonly used symbols. As a result, attention can be focused on mentally composing the words to type, and on the text appearing on the screen. Hot keys (chorded key combinations for frequently used commands) can also be learned to allow rapid selection of commands.

The automation of skills in procedural memory takes lots of practice. This can be formalized as the *power law of practice* (Anderson, 1980):

$$T = aP^b \qquad \text{(Equation 11)}$$

Where $T$ 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. The power law of practice helps explain why keyboard designs are so resistant to change. Several keyboard layouts that are faster than qwerty have been discovered, but the performance advantage typically does not merit the up-front investment required to relearn the skill of touch typing.

## 7.3 Trends in Keyboard Design

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, Potosnak & Magyar, 1997). Since these factors seem to be well understood, recent keyboard designs have stressed other means of improving overall input efficiency or user comfort. Some recent trends include the following:

*Ergonomic Design*: Split-angle keyboards may help maintain neutral posture of the wrist, and thereby help avoid ulnar deviation (Honan, Serina, Tal & D., 1995; Marklin, Simoneau & Monroe, 1997; Smutz, Serina, Bloom & Rempel, 1994). Ulnar deviation is associated with increased pressure in the carpal tunnel, which can lead to repetitive stress injuries (Putz-Anderson, 1988; Rempel et al., 1998).

*Integrated Pointing Devices*: An integrated isometric joystick or touchpad can allow pointing without having to move one's hand away from the keyboard. Such devices are useful when space is limited, or when there is no convenient flat surface for a mouse.

*Minature Keyboards*: Small foldaway keyboards are becoming popular to allow rapid text entry on small handheld devices.

*Wireless Keyboards* allow using a keyboard in one's lap or at a distance from the display.

*Extra Functionality*: Many recent keyboards include extra function keys, such as internet Forward and Back navigation. Keyboards can facilitate two-handed input by placing commonly used functions such as scrolling (Buxton et al., 1986) in a position convenient for the left hand[2] (MacKenzie & Guiard, 2001).

## 7.4 One-Handed Keyboards

Chording keyboards have a small number of keys that the user presses in tandem to enter text. Although chording keyboards take more time to learn, they can sometimes allow one to achieve higher peak performance because the fingers have minimal travel among the keys (Buxton, 1990a). It is also possible to type using just half of a standard qwerty keyboard (Mathias, MacKenzie & Buxton, 1996).

## 7.5 Soft Keyboards

Soft Keyboards, which are especially popular on mobile devices, depict keys in a graphical user interface to allow typing with a touchscreen or stylus. The design issues for soft keyboards differ tremendously from normal keyboards. Soft keyboards require significant visual attention because *the user must look at the keyboard* to coordinate the motion of the pointing device. Only one key at a time can be tapped, so much of the time is spent moving back and forth between keys (Zhai, Hunter & Smith, 2000).

---

[2] Approximately 95% of computer users hold the mouse in their right hand, since many left-handers learn to use the mouse this way. Therefore, for most users, placing functions on the left side reduces the demands on the mouse hand.

### 7.6 Character Recognition

Handwriting (even on paper, with no “recognition” involved) is much slower than skilled keyboard use, but its naturalness is appealing. Recognizing natural handwriting is difficult and error-prone for computers, although the technology continues to improve. To simplify the problem and make performance more predictable for the user, many handwriting recognizers rely on single-stroke gestures that are specially designed to be memorable (MacKenzie & Zhang, 1997), but easy for computers to recognize, such as *graffiti* for the 3COM PalmPilot.

## 8 The Future of Input

Researchers are looking to move beyond the current "WIMP" (Windows, Icons, Menus, and Pointer) interface, but there are many potential candidates for the post-WIMP interface. Nielsen (1993) suggests candidates such as virtual realities, sound and speech, pen input and gesture recognition, limited artificial intelligence, and highly portable computers. Weiser (1991) proposes the ubiquitous computing paradigm, which suggests that networked computers will increasingly become integrated with ordinary implements and that computers will be embedded everywhere in the user's environment.

Connecting the multiplicity of users, computers, and other digital artifacts through the internet and wireless networking technologies represents the foundation of the ubiquitous computing vision; indeed, it seems likely that 100 years from now the phrase “wireless

network” will seem every bit as antiquated as the phrase “horseless carriage” does today. Techniques that allow users to communicate and share information will become increasingly important. Biometric sensors or other convenient means for establishing identity will be important for security, but such technologies will also make services such as personalization of the interface and sharing data much simpler. As one early example of what is possible, the *pick and drop* technique (Rekimoto, 1997) supports cut-and-paste between multiple computers by using a stylus with a unique identifier code. From the user’s perspective one just “picks up” data from one computer and “drops” it on another, but behind the curtain the computer senses the unique ID and uses it as a reference to an internet location that stores the data. Other important input issues to effectively support multiple users working together include interaction on large work surfaces (Moran et al., 1997; Buxton, Fitzmaurice, Balakrishnan & Kurtenbach, 2000), combining large and small displays (Myers, Stiel & Gargiulo, 1998), and electronic tagging techniques for identifying objects (Want, Fishkin, Gujar & Harrison, 1999).

Because of the diversity of locations, users, and task contexts, intelligent use of sensors to acquire contextual information will also be important to realize this vision of the future (Buxton, 1995a). For example, point and shoot cameras often have just one button, but a multitude of sensors detect lighting levels, the type of film loaded, and distance to the subject, thus simplifying the technical details of capturing a good exposure. This perspective may lead to an age of ubiquitous sensors where inexpensive, special-purpose, networked sensors collect information that allows greatly simplified interaction with many devices (Saffo, 1997; Fraden, 1996). Examples include mobile devices that sense

location (Schilit, Adams & Want, 1994; Want, Hopper, Falcao & Gibbons, 1992), or that know how they are being held and moved (Harrison, Fishkin, Gujar, Mochon & Want, 1998; Hinckley et al., 2000; Schmidt, 1999).

Current computers are inexpensive enough to be *personal* or even *handheld*. In the future, computing will be all but free, so more and more tools and everyday objects will include computing power and network connectivity. Hence many computers will be specialized, application-specific devices, rather than general-purpose tools. Application-specific input devices can be tailored to suit specific users and task contexts, such as allowing surgeons to view brain scans in 3D by rotating a miniature doll's head (Hinckley et al., 1994a). *Tangible user interfaces* may allow users to employ the world as its own interface by tightly coupling atoms and bits (Ishii & Ullmer, 1997). Cameras, microphones, and other sensors may imbue computers with perceptual apparatus that can help computers to perceive the world and the activity of users (Pentland, 1999).

Advances in technology will yield new "elemental" inputs. For example, scanners and digital cameras make the input of *images* a user interface primitive. Tablets that sense multiple points of contact (Lee, Buxton & Smith, 1985), sensors that detect the pressure exerted by a user's hand in a high resolution grid of samples (Sinclair, 1997), or digital "tape" that senses twist and curvature (Balakrishnan et al., 1999a) all challenge the traditional concept of the pointing device since the interaction is no longer defined by a single point of contact. The difference is analogous to sculpting clay with the entirety of both hands rather than with the tip of a single pencil.

Another approach is to synthesize structure from low-level input (Fitzmaurice, Balakrisnan & Kurtenbach, 1999). For example, a user might receive email from a colleague saying “Let’s meet for lunch sometime next week.” Pattern recognition techniques can automatically recognize that a calendar is needed, and select appropriate dates. The fields for a meeting request can be filled in with information gleaned from the message, thus eliminating the need for many input actions that otherwise might be required of the user (Horvitz, 1999).

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. The forms and capabilities of the technologies we use will continue to advance, but we are probably stuck with the basic human senses and cognitive skills. The examples enumerated above underscore the need to have a broad view of interaction. One must consider not only traditional “pointing devices,” but also new sensors, high-dimensional input devices, and synthesis techniques that together will advance human interaction with computers.

## 9 Acknowledgements

I would like to thank Hugh McLoone for material on ergonomic device design, Gina Venolia for sharing many insights on input device design and evaluation, and Ed Cutrell and Steve Bathiche for commenting on early versions of this chapter.

## 10 References


Accot, J. & 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.

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

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

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

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

Balakrishnan, R., G. Fitzmaurice, G. Kurtenbach & K. Singh (1999a). Exploring Interactive Curve and Surface Manipulation Using a Bend and Twist Sensitive Input Strip. Proc. ACM Symposium on Interactive 3D Graphics. 111-118.

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

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

Balakrishnan, R. & I. S. MacKenzie (1997b). Performance Differences in the Fingers, Wrist, and Forearm in Computer Input Control. Proc. CHI'97 ACM Conf. on Human Factors in Computing Systems. 303-310.

Bartlett, J. F. (2000). "Rock 'n' Scroll Is Here to Stay." IEEE Computer Graphics and Applications(May/June 2000): 40-45.

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

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

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

Brooks, J., F. P. (1988). Grasping reality through illusion: interactive graphics serving science. Proceedings of CHI'88: ACM Conference on Human Factors in Computing Systems, Washington, DC, ACM, New York. 1-11.

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

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

Buxton, W. (1990a). The Pragmatics of Haptic Input. Proceedings of CHI'90: ACM Conference on Human Factors in Computing Systems, Tutorial 26 Notes, Seattle, Wash., ACM, New York.

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

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

Buxton, W. (1995b). 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.

Buxton, W. (2001). A Directory of Sources for Input Technologies. http://www.billbuxton.com/InputSources.html, Alias | Wavefront, Inc. & University of Toronto.

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

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

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

Campbell, C., S. Zhai, K. May & 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.

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

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

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

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

Conner, D., S. Snibbe, K. Herndon, D. Robbins, R. Zeleznik & A. van Dam (1992). Three-Dimensional Widgets. Computer Graphics (Proc. 1992 Symposium on Interactive 3D Graphics). 183-188, 230-231.

Douglas, S., A. Kirkpatrick & 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.

Douglas, S. & 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.

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

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

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

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

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

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

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

Greenstein, J. (1997). Pointing Devices. Handbook of Human-Computer Interaction. M. Helander, T. Landauer and P. Prabhu. Amsterdam, North-Holland**:** 1317-1348.

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

Guiard, Y. (2001). Disentangling Relative from Absolute Amplitude in Fitts' Law Experiments. Proc. ACM CHI 2001 Conf. on Human Factors in Computing Systems: Extended Abstracts. 315-316.

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

Harrison, B., K. Fishkin, A. Gujar, C. Mochon & 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.

Hinckley, K. & E. Cutrell (2001). "Quantitative Analysis of Scrolling Techniques." submitted for publication.

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

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

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

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

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

Hinckley, K., M. Sinclair, E. Hanson, R. Szeliski & 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.

Hinckley, K., J. Tullio, R. Pausch, D. Proffitt & N. Kassell (1997). Usability Analysis of 3D Rotation Techniques. Proc. ACM UIST'97 Symp. on User Interface Software and Technology, Banff, Alberta, Canada, ACM, New York. 1-10.

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

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

Igarashi, T. & K. Hinckley (2000). Speed-dependent Automatic Zooming for Browsing Large Documents. Proc. ACM UIST 2000 Symp. on User Interface Software & Technology. 139-148.

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

Jacob, R. (1991). "The Use of Eye Movements in Human-Computer Interaction Techniques: What You Look At is What You Get." ACM Transactions on Information Systems **9**(3): 152-169.

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

Jacob, R. J. K. (1996). Input Devices and Techniques. The Computer Science and Engineering Handbook. A. B. Tucker, CRC Press**:** 1494-1511.

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

Kabbash, P. & 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.

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

Karat, C., C. Halverson, D. Horn & 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.

Karl, L., M. Pettey & 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.

Keir, P., J. Back & D. Rempel (1999). "Effects of computer mouse design and task on carpal tunnel pressure." Ergonomics **42**(10): 1350-1360.

Kramer, A. (1994). Translucent Patches--Dissolving Windows. Proc. ACM UIST'94 Symp. on User Interface Software & Technology. 121-130.

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

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

Langolf, G. & D. Chaffin (1976). "An investigation of Fitts' law using a wide range of movement amplitudes." Journal of Motor Behavior **8**: 113-128.

Lee, S., W. Buxton & K. Smith (1985). A Multi-Touch Three Dimensional Touch-Sensitive Tablet. Proc. CHI'85 Conf. on Human Factors in Computing Systems. 21-25.

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

Liang, J., C. Shaw & M. Green (1991). On Temporal-Spatial Realism in the Virtual Reality Environment. Proc. ACM UIST'91 Symp. on User Interface Software & Technology. 19-25.

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

Mackenzie, C. & T. Iberall (1994). The Grasping Hand. Amsterdam, North Holland.

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

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

MacKenzie, I. S. (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.

MacKenzie, I. S. & W. Buxton (1992). Extending Fitts' law to two-dimensional tasks. Proc. ACM CHI '92 Conference on Human Factors in Computing Systems. 219-226.

MacKenzie, I. S. & 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.

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

MacKenzie, I. S., A. Sellen & 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.

MacKenzie, I. S. & C. Ware (1993). Lag as a Determinant of Human Performance in Interactive Systems. Proc. ACM INTERCHI'93 Conference on Human Factors in Computing Systems. 488-493.

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

Marklin, R., G. Simoneau & 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.

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

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

Moore, M., P. Kennedy, E. Mynatt & J. Mankoff (2001). Nudge and Shove: Frequency Thresholding for Navigation in Direct Brain-Computer Interfaces. Proc. ACM CHI 2001 Conf. on Human Factors in Computing Systems: Extended Abstracts. 361-362.

Moran, T., P. Chiu & W. van Melle (1997). Pen-Based Interaction Techniques for Organizing Material on an Electronic Whiteboard. Proc. ACM UIST'97 Symp. on User Interface Software & Technology. 45-54.

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

Nielsen, J. (1993). “Noncommand User Interfaces.” Communications of the ACM **36**(4): 83-89.

Oakley, I., S. Brewster & 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.

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

Pearson, G. & M. Weiser (1988). Exploratory Evaluation of a Planar Foot-Operated Cursor-Positioning Device. Proc. ACM CHI'88 Conference on Human Factors in Computing Systems. 13-18.

Pekelney, R. & R. Chu (1995). Design Criteria of an Ergonomic Mouse Computer Input Device. Proc. HFES Human Factors and Ergonomics Society 39th Annual Meeting. 369-373.

Pentland, A. (1999). Perceptual Intelligence. Handheld and Ubiquitous Computing (HUC'99), Springer-Verlag. 74.

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

Rekimoto, J. (1997). Pick-and-Drop: A Direct Manipulation Technique for Multiple Computer Environments. Proc. ACM UIST'97 Symp. on User Interface Software & Technology. 31-39.

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

Robertson, G. G., S. K. Card & J. D. Mackinlay (1989). The Cognitive Coprocessor Architecture for Interactive User Interfaces. Proc. UIST'89 Symposium on User Interface Software and Technology. 10-18.

Rubine, D. (1991). "Specifying Gestures by Example." Computer Graphics **25**(4): 329-337.

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

Saffo, P. (1997). Sensors: The Next Wave of Infotech Innovation. Institute for the Future: 1997 Ten-Year Forecast**:** 115-122.

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

Schmidt, A., Aidoo, K., Takaluoma, A., Tuomela, U., Van Laerhove, K., Van de Velde, W. (1999). Advanced Interaction in Context. Handheld and Ubiquitous Computing (HUC'99), Springer-Verlag. 89-101.

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

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

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

Sinclair, M. (1997). The Haptic Lens. SIGRRAPH'97 Visual Proceedings. 179.

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

Sutherland, I. E. (1968). A Head-mounted Three Dimensional Display. Proc. the Fall Joint Computer Conference. 757-764.

Want, R., K. P. Fishkin, A. Gujar & 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.

Want, R., A. Hopper, V. Falcao & J. Gibbons (1992.). “The active bage location system.” ACM Transactions on Information Systes **10**(1 (Jan.)): 91-102.

Ware, C. (1990). “Using Hand Position for Virtual Object Placement.” Visual Computer **6**(5): 245-253.

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

Wickens, C. (1992). Chapter 3: Attention in Perception and Display Space. Engineering Psychology and Human Performance. New York, HarperCollins**:** 74-115.

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

Zeleznik, R., K. Herndon & J. Hughes (1996). SKETCH: An Interface for Sketching 3D Scenes. Proceedings of SIGGRAPH 96, New Orleans, Louisiana. 163-170.

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

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

Zhai, S., W. Buxton & P. Milgram (1994). The "Silk Cursor": Investigating Transparency for 3D Target Acquisition. Proceedings of CHI'94: ACM Conference on Human Factors in Computing Systems, Boston, Mass., ACM, New York. 459-464.

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

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

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

Zhai, S., C. Morimoto & S. Ihde (1999). Manual and Gaze Input Cascaded (MAGIC) Pointing. Proc. ACM CHI'99 Conf. on Human Factors in Computing Systems. 246-253.

Zhai, S., B. A. Smith & 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.