INPUT/OUTPUT DEVICES AND INTERACTION TECHNIQUES

**Interaction Tasks, Techniques, and Devices**

A designer looks at the interaction tasks necessary for a particular application (Foley, Wallace & Chan, 1984). Interaction tasks are low-level primitive inputs required from the user, such as entering a text string or choosing a command. For each such task, the designer chooses an appropriate interaction technique. In selecting an interaction device and technique for each task in a human-computer interface, simply making an optimal choice for each task individually may lead to a poor overall design, with too many different or inconsistent types of devices or dialogues. Therefore, it is often desirable to compromise on the individual choices to reach a better overall design.

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

**Properties of Input Devices**

The breadth of input devices and displays on the market today can be completely bewildering. Fortunately, there are a number of organizing properties and principles which can help to make sense of the design space and performance issues. First, we consider continuous, manually operated pointing devices (as opposed to discrete input mechanisms such as buttons or keyboards, or other devices not operated with the hand, which we will discuss briefly later). For further insight readers may also wish to consult complete taxonomies of devices ((Buxton, 1983; Card, Mackinlay & Robertson, 1991)). As we shall see, however, it is nearly impossible to describe properties of input devices without reference to output—especially the resulting feedback on the screen—since after all input devices are only useful insofar as they support interaction techniques that allow the user to accomplish something.

**Physical property sensed**

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

**Transfer function**

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

**Number of dimensions**

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

**Pointing speed and accuracy**

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

**Input Device States**

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

**Hardware criteria**

Various other characteristics can distinguish input devices, but are perhaps less important in distinguishing the fundamental types of interaction techniques that can be supported. Engineering parameters of a device’s performance such as sampling rate, resolution, accuracy, and linearity can all influence performance. Latency is the end-to-end delay between the user’s physical movement, sensing this, and providing the ultimate system feedback to the user. Latency can be a devious problem as it is impossible to completely eliminate from system performance; latency of more than 75-100 milliseconds significantly impairs user performance for many interactive tasks (Robertson, Card & Mackinlay, 1989; MacKenzie & Ware, 1993). For vibrotactile or haptic feedback, users may be sensitive to much smaller latencies of just a few milliseconds.

inadvertent - непреднамеренный

haptic = tactile

sampling rate – частота дискретизации, частота выборки –

интервал (шаг) сетки выборки, используемый при дискретизации аналогового сигнала; определяет, сколько раз за единицу времени измеряется аналоговый сигнал для его преобразования в цифровую форму, кодирования или модуляции.

linearity – линейность