New Renaissance Institute®

Technology White Paper (Preliminary)

Rich Multi-Parameter Computer Pointing Devices

New Renaissance Institute®

P.O. Box 128
Belmont, California 94002
www.newrenaissanceinstitute.com
Email: inquiries@newrenaissanceinstitute.com

 Updated May 23, 2007

ABSTRACT

Conventional computer pointing devices, such as the mouse and touchpad, are limited to two degrees of freedom, thus providing control over only two parameters, properties, or dimensions at any one time. But for many operations the user needs to control more parameters. Today the user does so using these conventional input devices, but only at a significant cost in efficiency. The reason is that in order to control other parameters, the user must typically first use the input device to change context--i.e. to specify which other parameters are to be controlled--which introduces substantial overhead.

This document describes novel computer pointing devices that offer simultaneous control over more than two parameters. A number of examples of various types of these devices and a wide range of applications are presented. Abstract and practical advantages are considered, both in terms of models and specific examples.

Introduction

Conventional computer pointing devices, such as the mouse and trackball, are limited to two degrees of freedom. As a result, they provide control over only two independent parameters, properties or dimensions, at one time. (Note that, strictly, mouse and trackball buttons afford additional degrees of freedom. However, since they have only two states, they provide only very limited control over parameters besides the two that the position of the mouse and trackball control, and are of very limited utility. Consequently, we will for the most part set them aside here.) For example, in editing a graphics document, the user can use the mouse to drag a graphical object to the right or the left of its current location--exploiting one degree of freedom--and above or below the current location--exploiting the second degree of freedom. Thus, in this case, the user has simultaneous control over the vertical position parameter and the horizontal position parameter of the object he drags.

For many tasks, however, the user will need to adjust other parameters of the object, such as height, width, color or border thickness. This can be done using conventional pointing devices, but only at a significant cost in efficiency and ease of use. The reason is that in order to control other parameters, the user must typically first use the input device to change context--i.e. to specify which other parameters are to be controlled--which requires substantial overhead. For instance, if the user wants to change the color of the object he has been positioning, he will typically first need to use the mouse or trackball to select an item on a menu, which may then (depending on the implementation) present him with a two-dimensional array of color choices. He may then exploit the two degrees of freedom of the pointing device to make selections from the array to try out various colors. However, if he then wants to move the same object or a different object, he will typically first have to change the context again, for instance, by moving the cursor over the drawing area and clicking on the object. If he wants to change the location and color of a number of objects, or try out various locations and colors for one object, the need to repeatedly change context can become cumbersome and interfere with the task at hand.

This is a single instance of a kind of problem encountered in carrying out many different kinds of tasks in a wide range of applications--that is, because the conventional mouse and trackball provide simultaneous control over only two parameters, the user must constantly carry out operations to switch context in order to adjust different pairs of parameters. In the best possible case--as in the example just given of changing the location and color of graphical objects--the context switching only increases the number of operations the user must carry out and does not interfere with his ability to do so. But in many cases--such as adjusting the three position parameters and three orientation angles of an object in three-space--the need to constantly switch context interferes with the task and makes it impractical to carry it out.

So there is clearly a need for input devices that provide simultaneous control over more than two parameters. We will call such devices *rich multi-parameter computer pointing devices* or, for short, *rich pointing devices*. However, a device that has four, six, eight or more degrees of freedom runs the risk of being confusing to use. Thus the challenge is to provide the user with simultaneous control over more than two parameters but in a way that is intuitive and perspicuous.

The purpose of this document is to provide further motivation for rich pointing devices, to describe some examples of such devices that have been developed by New Renaissance Institute® (NRI) and to describe some applications for them. The technologies described in this whitepaper are protected by U.S. Patent 6,570,078, U.S. Patent Applications 2005/0179655, 2005/0179650, 2005/0179663, 2005/0179652, 2005/0179651, and other affiliated patents licensable from NRI®. NRI® can provide detailed hardware and software reference designs under negotiable terms. All financial or in-kind proceeds from such arrangements are used to fund pure academic research at NRI®.

Examples of Rich Multi-Parameter Computer Pointing Devices

Although the scroll wheel mouse is widely manufactured and widely available, it is rarely used to adjust three parameters simultaneously (as would be useful, for instance, to position a rendered graphical object in three-space). Nevertheless, the scroll wheel mouse does serve as an example of a computer pointing device where the user benefits from the elimination of the need to explicitly switch context between using the mouse to vertically scroll through a document and using the mouse to perform other tasks, such as selecting text. A number of other rich pointing devices have been developed through the years. Figure 1 provides a table comparing a selection of rich pointing devices that are or have been commercially available. Note in particular the Kawala Pad™, from the 1980s. It is that technology, albeit in a recast form, that is used for the pointing device in many laptop computers. Although not cited here, the reader may also find of interest, patents by Westerman and Elias [14,15], as well as the associated proximity detection products made by Finger Works™, Inc.

Figure 1 also includes (at the bottom) two kinds of rich pointing devices developed and patented by NRI®. The NRI® solutions to the challenge of creating such devices utilize two strategies:

1. quantitatively and qualitatively sensing multiple kinds of movements and contact, and interpreting these as user input information
2. combining two or more pointing device technologies to create compound pointing devices

The first strategy is utilized by the NRI® touchpad, which is disclosed in U.S Patent 6,570,078. The touchpad, which can comprise an array of pressure sensors, provides control over a large number of parameters (six or more), is very flexible, can be operated in multiple modalities, and supports a wide range of metaphors. The second strategy is utilized by the NRI® compound pointing devices, which in some cases also utilize the first strategy. These devices are disclosed in a series of pending patent applications, 2005/0179655, 2005/0179650, 2005/0179663, 2005/0179652, 2005/0179651. The touchpad and the compounding pointing devices are described in the remainder of this section. Subsequent sections describe some applications for rich pointing devices and compare their performance with the performance of conventional pointing devices.

Figure 1

USER INTERFACE TECHNOLOGY	CONTINUOUS DEGREES OF FREEDOM	DISCRETE DEGREES OF FREEDOM	METHOD	ADVANTAGES	DISADVANTAGES
TRADITIONAL MOUSE OR TRACK BALL	2 + scroll	2 to 3	MOVABLE BOX	WELL-UNDERSTOOD	* PHYSICALLY AWKWARD * NEEDS TABLE-SPACE
SPACE-BALL™	0	6 BIDIRECTIONAL DISPLACEMENTS	MOVABLE GRIP BALL ON SHAFT	LARGE # PARAMETERS	ONLY ON/OFF CONTROL
KAWALA PAD™	2	2 TYPICAL	TACTILE PAD	* NO PHYSICAL OBJECT * CAN USE STYLUS * NO GUI NEEDED	POOR RESOLUTION
3-D TRACKER™	6	0	WAND IN 3-SPACE	LARGE # PARAMETERS	WAND MUST BE WAVED IN SPACE
JOYSTICKS	2 to 3	0 to 1	MOVABLE SHAFT	NO UP NEEDED	* AWKWARD * ABSOLUTE COORDINATES * POOR RESOLUTION
TOUCH SCREEN	2 to 3	0	TRANSPARENT TACTILE PAD	* IMAGE OVERLAY * NO PHYSICAL OBJECT TO HANDLE	
STYLUS	2 to 6 or more	5 TYPICAL	SPECIALIZED	LIKE PEN & PAPER	STYLUS MUST BE USED
NRI® TOUCHPAD	6 or more	FLEXIBLE	TACTILE PAD	* NO PHYSICAL OBJECT * LARGE # OF PARAMETERS * VERY FLEXIBLE CONTROL	
NRI® COMPOUND DEVICES	4 to 8 or more	2 or more	MOUSE PLUS OTHER DEVICES	* EASE OF USE * MANY IMPLEMENTATIONS * CAN BE MODULAR	

High Parameter-Count Touchpads

The novel touchpad technology of U.S. Patent 6,570,078 employs two-dimensional pressure sensor arrays for gathering information in the form of real-time “images” of two-dimensional pixelated pressure profiles of parts of the user's hand [6,7,8,10]. The pressure distribution on the sensor surface is measured and presented to real-time image processing and recognition functions that derive values for a large number of independent parameters. These values can be used to control arbitrary external systems and applications.

Figures 2a-c illustrate possible physical appearances and configurations of the touchpad. In Figure 2a the sensor array and the supporting hardware share the same housing. In Figure 2b the sensor array and at least part of the supporting hardware are separately housed and connected by a flexible cable to permit a smaller and more portable sensor housing. In Figure 2c the supporting hardware is incorporated into a larger physical housing, which also contains the system controlled by the controller (e.g., a computer workstation [7], a robotics system [7], an electronic musical instrument [10] or a lighting control system [7,10]). Many other configurations are also possible.

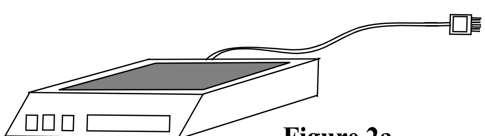

Figure 2a

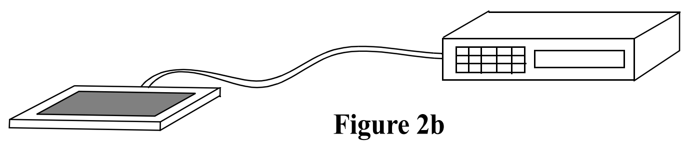

Figure 2b

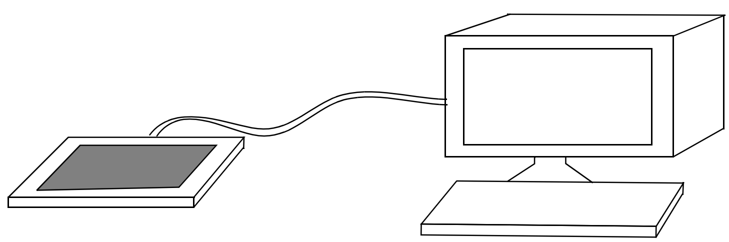

Figure 2c

To show how the touchpad works, we will consider a simple example of the acquisition of a pressure image. Assume the sensor array will be contacted by at most a single finger, as shown in Figure 3a.

The pressure profile produced by such contact is illustrated in Figure 3b.

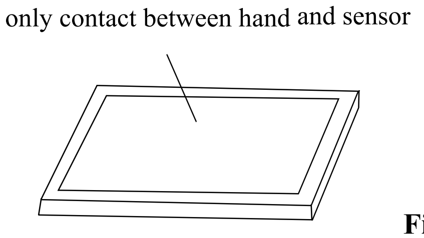

Figure 3a

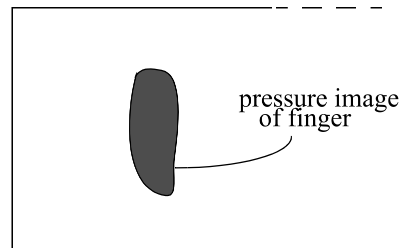

Figure 3b

The elliptical form of this image produces a measurable orientation (as suggested by the figure), which can be changed by moving the wrist. Since the finger can contact the sensor array anywhere on its surface, the geometric center of the image determines two coordinates. It is additionally possible to independently control two coordinates of the finger's "tilt," i.e., the degree to which pressure is concentrated with respect to left-right and front-back axes. The user can easily learn to keep all these parameters relatively constant while varying the average or total pressure exerted -- yet another parameter that can be controlled.

Thus, in this simple example, the user can independently control six different parameters simultaneously by making six different kinds of movements: moving the finger left and right, moving it forward and back, changing the orientation of the ellipse the finger tip creates by rotation, tilting the finger left and right, tilting the finger front and back, and varying the amount of pressure exerted on the sensor array. These various movements are illustrated in Figure 3c. By using the touchpad in more complex ways, such as accepting as input pressure from more than one finger, the user can control many more parameters simultaneously. See the NRI® whitepaper *Rich Multi-Parameter Touchpad User Interface: Background, Capabilities, and Applications* [7] for more examples and discussion. For more information about the touchpad technology and ways to implement it, see the NRI® whitepaper *Rich Multi-Parameter Touchpad User Interface: Technology and Sample Implementations* [10].

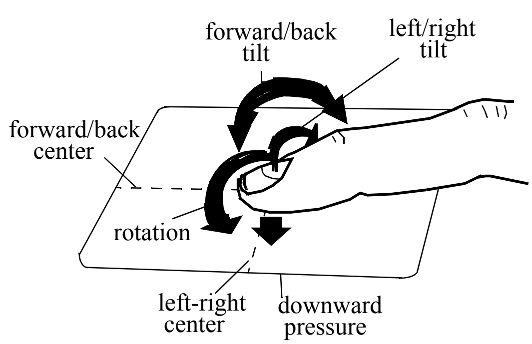

Figure 3c Six Simultaneous Parameters of Real-Time Control with a Single Fingertip

Compound Pointing Devices

In the compound pointing devices developed by NRI® [1,2,3,4,5] and others, conventional or novel pointing devices are combined to enable the user to control multiple parameters simultaneously. There are many possible implementations of such devices. A few will now be described.

Computer mice with a single scroll wheel for controlling vertical scroll bars are familiar to many computer users. Such mice enable the user to control an additional parameter besides the horizontal position and the vertical position of the cursor on the screen, namely, the position of the vertical scroll bar. In one implementation of the invention, an additional scroll wheel for controlling horizontal scroll bars is affixed to the mouse, either in front of the vertical scroll wheel (Figure 4a), or behind (Figure 4b).

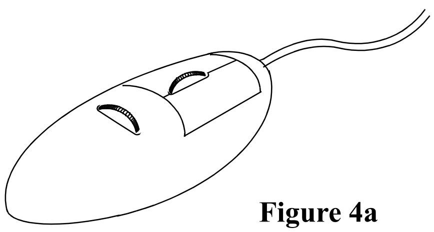

Figure 4a

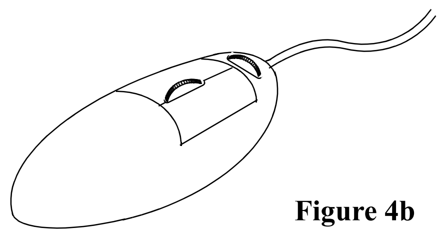

Figure 4b

In another implementation, the mouse has a single trackball mounted on top of it (Figures 5a-c). The mouse includes buttons as in conventional mice. Relatively short buttons can be affixed on either side of the trackball on the surface of the mouse (Figure 5a) or on the sides of the mouse (Figure 5b). Or long, angular buttons can be affixed on either side of the trackball to make them easier to press (Figure 5c).

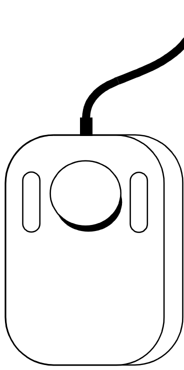

Figure 5a

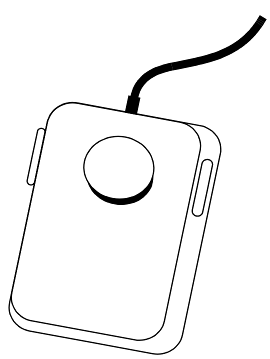

Figure 5b

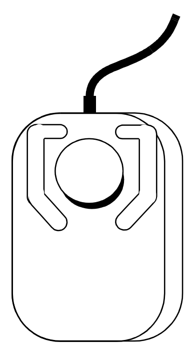

Figure 5c

When the mouse and the mounted trackball are used as in conventional devices, the mouse provides control over two parameters and the trackball provides control over two more parameters. But, by accepting as input some novel ways of moving a trackball, more parameters can be controlled. Figure 6a illustrates the conventional ways of moving a trackball, either towards-away from the user or side-to-side. But the user can also twist the trackball to the right or left (Figure 6b), press down on the trackball (Figure 6c), or displace the trackball in its housing towards or away from the user, or to the right or left (Figure 6d). These ways of moving the trackball enable the user to control four parameters, in addition to the two parameters conventional ways of moving a trackball enable the user to control. Thus, a mouse with a single trackball mounted on it can enable the user to control up to eight parameters.

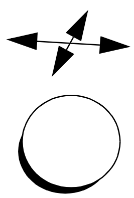

Figure 6a

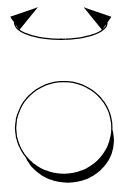

Figure 6b

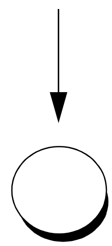

Figure 6c

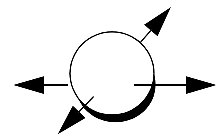

Figure 6d

In another implementation, a touchpad instead of a trackball is mounted on a mouse (Figures 7a-c). As with the trackball-mounted mouse, the mouse buttons can be positioned on the surface of the mouse on either side of the touchpad (Figure 7a) or on the sides of the mouse (Figure 7b); or long, angular buttons can be positioned on either side of the touchpad (Figure 7c). The touchpad can be used in the conventional way, so that the vertical and horizontal position of the finger on the screen are accepted as input, and the touchpad provides control over two additional parameters. But the touchpad can also be used in the novel ways described earlier, so that the orientation of the ellipse created by pressing the finger on the screen, the tilt of the finger, and so on, are also accepted as input, providing control over many more parameters.

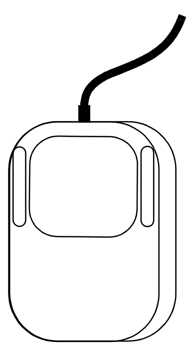

Figure 7a

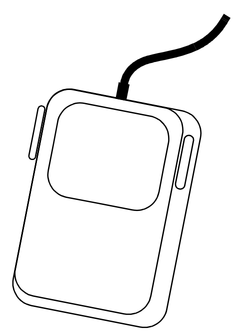

Figure 7b

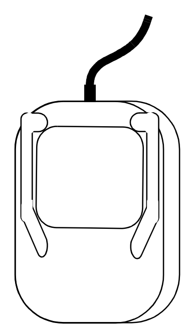

Figure 7c

The two compound pointing devices described so far combine a mouse with one other pointing device. But multiple pointing devices, in various combinations, can be mounted on the mouse. To give a few examples, the mouse can be combined with two trackballs (Figure 8a), with two touchpads (Figure 8b), a touchpad and a trackball (Figure 8c), or three trackballs and a vertical scroll wheel (Figure 8d). It should be clear that large numbers of parameters can be simultaneously controlled by using such devices, and that many other compound pointing devices can be implemented.

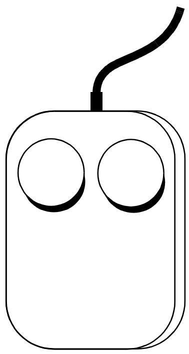

Figure 8a

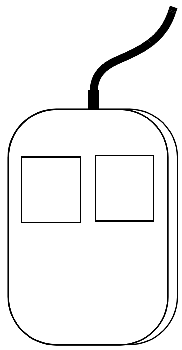

Figure 8b

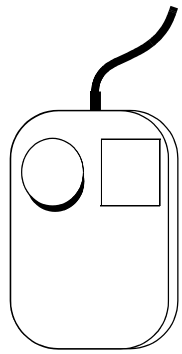

Figure 8c

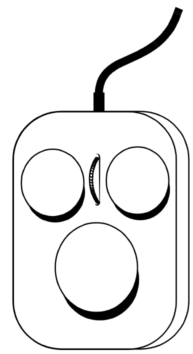

Figure 8d

Example: Traditional 2D Layout, CAD and Graphics Applications

In most contemporary 2-dimensional layout and graphics applications, such as those commonly used for viewgraphs, page layout, and electronic CAD, setting and changing object attributes require many mouse operations. Typically, a substantial proportion of these mouse operations is required because the mouse only allows for the adjustment of two parameters at a time, and the user must change context several times to adjust different parameters.

Figure 9 illustrates the creation of a viewgraph. This task includes the creation of a flowchart (here depicting a business workflow process) and as such, also illustrates requirements of 2D CAD programs for layout of diagrams of abstract objects (such as algorithms) or physical objects (such as circuits, PC boards, control panels, semiconductors, and photolithography masks). Since this example involves drawing, it also illustrates the requirements of paint-box and electronic drafting applications.

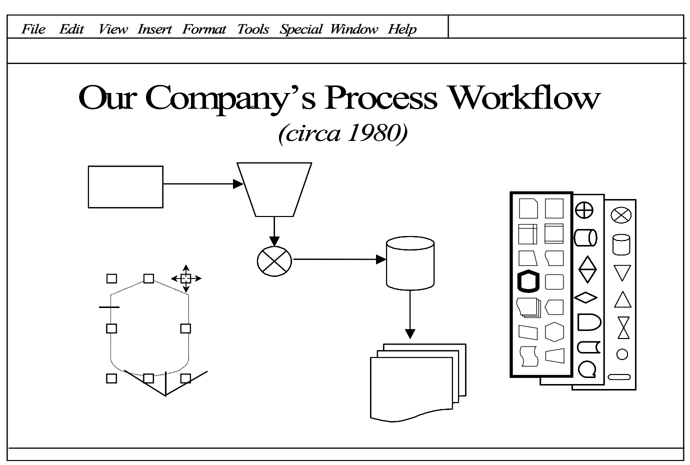

Figure 9

In this representative application, an application window is shown comprising a menu area and a drawing area. Within the drawing area, a viewgraph title and a portion of the flowchart, comprising a sequence of flowgraph objects connected by arrows, have already been created. New flowgraph objects can be added in standard fashion by selecting the type of new object from a palette, putting an instance of the object in a convenient place in the drawing area, adjusting the object's size, orientation, color, or other attributes, and putting it in its final position. Often, some of the last few steps are cycled through multiple times before the newly introduced object is adequately drawn and the user can direct attention to the next task. In this example, the palette of available objects is shown as an overlapping stack of three sub-class palettes, each providing the available objects within that sub-class. Here a sub-class palette has been selected (as indicated by a heavy line), and an object has been selected (also indicated by a heavy line) from other available objects within the selected sub-class palette. Upon selection, a specific instance of the selected object appears.

The specific instance of the object is surrounded by graphical handles, which are used for sizing and rotation (for example, by using the mouse with the ALT key held down). In conventional systems, the cursor controlled by the mouse can be moved within an object, and a mouse button depressed, to move the object to another location in the drawing area; or, as shown in Figure 9, it can be positioned over one of the graphical handles, and a mouse button depressed, to permit the mouse to adjust the horizontal and vertical dimensions of the object to change its size or aspect ratio. In some applications, the adjustment of the aspect ratio can be used to collapse the object to “zero” thickness in one of the dimensions and, by continuing the adjustment, to redraw the object in its mirror image, thus additionally providing a form of vertical and horizontal flip.

As common as such operations are, there is considerable overhead involved in repeatedly selecting pairs of parameters from larger collections of parameters and adjusting them. To see some of the advantages of the present invention, the general form of these operations will now be described in detail.

Figure 10 is a flowchart showing the operations required, in a representative case, to select and adjust one of several pairs of parameters, by using a user interface device permitting the adjustment of no more than two parameters at a time. In Figure 10, the goal is simply to adjust one pair of parameters. However, since more parameters can be adjusted than are available at any one time, the specific pair to be adjusted must first be selected. In most graphical user interface systems, this requires first using a pointing device to move a cursor to a selection area, such as a palette of objects, in a first overhead operation (Operation 1 in Figure 10) and then selecting the adjustment context (parameter pair) by, say, clicking on an object in the palette, in a second overhead operation (Operation 2). In some situations the selected parameters may immediately be adjusted (Operation 5--the goal). But in many cases the cursor must then be moved in a third overhead operation (Operation 3) from the selection area to another location, such as a drawing or typing area. It may then be possible to make the selected adjustment (Operation 5), but typically the context must be activated by, say, clicking in the drawing area in a fourth overhead operation (Operation 4). After adjusting the selected parameters (for example, by sizing a rectangle) the cycle may then be repeated for another pair of parameters. But typically the parameters must be deselected by, for example, clicking to set the final value in a fifth overhead operation (Operation 6) before the sequence can be repeated to adjust another pair of parameters.

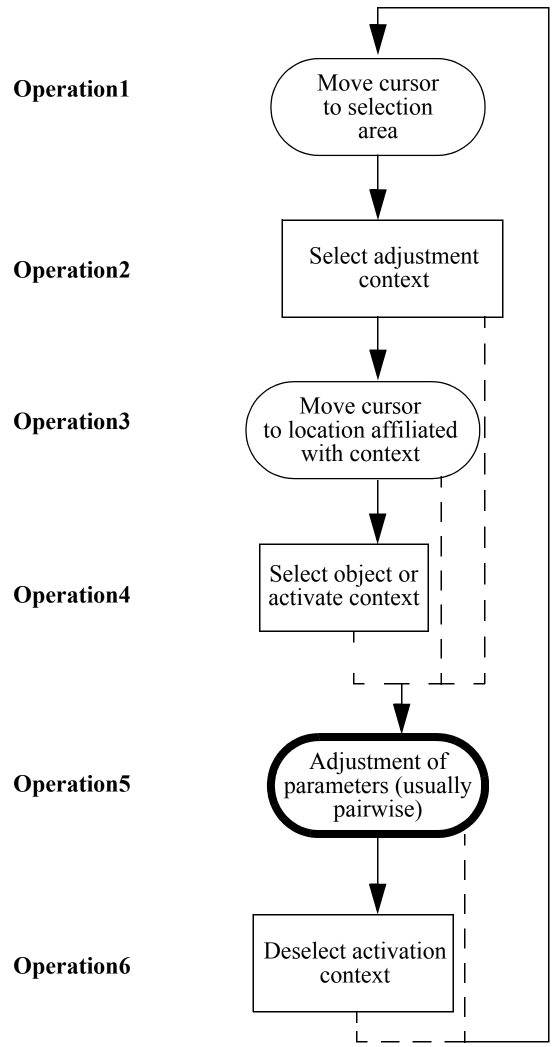

Figure 10

In summary, in order to adjust one of several pairs of parameters (Operation 5), as many as five overhead operations (two cursor movements, Operations 1 and 3, and three select/de-select clicks, Operations 2, 4 and 6) are commonly required.

Figures 11a-b illustrate the impact of the large number of overhead operations in cases where multiple pairs of parameters need to be adjusted. Figure 11a depicts the sequential adjustment of different pairs of parameters chosen from a larger group where each pair is only adjusted once. One pair of parameters is adjusted, with up to five overhead operations, in Action 1; then a second pair of parameters is adjusted, with up to five overhead operations, in Action 2; then a third pair of parameters is adjusted, with up to five overhead operations, in Action 3; and so on. Here the overhead slows things down but may not have a significant impact on achieving the overall goal of the actions.

Figure 11b depicts the adjustment of different pairs of parameters selected from a larger group where iteration is required, perhaps because how some parameters are set affects how others should be set. Here the overhead is likely to have a significant impact on achieving the overall goal of the individual pair-wise adjustments Actions 1, 2 and 3. For example, consider adjusting six parameters, two at a time, by carrying out Actions 1, 2 and 3. Not only are up to five overhead operations required for each action, but these actions must be repeated since how some parameters should be set depends on how others are set. The situation is even worse if an action to set additional parameters, Action 4, is required, as Figure 11b also illustrates. Here many more iterations are needed (represented by the arrows connecting Action 4 to the others).

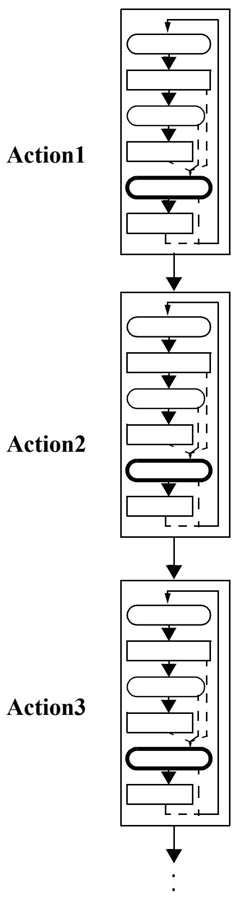

Figure 11a

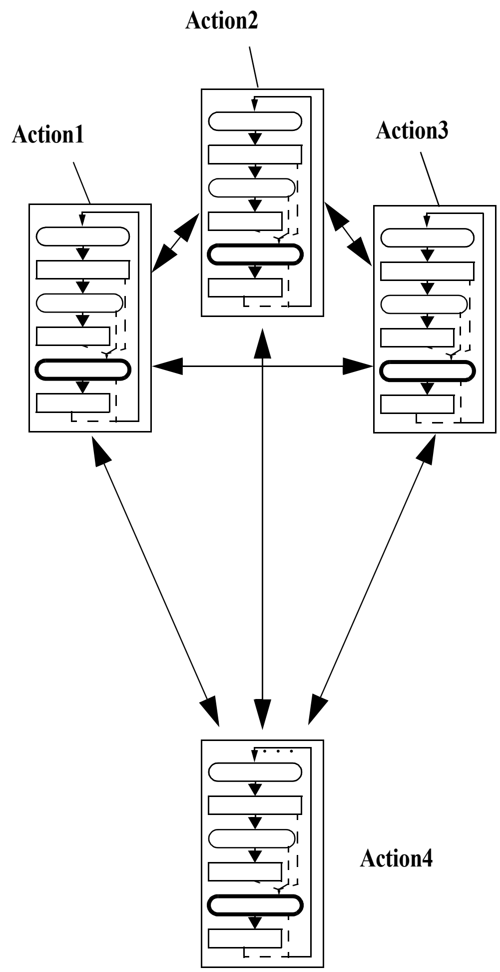

Figure 11b

Examples of Applications Utilizing Simultaneous Control Over More Than Two Or Three Parameters

In contrast to the large number of overhead operations required to adjust different pairs of parameters sequentially or iteratively when using conventional computer pointing devices, pointing devices providing simultaneous control over four, six, eight or higher numbers of parameters can eliminate many of the overhead operations depicted in Figures 10, 11a-b, and many of the iterations depicted in Figure 11b. This section considers a number of applications benefiting from such rich pointing devices. Additional applications in music, robotics, and assistive technology for the physically handicapped can be found in subsections to follow, a companion touchpad whitepaper [7] and a companion music touchpad whitepaper [9].

2D Graphics, Layout, and CAD Applications

For the operation of rich, compound pointing devices to be as efficient as possible, care must be taken in assigning operations to their constituent input devices. We will now describe a few different ways in which this might be done to carry out the graphics task described earlier in connection with Figure 9. The following operations are routinely performed in 2D graphics, layout and CAD applications:

- Selecting a palette of objects
- Selecting an object from a palette
- Selecting a "layer" for an object (common in CAD, but rare in standard drawing and paint packages)
- Positioning an object
- Sizing objects
- Rotating objects
- Setting line thickness
- Setting line color
- Setting fill color
- Setting fill pattern

One possible assignment of operations takes account of which operations are commonly used together, assigning each operation to a different constituent device, or way of operating a constituent device, to make all the operations available at the same time. (This may be application specific.) For example, since selecting an object from a palette (B) is often followed by positioning (D) and sizing (E) the object, the operations might be assigned as follows:

- Employing a four-parameter pointing device:
    - Step 1: Mouse to select object from palette (B)
    - Step 2: Mouse to position object (D)
    - Step 3: Trackball or touchpad to size object (E), saving up to five overhead operations
- Employing a six-parameter pointing device comprising a 4-parameter touchpad:
    - Step 1: Mouse for selecting object from palette (B)
    - Step 2: Touchpad finger-location to position object (D), saving up to five overhead operations

  - Step 3: Touchpad finger-tilt to size object (E),
    saving up to five overhead operations
- Employing a six-parameter version of the invention comprising a mouse and two trackballs or touchpads:
  - Step 1: Mouse for selecting object from palette (B)
  - Step 2: First trackball/touchpad to position object (D),
    saving up to five overhead operations
  - Step 3: Second trackball/touchpad to size object (E),
    saving up to five overhead operations

Note that operations B (selecting objects from a palette), D (positioning objects), and E (sizing objects), are almost always utilized; A (selecting a palette of objects), G (setting line thickness), and I (setting fill color), are frequently utilized; and C (selecting a "layer" for an object), F (rotating objects), H (setting line color), and J (setting fill pattern), are rarely utilized. As an alternative to the preceding example, we might exploit this fact in assigning operations to constituent pointing devices. For instance, it may be advantageous to assign the principal pointing device (the one on which the others are mounted) to the most frequent operations, and reserve the auxiliary pointing devices (the pointing devices mounted on the principal pointing device) to less frequent operations. The assignment of operations to the auxiliary devices can then depend on the context set by the last operation of the principal pointing device, as in the following four examples:

- Employing a four-parameter pointing device:
  - Context 1: Mouse to select object from palette (B), trackball or touchpad for selecting a palette of objects (A), and/or selecting a "layer" for an object (C), saving as many as five steps
  - Context 2: Mouse for positioning an object (D), trackball or touchpad for sizing objects (E), and/or rotating objects (F), saving as many as five steps
  - Context 3: Mouse for setting line thickness (G), trackball or touchpad for setting line color (H), saving as many as five steps
  - Context 4: Mouse for setting fill color (I), trackball or touchpad for setting fill pattern (J), saving as many as five steps
- Employing a six-parameter pointing device with a 4-parameter touchpad:
  - Context 1: Mouse for selecting an object from a palette (B), touchpad finger-location for selecting a palette of objects (A) and touchpad finger-tilt for selecting a "layer" for an object (C), saving as many as ten steps
  - Context 2: Mouse for positioning an object (D), touchpad finger-location for sizing objects (E), and touchpad finger-tilt for rotating objects (F), saving as many as ten steps
  - Context 3: Mouse for setting line thickness (G), touchpad finger-location for setting line color (H), saving as many as five steps

    - Context 4: Mouse for setting fill color (I), touchpad finger-location for setting fill pattern (J), saving as many as five steps

- Employing a six-parameter pointing device with a mouse and two trackballs or touchpads:
    - Context 1: Mouse for selecting objects from a palette (B), first trackball/-touchpad for selecting a palette of objects (A), and second trackball/touchpad for selecting a "layer" for an object (C), saving as many as ten steps
    - Context 2: Mouse for positioning an object (D), first trackball/touchpad for sizing objects (E) and second trackball/touchpad for rotating objects (F), saving as many as ten steps
    - Context 3: Mouse for setting line thickness (G), first trackball/touchpad for setting line color (H), saving as many as five steps
    - Context 4: Mouse for setting fill color (I), first trackball/touchpad for setting fill pattern (J), saving as many as five steps

As another example, the user may specify the assignment of the degrees of freedom of a pointing device to particular operations to match the task or tasks at hand. These assignments may be stored, assigned names, and later retrieved. The assignments may be saved with specific files or applications, or used as a general template for a number of applications. User-specified assignments may be particularly useful to prevent hand fatigue and injury, and to compensate for temporary or permanent disabilities.

Double-Scrollbar Application

As mentioned, contemporary mice often have a scroll wheel for adjusting the vertical scroll bar of a window. In one application of the invention, the left-right sensing capability of a trackball or touchpad affixed to a mouse is used to add an analogous capability for adjusting horizontal scroll bars.

In an implementation in which one trackball is mounted on a mouse, as shown in Figure 12, a user can move the vertical scroll bar up by rotating the trackball away from him/herself, and the vertical scroll bar down by rotating the trackball towards him/herself. By rotating the trackball to the left, the user can move the horizontal scroll bar left. Similarly, the user can move the horizontal bar right by rotating the trackball to the right.

In an implementation in which one touchpad is mounted on a mouse, as shown in Figures 5a-c, the user can move the vertical scroll bar up by sliding a finger away from her/himself, and the scroll bar down by sliding a finger towards her/himself. Similarly, the user can move the scroll bar left by sliding a finger to the left, and the scroll bar right by sliding a finger to the right.

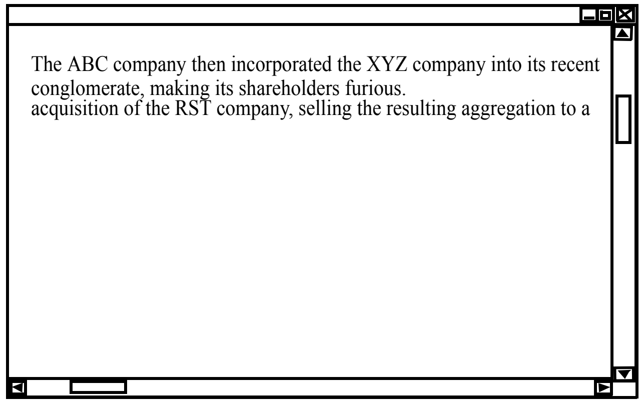

Figure 12

In another implementation, the vertical and horizontal scroll bars are adjusted with a conventional scroll wheel mouse that has been fitted with an additional scroll wheel. Figures 4a-b depict examples of such implementations. They comprise the usual components of a scroll wheel mouse, including a mouse body, buttons and a conventional scroll wheel. These implementations also include an additional scroll wheel whose direction of rotation is perpendicular to that of the first scroll wheel. Figure 4a illustrates an implementation in which the additional scroll wheel is located closer to the user, while Figure 4b illustrates an implementation in which the additional scroll wheel is located farther away. In both implementations, the two scroll wheels are located on the midline of the mouse body. To make simultaneous adjustments, however, it may be advantageous to locate the additional scroll wheel to one side of the midline. To support both left-handed and right-handed users, it may be useful to include two additional scroll wheels, one on either side of the midline; this scroll wheel configuration may also have other applications.

These novel pointing devices greatly reduce the number of operations required to move the horizontal scroll bar. To move the horizontal scrollbar with a conventional pointing device, the user must move the mouse to a position over the scrollbar, depress a mouse button, and then drag the scrollbar. In contrast, moving the horizontal scrollbar using the novel pointing devices requires only a single operation: moving the horizontal scroll wheel on the multiple scroll wheel mouse, moving a finger horizontally on the touchpad, or rolling the trackball to the right or left.

Multi-Resolution Pointing and Data Entry

For most applications that use graphical user interfaces, it is advantageous to have multiple scales of resolution for pointing device operations, such as navigating through documents, selecting text and manipulating graphical objects. For example, it would be useful to have available a finer resolution than the normal one to edit text where small fonts are used or in thumbnail overviews. Similarly, in graphics work, a fine-resolution setting would be useful to make fine adjustments to figures. For reasons like these, many windowing systems provide an "acceleration" setting that changes the pointing values on a more significant scale than is normally used so that documents can be scrolled through more quickly, and many applications change the resolution as the visual display is zoomed in and out.

In another family of applications for compound pointing devices, one constituent pointing device (for example, the mouse body) is used for either coarse or fine adjustment, while another constituent pointing device (for example, a trackball or touchpad mounted on the mouse) is used for the remaining level. In making fine adjustments to figures, it may further be advantageous to use the constituent pointing devices together with snap-grids. Each constituent device can be associated with a different snap-grid, and the spacing of each grid can correspond to the resolution of the corresponding device. This would be particularly useful where the spacing of one grid is an integer multiple of the spacing of the other. A useful extension of this would be to impose local grid spacings on individual graphical objects, particularly objects which have been resized so that the standard snap-grid spacing is no longer useful.

In a further application, the pointing devices can be used to make non-positional adjustments, such as changing the angle of rotation or the color of a graphic object. Here multiple resolutions may be useful for making fine adjustments and coarse adjustments as needed. Similarly, the adjustment of scroll bars for long documents may also benefit from rapid access to multiple resolution scales. For example, one constituent of a compound pointing device can be used to navigate within a page (using a fine-grained scale), while a second constituent can be used to navigate across pages (using a coarse-grained scale).

Provision for Both Absolute and Relative Positioning

Some types of pointing devices, such as the touchpad and X-Y joystick, have a limited range of operation, with definite minimum and maximum input values. Others, such as the mouse, trackball, and scroll wheel have an unlimited range of operation. Signals from most pointing devices are interpreted in relative terms. That is, a signal from a device is interpreted as a command to move a cursor, scroll bar, etc. incrementally in some direction relative to the current position. But signals from any of these types of devices may be interpreted as either relative or absolute quantities.

With the compound pointing devices described in this document, one constituent pointing device can be used to specify absolute positions for cursors, scroll bars, etc. (or to specify absolute quantities for other parameters) while another constituent pointing device can be used for the more familiar operation of specifying relative quantities for such parameters. For example, a mouse body or trackball can be used to specify relative positions for a scroll bar, and the touchpad to specify absolute positions, where the extreme positions of the scroll bar correspond to edges of the touchpad.

Where the relative and absolute pointing devices control the same parameters, they may adjust them with different resolutions. Further, at least one of the resolution scale factors may be adjusted automatically. For example, in a document editor, as the number of pages of the document varies, the resolution of the absolute positioning sensor may correspondingly vary, so that, the top of the touchpad corresponds to the top of the first page, and the bottom to the end of the last page. But the relative positioning sensor may retain the same vertical scrolling resolution scale regardless of the number of pages.

Color-Selection Application

To specify a color, three parameters are typically used (RGB, HSB, YUV, etc.). Since the present invention makes it possible to adjust at least four parameters simultaneously, all three color parameters can be adjusted simultaneously rather than two at a time, as with conventional pointing devices. So the invention is useful for selecting colors from a complete color space. Further, if one constituent of a compound pointing device alone provides simultaneous control over three parameters, it can be used to adjust the color or other attributes of selected objects, while another constituent (for example, a mouse body) can be used to select objects.

Multi-Level Graphic Object Grouping and Editing Application

In many drawing applications, lower-level graphical objects (such as lines or basic shapes) and other objects (such as text areas) can be grouped to form aggregate or "grouped" objects. These grouped objects can then be moved, rotated, flipped, resized, etc. as if they were lower-level, unitary objects. Grouping can typically be done either hierarchically or in mixed hierarchies -- i.e., lower-level objects are first grouped, and the resulting arranged object is then placed with other grouped or lower-level objects.

Often, some of the lower-level objects making up a grouped object will need modification. Most applications permit modifications to be made to text in individual text objects within a grouped object. However, to modify lower-level graphical objects making up a grouped object, they must first be disaggregated or "ungrouped." After modification, the grouping must be reconstructed. Often this is cumbersome, particularly where some adjustments within the group are made in response to other adjustments made within the group.

The simultaneous control of more than two parameters made possible by the invention may be advantageously applied to this problem. For example, one constituent of a compound pointing device can be used to select one level of grouping, and another constituent can be used to perform operations on (lower-level or grouped) objects within the selected level.

Figure 13a shows a portion of a larger drawing, which features a rectangle, two arrows, and a grouped object comprising two triangles connected by a line.

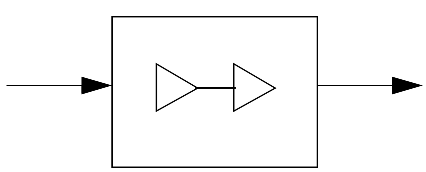

Figure 13a

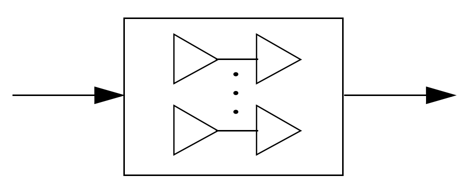

Figure 13b

The grouped object is grouped with the rectangle to form a second grouped object, and this second grouped object is grouped with the two arrows to form a third grouped object. The user's task is to modify Figure 13a to create Figure 13b. To do this, the user must do the following:

1. Move the first grouped object from its original position in the middle of the rectangle, as shown in Figure 13a, to its new position closer to the top of the rectangle, as shown in Figure 13b

2. Copy the first grouped object and position the copy towards the bottom of the rectangle
3. Create a vertically-oriented ellipsis and position it within the box (typically, a vertically-oriented ellipsis is either rotated text or a grouped object created from three aligned text elements)

With this invention, one constituent of a compound pointing device can be used to select the first grouped object, move it and copy it, without requiring any ungrouping operations. Other constituents can be used to select and perform operations on the second grouped object, the third grouped object, or the lowest-level objects, such as the objects the first grouped object comprises. Of course, the same modifications can be made with a conventional pointing device. However, such devices must be used first to ungroup the third grouped object and then to ungroup the second grouped object before the first grouped object can be selected, moved, and copied. To make subsequent modifications at a different level of grouping, the grouping would have to be reconstructed, thus adding overhead, as depicted in Figures 10 and 11a-b. Although grouped objects, can be modified in this way, where only two parameters can be adjusted at a time, having an additional number of simultaneously controllable parameters is clearly more efficient, as many or all of the overhead operations can be eliminated.

3D Graphic Object Placement and Orientation Application

CAD and drawing packages that allow 3D object positioning and orientation within 3D space, extend the capabilities of traditional 2D layout, CAD, and graphics applications. To position and orient a 3D object in 3D space requires the user to specify one additional position coordinate and two additional orientation angles beyond what must be specified to position and orient a 2D object in 2D space. This situation, in which a total of three position coordinates and three orientation angles must be specified, is illustrated in Figure 14.

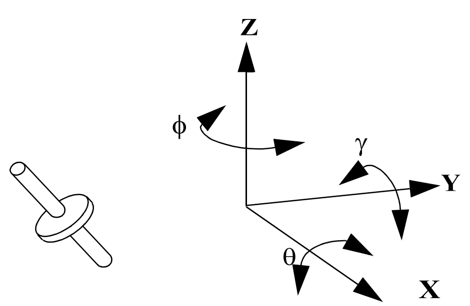

Figure 14

To adjust these parameters pair-wise, requires complex, repetitive passes of high-overhead operations. The necessity of making many high-overhead operations can be disruptive as well as slow and inefficient. The ability to adjust all three position coordinates and all three orientation angles without the overhead of context switching is thus of extremely high value.

There is a wide range of possible mappings between, on the one hand, the six position and orientation parameters of an object in a 3D space, and, on the other hand, the multiple degrees of freedom of rich pointing devices. For example, a mouse fitted with two trackballs (Figure 8a) can be used to adjust these six parameters in many different ways. One possibility is to use the position of the mouse body to control two position coordinates, one trackball to control the two orientation angles corresponding to these two coordinates, and the remaining trackball to control the position relative to the remaining position coordinate and the corresponding orientation angle. Thus, in this example, the trackballs are configured to control two parameters.

In another example, the mouse of Figure 8a, which is fitted with two trackballs, is used. Each trackball is configured to control three parameters. One of the trackballs is used for controlling the three position coordinates, and the other is used to control the three corresponding orientation angles. The position of the mouse body can be used for other drawing operations.

In yet another example, a touchpad is configured to control four parameters, two using finger position and two using finger tilt. The touchpad can be combined with a trackball controlling two parameters. Finger position can be used to control two position coordinates, finger tilt to control the two orientation angles corresponding to the two position coordinates, and the trackball to control the remaining position coordinate and the corresponding orientation angle. If the configuration includes a mouse body, its position can be used to control other drawing operations.

In a final example, a mouse with three mounted trackballs, like that of Figure 8d, is used. One trackball, at the upper left, is used to control one position coordinate and the corresponding orientation angle. Another trackball, at the upper right, is used to control another position coordinate and the corresponding orientation angle. And the third trackball, below the first two, is used to control the remaining position coordinate and the corresponding orientation angle. A right-handed user can control the upper left trackball with the thumb, index finger and middle finger; the upper right trackball with the thumb, ring finger and little finger; and the lower trackball with the palm.

Multiple Cursors and Cutting-and-Pasting

In another application, multiple pairs of degrees of freedom of a rich pointing device are used to control multiple cursors. Each cursor can be used independently of the others to perform such tasks as editing text and graphics. As a result, multiple cursors can be used to simultaneously make multiple selections of text or graphical objects and specify multiple insertion points, and to perform other, similar operations. There are many possible ways to use multiple, independent cursors. A few of these possibilities will be illustrated in an example of cut-and-paste editing.

Cut, copy and paste operations using conventional pointing devices typically require multiple operations to switch between contexts, which requires considerable overhead. Figure 15a illustrates an example in which the highlighted text has been selected so that it can be moved to a new location. To perform an operation like this with a conventional two-parameter pointing device, the cursor is first used to select the text and then used to select the insertion point.

When writing or editing it is often the case that material needs to be fetched from elsewhere and put where one is currently writing. In conventional systems, the cursor is initially positioned where the text is to be inserted. The user then loses the cursor position when she searches for, selects, and cuts or copies the text to be inserted. The user must then search for and re-select the original location, which can take considerable time.

There are also many situations where material must be divided up and distributed in a number of far-flung places. In such cases, the cursor is initially positioned where the material to be divided is located. The user must repeatedly select a portion of the material and search for its new location, each time losing the original cursor position. After the user inserts the text, it is necessary search again for the original location and reposition the cursor there. Thus, to perform this kind of task with a conventional pointing device, many pointer operations are required to switch between different locations in a document and to re-establish the same cursor locations.

In both of these cases it would clearly be advantageous if the user could "bookmark" the initial cursor location, search for the other location, perform the desired fetch or relocation operation, and then return without searching to the bookmarked location. The invention can accomplish this in a simple implementation in which the user can simultaneously control two pairs of parameters by means of a pointing device with four degrees of freedom. In the example illustrated in Figure 15a, two degrees of freedom have been used to specify the location of one cursor, which has been used to select text, while the other two degrees of freedom have been used to specify the location of a second cursor, which has been used to set an insertion point. The user has then performed a cut-and-paste operation with a single mouse operation, inserting the selected text at the location specified by the second cursor, as shown in Figure 15b. Either cursor can now be used to select more text to be cut or copied, which is inserted at the location specified by the other cursor.

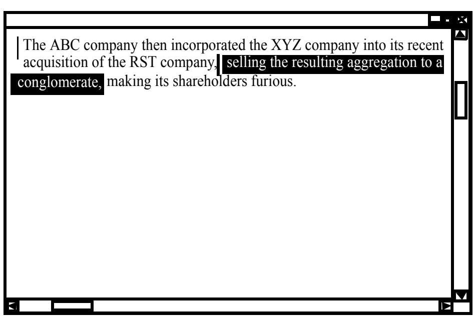

Figure 15a

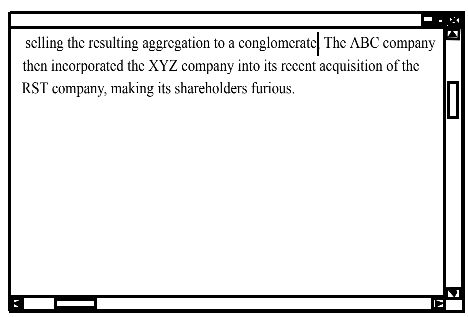

Figure 15b

Although in this example the two cursor locations were close enough to be displayed in the same window, the value of this application of the invention is significantly increased if the two locations are separated by many pages, many tens of pages, or even many hundreds of pages. Such situations can be handled using any number of approaches. In one approach, involving a single display window, the area surrounding the cursor whose corresponding pointing device was last manipulated, is displayed in the window. In another approach, also involving a single display window, a click or other user input can be used to toggle among the various cursor locations. In yet another approach, two windows are displayed. One cursor is displayed and operable within one window, and a second cursor is displayed and operable in a second window.

These same general principles can be applied to other kinds of applications, such as spreadsheets, drawing programs and CAD.

Simulation, Processing, and Analysis Applications

Simulation, processing and analysis applications typically require the adjustment of large number of parameters to model or affect the behavior of complex systems. Conventional two-parameter pointing devices require the user to adjust these parameters two at a time, which requires repetitive passes of complex, high-overhead operations. As in the case of positioning and orienting 3D objects, having to constantly use a pointing device to change context not only makes carrying out such tasks slow and inefficient, but can interfere with the user's ability to do so. Thus the capability that rich pointing devices provide to adjust more than two parameters simultaneously is of extremely high value in these applications too.

Live Signal Processing and Lighting Applications

In applications intended for artistic performance, composition, recording, mixing, video and lighting control, the simultaneous control of large numbers of parameters is frequently required. Conventional recording, mixing, video and light control consoles typically have large numbers of controls with carefully designed spatial layouts to facilitate the rapid and precise adjustment of multiple parameters. These controls include knobs, sliders, pushbuttons, and toggle switches. The introduction of graphical user interfaces has added considerable value and many new capabilities, including "soft" reconfigurable consoles and programmable functional assignments. But these interfaces typically encumber users -- accustomed to rapid and precise operation of multiple parameters -- with the overhead of repeated context-switching operations that two-parameter pointing devices require. Again, having to constantly use a pointing device to change context not only makes carrying out these tasks slow and inefficient, but can interfere with the user's ability to do so.

The ability to adjust larger collections of parameters simultaneously is thus of extremely high value in this family of applications too. This is so important that new generations of generalized hardware "control surfaces" providing configurable sliders, switches, buttons, etc. have begun to appear (see, for example, [13]). Since rich pointing devices provide simultaneous control of large numbers of parameters, they provide an alternative to conventional pointing devices, on the one hand, and to more complicated and expensive "control surface" hardware technologies, on the other.

The present invention can be specialized to include additional kinds of controls particularly well-suited to these applications. For example, as shown in Figures 16a-b, the top surface of a mouse can be outfitted with finger-operated sliders, a row of narrow touchpads, or other types of controls used in conventional control consoles, in addition to or in place of the usual mouse buttons. As illustrated in Figure 16c, the multiple individual controls on top of the mice of Figures 16a-b can alternatively be implemented with a single touchpad. To do so, the touchpad would be implemented to process in parallel the input provided by different fingers moving along each finger slot, and to produce separate control signals associated with each slot. Implementing the individual controls in this way would result in a more adaptable pointing device with a wider range of applications, like the pointing devices shown in Figures 7a-c. If the controls are implemented in this way, it may be advantageous to add an overlay bezel with finger slots and graduation markings like that shown in Figure 16d.

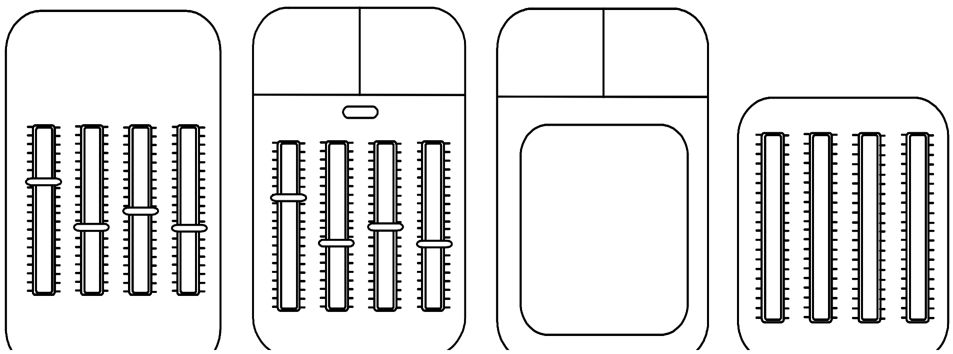

Figure 16 (a) (b) (c) (d)

Real-Time Machine Control and Plant Operations

Like the preceding applications, real-time machine control and plant (manufacturing, chemical, energy, etc.) operations require the simultaneous adjustment of large numbers of parameters. Conventional real-time machine control and plant operation consoles typically have large numbers of controls with carefully designed spatial layouts to facilitate the rapid and precise adjustment of multiple parameters. As in the case of the preceding applications, graphical user interfaces provide many benefits, but using them with conventional pointing devices can be slow, inefficient and functionally disruptive because of the high number of overhead operations. The ability to adjust larger collections of parameters simultaneously is thus of extremely high value in machine control and plant operations too.

Examples of these applications include robotics control, computer-controlled manufacturing tools, industrial optical and electron microscopy, camera control (pan, tilt, zoom, focus, aperture) and plant process elements (heaters, pumps, valves, stirrers, aerators, actuators, activators, etc.).

Zoom Control

In complex drawing, layout, and other visual-interface applications, it is often necessary to zoom in and out repeatedly to make adjustments at various levels of detail. Switching between drawing/layout modes and zoom adjustment modes typically requires many overhead operations. The simultaneous control of large numbers of parameters provided by rich pointing devices can solve this problem by eliminating the need to use pointer operations to switch modes. This can be done in a number of ways. One possible approach is to use one constituent of a compound pointing device to control the zoom level of an image, diagram, etc. while the other device is used for drawing and layout.

Another possible approach involves displaying the same image simultaneously in two different windows, one providing a macroscopic view and the other providing a microscopic view. One constituent of a compound pointing device is used to modify the drawing in the macroscopic view while another constituent device is used to modify the drawing in the microscopic view, with an adjustment resolution appropriate for each level of detail. Changes made at one level of detail are reflected in both views. (This is similar to the multiple resolution capabilities just described, but here each constituent device is associated with a different view.) The relative sizes of the macroscopic and microscopic views can be fixed or adjustable. To take this approach with a conventional mouse would require the overhead of moving the mouse and clicking on a window to switch between the microscopic and the macroscopic views.

Interactive Document Style Adjustment

In desktop publishing, website design, and other applications, style sheets, master pages, macros, etc., are used to ensure uniformity of font sizes, font styles, page margins, backgrounds, borders, indentations, list formats, paragraph formats, line widths, figure captions, figure borders, multi-column formats, etc. These style specifications may need to be changed as the user works. Here too it can be advantageous to have pointing devices that provide simultaneous control over large numbers of parameters. For instance, one component of a compound pointing device can be used to change font characteristics while another component is used for editing. In this way, style specifications can be changed as documents are edited without extensive context switching.

Relief and Prevention of Fatigue and Injury to the Hand, Wrist, and Arm

Severe injuries to the wrist, hand and arms from extensive mouse use are increasingly prevalent. Since the present invention provides many different configurations of pointing devices, it is well suited for preventing and relieving fatigue and injury due to overuse. In one possible implementation, the same user input can be provided either by moving a mouse body or by using a mounted trackball or touchpad. Thus, a user with a tiring hand can change the pointing device depending on how his hand feels, or on the nature of the current task. In addition, the user can switch back and forth between moving the mouse body and using the trackball or touchpad, either by feeling, or by following auditory or visual prompts from a time or usage monitor. Further, since rich pointing devices can greatly reduce the number of operations needed to perform a given task by reducing or eliminating context switching operations, they can save considerable wear and tear on a user's body.

Concluding Remarks

This document describes some implementations of novel computer pointing devices that provide simultaneous control over more than two parameters. Such devices are of value in a wide range of applications, some of which have been described in this document.

REFERENCES AND ON-LINE LINKS

[1] NRI U.S. Patent Application Publication 2005/0179655, "Electronic Document Editing Employing Multiple Cursors," Aug, 18, 2005.
(PDF available at http://www.newrenaissanceinstitute.com/PreGrant-PDF/Electronic-Cursors.pdf).

[2] NRI U.S. Patent Application Publication 2005/0179650, "Extended Parameter-Set Mouse-Based User Interface Device Offering Offset, Warping, and Mixed-Reference Features," Aug, 18, 2005. (PDF available at http://www.newrenaissanceinstitute.com/PreGrant-PDF/ExtendedFeatures.pdf).

[3] NRI U.S. Patent Application Publication 2005/0179663, "Freely Rotating Trackball Providing Additional Control Parameter Modalities," Aug, 18, 2005.
(PDF available at http://www.newrenaissanceinstitute.com/PreGrant-PDF/FreelyModalities.pdf).

[4] NRI U.S. Patent Application Publication 2005/0179652, "Mouse-Based User Interface Device Employing User-Removable Modules," Aug, 18, 2005.
(PDF available at http://www.newrenaissanceinstitute.com/PreGrant-PDF/MouseModules.pdf).

[5] NRI U.S. Patent Application Publication 2005/0179651, "Mouse-Based User Interface Device Providing Multiple Parameters and Modalities," Aug, 18, 2005.
(PDF available at http://www.newrenaissanceinstitute.com/PreGrant-PDF/MouseModalities.pdf).

[6] NRI U.S. Patent 6,570,078, "Tactile, visual, and array controllers for real-time control of music signal processing, mixing, video, and lighting," May 27, 2003 (PDF available at http://www.newrenaissanceinstitute.com/Issued.Patents-PDF/Tactile_Visual_and_Array.pdf).

[7] NRI Technology Whitepaper, "Rich Multi-Parameter Touchpad User Interface: Overview: Background, Capabilities, and Application," 2004.
(PDF available at www.newrenaissanceinstitute.com/Whitepaper-PDF/Touchpad.Overview.pdf).

[8] NRI. "Rich Multi-Parameter Touchpad User Interface: Modular Implementations." 2004. (PDF available at www.newrenaissanceinstitute.com)

[9] NRI Technology Whitepaper, "Rich Multi-Parameter Touchpad User Interface: Music Applications," 2004.
(PDF available at http://www.newrenaissanceinstitute.com/Whitepaper-PDF/Touchpad.Music.pdf).

[10] NRI. "Rich Multi-Parameter Touchpad User Interface: Technology and Sample Implementations." 2004. (PDF available at www.newrenaissanceinstitute.com)

[11] NRI. "Touchpad Spreadsheet Model Software."
(available by special arrangement; contact NRI for information).

[12] NRI. "Touchpad Spreadsheet Model User's Guide."
(available by special arrangement; contact NRI for information)

[13] Shrock, R. "High-End Control Surfaces." *Electronic Musician*, February 2002, pp. 72-80.

[14] Westerman, W. and Elias, J. "Method and Apparatus for Integrating Manual Input." U.S. Patent 6,323,846, November 27, 2001.

[15] Westerman, W. and Elias, J. "Multitouch System and Method for Emulating Modifier Keys via Fingertip Chords." U.S. Patent 6,570,557, November 27, 2001.

