# A THREE-STATE MODEL OF GRAPHICAL INPUT[*+1]

William A.S. BUXTON

Computer Systems Research Institute, University of Toronto, Toronto, Ontario, Canada M5S 1A4

A model to help characterize graphical input is presented. It is a refinement of a model first introduced by Buxton, Hill and Rowley (1985). The importance of the model is that it can characterize both many of the demands of interactive transactions, and many of the capabilities of input transducers. Hence, it provides a simple and usable means to aid finding a match between the two.

After an introduction, an overview of approaches to categorizing input is presented. The model is then described and discussed in terms of a number of different input technologies and techniques.

## 1. INTRODUCTION

All input devices are not created equal in their capabilities, nor input techniques (such as pointing, dragging, and rubber-banding) in their demands. The variation in each is both qualitative (types of signals) and quantitative (amount of signal, or bandwidth). Because of this variation, it is important that designers be able to answer questions like, "What particular demands does dragging make of a transducer, compared to selection?" or "How well are these task requirements met by a touch screen or a trackball?" Unfortunately, there is little available in the way of help from either theory or guidelines.

If language is a tool of thought, then a start to addressing this shortcoming is to develop a vocabulary common to both devices and techniques, and which augments our ability to recognize and explore relationships between the two.

In the remainder of this paper, a model which contributes to such a vocabulary is presented. It is a simple state-transition model which elucidates a number of properties of both devices and techniques.

## 2. TAXONOMIES OF INPUT

Traditionally, input devices have been discussed in terms of their mechanical and electrical properties (Foley & Van Dam, 1982; Sherr, 1988). Discussions centre around "joysticks," "trackballs," and "mice," for example.

---

[*] The research reported has been sponsored by the Natural Sciences and Engineering Research Council of Canada and Xerox PARC.

[+] Citation: Buxton, W. (1990). A three state model of graphical input. In D. Diaper et al. (Eds), *Human-Computer Interaction - INTERACT '90.* Amsterdam: Elsevier Science Publishers B.V. (North-Holland), 449-456.

[1] In the published version of this paper my scholarship was lacking in that I missed a very important reference, that of Newman (1968). I have added the reference for now, and will integrate a discussion of that classic paper to my text in order to place the two in proper context.

Several studies have attempted to evaluate the technologies from the perspective of human performance. Many of these are summarized in Greenstein and Arnaut (1988) and Milner (1988). A common problem with such studies, however, is that they are often overly device-specific. While they may say something about a particular device in a particular task, many do not contribute significantly to the development of a general model of human performance. (There are exceptions, of course, such as Card, English and Burr, 1978.)

With the objective of isolating more fundamental issues, some researchers have attempted to categorize input technologies and/or techniques along dimensions more meaningful than simply "joystick" or "trackball." The underlying assumption in such efforts is that better abstractions can lead us from phenomenological descriptions to more general models, and hence better analogies.

An early development in this regard was the concept of *logical devices* (GSPC, 1977, 1979). This occurred in the effort to evolve device-independent graphics. The object was to provide standardized subroutines that sat between physical input devices and applications. By imposing this intermediate layer, applications could be written independent of the idiosyncrasies of specific devices. The result was more standardized, general, maintainable, and portable code.

As seen by the application, logical devices were defined in terms of the type of data that they provided. There was one logical device for what the designers felt was each generic class of input. Consequently, we had the beginnings of a taxonomy based on use. The devices included a *pick* (for selecting objects), a *locator,* and *text.*

Foley, Wallace, and Chan (1984) took the notion of logical devices, and cast them more in the human perspective than that of the application software. They identified six generic transactions (which were more-or-less the counterparts of the GSPC logical devices) that reflected the user's intentions:

- *select* an object
- *position* an object in 1, 2, 3 or more dimensions;
- *orient* an object in 1, 2, 3 or more dimensions;
- *ink,* i.e., draw a line;
- *text,* i.e., enter text;
- *value*, i.e., specify a scalar value.

They then proceeded to enumerate a relatively comprehensive set of techniques and technologies capable of articulating each of these basic primitives.

Buxton (1983) introduced a taxonomy of input devices that was more rooted in the human motor/sensory system. The concern in this case was the ability of various transducers to capture the human gesture appropriate for articulating particular intentions. Consequently, input devices were categorized by things such as the property sensed (position, motion, or pressure), the number of dimensions sensed, and the muscle groups required to use them.

Recent research at Xerox PARC has built on this work (Card, Mackinlay and Robertson,1990; Mackinlay, Card and Robertson, 1990). Their taxonomy captures a broad part of the design space of input devices. The model captures both the discrete and continuous properties of devices, (unlike that of Buxton, which could only deal with the latter).

Together, this collected research begins to lay a foundation to support design. However, none of the models are complete in themselves, and there are still significant gaps. One is the lack of a vocabulary that is capable of capturing salient features of interactive techniques and technologies in such a way as to afford finding better matches between the two.

In what follows, a model (first suggested in Buxton, Hill and Rowley, 1985) is developed which provides the start to such a vocabulary. It takes the form of a simple state-transition model and builds on the work mentioned above. In particular, it refines the notion of device state introduced in the PARC model of Card, Mackinlay and Robertson (1990) and Mackinlay, Card and Robertson (1990).

## 3. A THREE-STATE MODEL

The model can be introduced within the context of a direct manipulation interface, such as that of the Apple Macintosh. Consider moving the mouse without the button pushed. One way to characterize the state of the system at this point is as *tracking.* If, however, we point at an icon, depress the mouse button and move the mouse while holding the button down, we have entered a new state, *dragging.* These simple states are represented in Fig. 1.

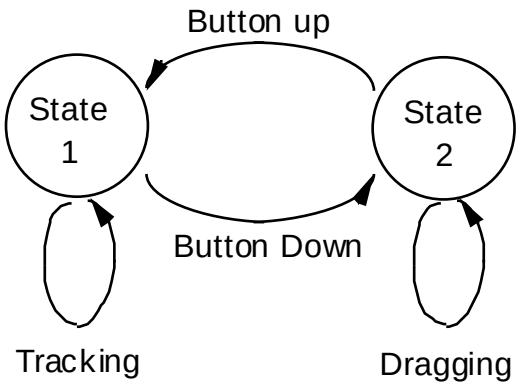


**Figure 1.** Simple 2-State Transaction
*In State 1, moving the mouse causes the tracking symbol to move. Depressing the mouse button over an icon permits it to be dragged when the mouse is moved. This is State 2. Releasing the mouse button returns to the tracking state, State 1.*

Consider now the situation if a touch tablet rather than a mouse was connected to the system. For the purpose of the example, let us assume that the touch tablet is capable of sensing only one bit of pressure, namely touch or no-touch.

In this case we also get two states, but only one is common to the previous example. This is illustrated in Fig. 2. The first state, (State 0), is what we will call *out of range*, (OOR). In this state, any movement of the finger has no effect on the system. It is only when the finger comes in contact with the tablet that we enter the State 1, the tracking state seen in the previous example.

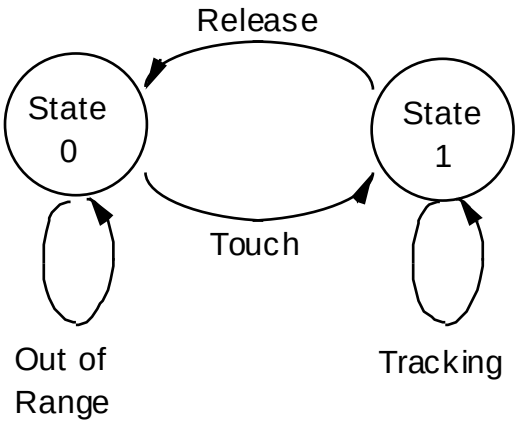


**Figure 2.** State 0-1 Transaction

*Assume a touch tablet. In State 0, moving the finger is out of range (OOR), so has no effect. When the finger is in contact with the tablet, the tracking symbol follows the finger's motion (State 1: tracking). The system returns to State 0 when the finger releases contact from the tablet surface.*

Each example has one state that the other cannot reach. Lifting a mouse off of one's desk is not sensed, and has no effect. No interactive technique can be built that depends on this action. Consequently, State 0 (the OOR condition) is undefined. Conversely, without some additional signal, the touch tablet is incapable of moving into the dragging state (State 2). To do so would require a signal from a supplementary key press or from a threshold crossing on a pressure-sensitive tablet (Buxton, Hill and Rowley, 1985).

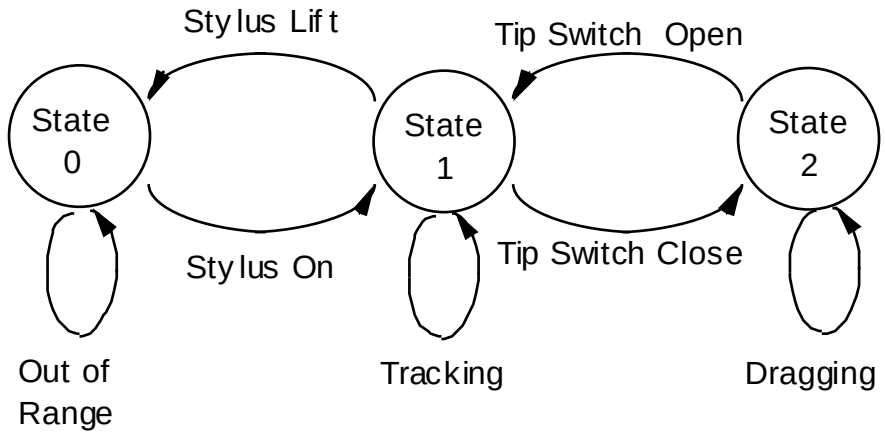


**Figure 3.** State 0-1-2 Transaction

*Assume a graphics tablet with stylus. In State 0, the stylus is off of the tablet and the tip switch in its open state. Moving the stylus has no effect since it is out of range (OOR). When the stylus is in range, the tracking symbol follows the stylus' motion (State 1: tracking). Extra pressure on the stylus closes the tip switch, thereby moving the system into State 2.*

There are, however, transducers that are capable of sensing all three states. A graphics tablet with a stylus would be one example.[1] This is illustrated in Fig. 3.

The three states introduced in the above examples are the basic elements of the model. There can be some variations. For example, with a multi-button mouse (or the use of multiple clicks), State 2 becomes a set of states, indexed by button number, as illustrated in Fig. 4.

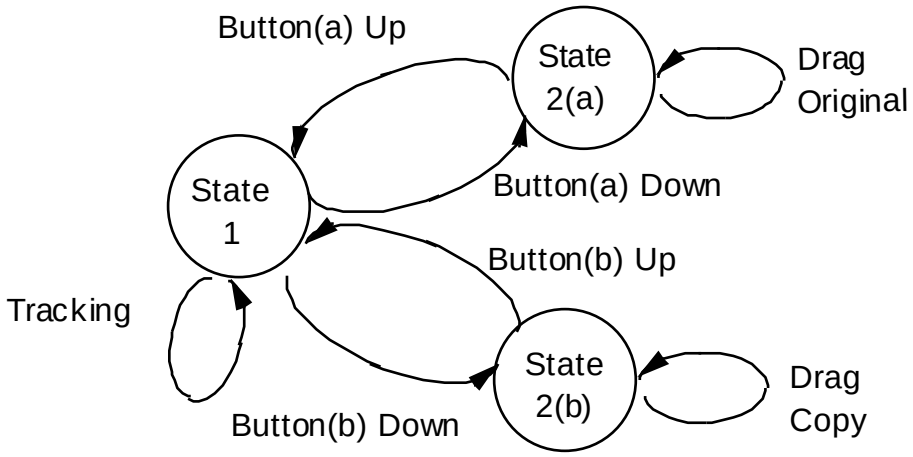


**Figure 4.** State 2 Set

*With a multi-button mouse, for example, multiple State 2's are available. For example, selecting an object with button$_a$ may cause the selected object to be dragged, whereas selecting the object with button$_b$ may mean that a copy is dragged. Multiple State 2s can also be realized by multiple clicks on the mouse, as with the Apple Macintosh, where single clicks are used to select and double clicks to "open."*

While the model is simple, it will be shown how important properties of devices and interactive techniques can be characterized in terms of these three states, and how this representation can be used in design. Weaknesses of the the model and areas for future research will also be discussed.

Note that in what follows, states will be referred to by number (State 1, for example) rather than by description. The consequences of the action performed in a particular state can vary. For example, State 2 could just as easily been "inking" or "rubber banding" as "dragging." The ordinal nomenclature is more neutral, and will be used.

## 4. DEVICES AND TRANSACTIONS: TABULATING ATTRIBUTES

Two tables are presented. The first summarizes the demands of a number of transaction types expressed in terms of the states and state transitions that they require from a supporting transducer. The second summarizes the capabilities of a number of input devices, expressed in terms of this same type of state information. By comparing the two tables, a simple means is provided to evaluate the match between transducers and transactions.

| Transaction | State 0 | State 1 | State 2 | Notes |
|---|---|---|---|---|
| Point | | √ | | |
| Pursuit Track | | √ | | |
| Point/Select | | √ | √ | |
| Drag | | √ | √ | State 2 motion |
| Rubber Banding | | √ | √ | State 2 motion |
| Sweep Region | | √ | √ | State 2 motion |
| Pop/Pull Menu | | √ | √ | State 2 motion |
| Ink | √ | √ | √ | State 2 motion |
| Char Recognition | | √ | √ | State 2 motion |

**Table 1:** State Characteristics of Several Classes of Transaction

*A number of representative types of transactions are listed showing their state and state transition requirements. This table is of use as a means to help verify if a particular transducer is well suited to that class of transaction.*

| Device | State 0 | State 1 | State 2 | Notes |
|---|---|---|---|---|
| Joystick | | √ | 4 | |
| Joystick & Button | | √ | √[3] | |
| Trackball | | √ | 4 | |
| Mouse | | √ | √ | |
| Tablet & Stylus | √ | √ | √ | |
| Tablet & Puck | √[1] | √ | √ | |
| Touch Tablet | √ | √ | 4, 5 | |
| Touch Screen | √ | 2 | √[2] | 6 |
| Light Pen | √ | √ | √ | 6 |

1. The puck can be lifted, but shape and weight discourages this.
2. If State 1 used, then State 2 not available.
3. Button may require second hand, or (on stick) inhibit motion while held.
4. Has no built in button. May require second hand. If same hand, result may be interference with motion while in State 2.
5. State 1-0 transition can be used for selection. See below.
6. Direct device. Interaction is directly on display screen. Special behaviour. See below.

**Table 2:** State Characteristics of Several Input Devices

*A number of representative input devices are listed showing their state and state transition properties. This table is of use as a means to check if a transducer meets the state characteristics required by a particular type of transaction.*

### 4.1 State 1-0 Transitions and Gestures

Transitions from State 1 to State 0 are not significant in most direct manipulation systems. As stylus-driven interfaces using character and gesture recognition become more important, so will this class of state transition. The reason is that this signal is a prime cue to delimit characters. You can think of it as the need for both the system and the user to be able to sense and agree when the pen has left the "paper."

If this cue is to be used in this or other types of transitions, it is important to note that input devices vary in how well they signal the transition. In particular, the majority of tablets (touch and otherwise) give no explicit signal at all. Rather, the onus is on the application to sense the absence of State 1 tracking information, rather than on the transducer to send an explicit signal that the pointing device has gone out of range.

Not only does this put an additional burden on the software implementation and execution, it imposes an inherent and unacceptable delay in responding to the user's action. Consequently, designers relying heavily on this signal should carefully evaluate the technologies under consideration if optimal performance and efficiency are desired.

### 4.2 Point/Select and State Transitions

*Point and select* is an integral component of most direct manipulation interfaces. The transaction is compound: pointing, which is a continuous task, and selection, which is binary. In our vocabulary, the binary selection signal is represented as a state change.

As commonly implemented, the pointing task is undertaken in State 1, and the selection is articulated by a State 1-2-1 transition, with no motion in State 2. This can be easily supported with any of the devices in Table 2 that have plain check marks (√) in the State 1 and State 2 columns.

Some transducers, including trackballs, many joysticks, and touch tablets do not generally support State 2. For the most part this is due to their not having buttons tightly integrated into their design. Therefore, they warrant special mention.

One approach to dealing with this is to use a supplementary button. With joysticks and trackballs, these are often added to the base. With trackballs, such buttons can often be operated with the same hand as the trackball. With joysticks this is not the case, and another limb (hand, foot, etc.) must be employed[2]. As two-handed input becomes increasingly important, using two hands to do the work of one may be a waste. The second hand being used to push the joystick or touch tablet button could be used more profitably elsewhere.

An alternative method for generating the selection signal is by a State 1-0 transition (assuming a device that supports both of these states). An example would be a binary touch tablet, where lifting your finger off the tablet while pointing at an object could imply that the object is selected. Note, however, that this technique does not extend to support transactions that require motion in State 2 (see below). An alternative approach, suitable for the touch tablet, is to use a pressure threshold

crossing to signal the state change (Buxton, Hill, Rowley, 1985). This, however, requires a pressure sensing transducer.

The selection signal can also be generated via a timeout cue. That is, if I point at something and remain in that position for some interval $\Delta t$, then that object is deemed selected. The problem with this technique, however, is that the speed of interaction is limited by the requisite $\Delta t$ interval.

### 4.3 Continuous Motion in State 2

Unlike *point and select,* most of the transactions employing State 2 require continuous motion in that state. These include:

- *dragging:* as with icons;
- *rubber-banding:* as with lines, windows, or sweeping out regions on the screen;
- *pull-down menus;*
- *inking:* as with painting or drawing;
- *character recognition:* which may or may not leave an ink trail.

Consequently, two prerequisites of any supporting technology are:

- State 1 to/from State-2 transitions
- Ease of motion while maintaining State 2.

The first is more obvious than the second, and has been discussed in the previous section.

It is this more obscure second point which presents the biggest potential impediment to performance. For example, this paper is being written on a Macintosh Portable which uses a trackball. While pointing and selecting work reasonably well, this class of transaction does not. Even though both requisite states are accessible, maintaining continuous motion in State 2 requires holding down a space-bar like button with the thumb, while operating the trackball with the fingers of the same hand. Consequently the hand is under tension, and the acuity of motion is seriously affected, compared to State 1, where the hand is in a relaxed state.

### 4.4 Direct Input Devices are Special

*Direct input devices* are devices where input takes place directly on the display surface. The two primary examples of this class of device are *light pens* and *touch screens.*

In terms of the model under discussion, these devices have an important property: in some cases (especially with touch screens), *the pointing device itself (stylus or finger) is the tracking "symbol."* What this means is that they "track" when out of range. In this usage, we would describe these devices as making transitions directly between State 0 and State 2, as illustrated in Fig. 5.

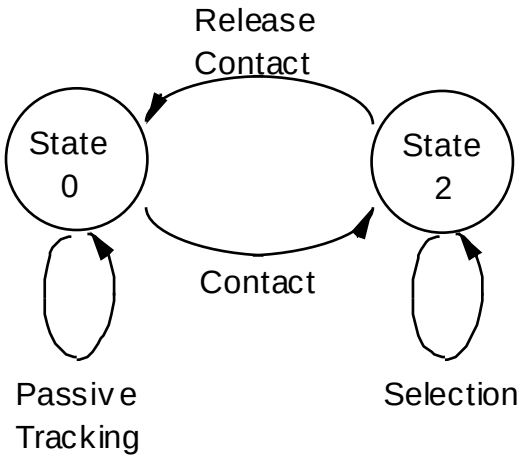


**Figure 5.** State 0-2 Transitions
*With direct devices such as a light pen and touch screen, the pointing device (pen and finger, respectively, is the tracking mechanism. Hence, State 1 is bypassed. Since the tracking is passive (the system does not know what is being pointed at until contact), this tracking state should not be considered as State 1.*

Another way that one might think of characterizing this would be as a simple State 1-2 transition, as shown in Fig. 1. There are at least two reasons that this is not done, however. First, in this case the tracking is passive. The system has no sense of what the finger is pointing at until it comes into contact. This is a significant difference.

Second, there are examples where these same direct devices are used with an explicit State 1. For example, light pens generally employ an explicit tracking symbol. Touch screens can also be used in this way, as been shown by Potter, Shneiderman, and Weldon (1988), and Sears and Shneiderman (1989), among others. In these touch screen examples, the purpose was to improve pointing accuracy. Without going into the effectiveness of the technique, what is important is that this type of usage *converts the direct technology into a State 0-1 device*.

Consider the case of the touch screen for a moment. Choosing this approach means that the price paid for the increased accuracy is direct access to State-2 dependent transactions (such as selection and dragging). Anything beyond pointing (however accurately) requires special new procedures (as discussed above in Sections 4.1 and 4.2).

## 5. SUMMARY AND CONCLUSIONS

A state-transition model has been presented that captures many important aspects of input devices and techniques. As such, it provides a means of aiding the designer in evaluating the match between the two. While discussed in the context of well-known devices, the model can be applied to newer classes of transducers such as the VPL dataglove (Zimmerman, Lanier, Blanchard,Bryson & Harvill, 1987).

The model goes beyond that previously introduced by Buxton (1983) in that it deals with the continuous and discrete components of transducers in an integrated manner. However, it has some weaknesses. In particular, in its current form it does not cope well with representing transducers

capable of pressure sensing on their surface or their buttons (for example, a stylus with a pressure sensitive tip switch used to control line thickness in a drawing program).

Despite these limitations, the model provides a useful conceptualization of some of the basic properties of input devices and interactive techniques. Further research is required to expand and refine it.

## ACKNOWLEDGEMENTS

I would like to acknowledge the contribution of the members of the Input Research Group at the University of Toronto and colleagues at Xerox PARC who provided the forum to discuss and refine the ideas contained in this paper. In particular, I would like to acknowledge Abigail Sellen and Gordon Kurtenbach who made many helpful comments on the manuscript. This work has been supported by the Natural Science and Engineering Research Council of Canada and Xerox PARC.

## NOTES

1. For the example, we are assuming that stylus position is only sensed when in contact with the tablet. This is done for purposes of simplicity and does not detract from the model.

2. I distinguish between joysticks with buttons integrated on the stick, and those that do not. With the former, the stick and button can be operated with one hand. With the latter, two handed operation is required. Note, however, that in the former case operation of the button may affect performance of operation of the stick.

## REFERENCES

Buxton, W. (1983). Lexical and Pragmatic Considerations of Input Structures. *Computer Graphics*, 17 (1), 31-37.

Buxton, W. Hill, R. & Rowley, P. (1985). Issues and Techniques in Touch-Sensitive Tablet Input, *Computer Graphics*, 19(3), Proceedings of SIGGRAPH '85, 215-224.

Card, S., English & Burr. (1978), Evaluation of Mouse, Rate-Controlled Isometric Joystick, Step Keys and Text Keys for Text Selection on a CRT, *Ergonomics*, 21(8), 601-613.

Card, S., Mackinlay, J. D. & Robertson, G. G. (1990). The design space of input devices. *Proceedings of CHI '90*, ACM Conference on Human Factors in Software, in press.

Foley, J. & Van Dam, A. (1982). *Fundamentals of Interactive Computer Graphics*, Reading, MA: Addison-Wesley.

Foley, J.D., Wallace, V.L. & Chan, P. (1984). The Human Factors of Computer Graphics Interaction Techniques. *IEEE Computer Graphics and Applications*, 4 (11), 13-48.

Greenstein, Joel S. & Arnaut, Lynn Y. (1988). Input Devices. In Helander, M. (Ed.). *Handbook of HCI*. Amsterdam: North-Holland, 495-519.

GSPC (1977). Status Report of the Graphics Standards Planning Committee, *Computer Graphics*, 11(3).

GSPC (1979). Status Report of the Graphics Standards Planning Committee, *Computer Graphics*, 13(3).

Mackinlay, J. D., Card, S. & Robertson, G. G. (1990). A semantic analysis of the design space of input devices. *Human-Computer Interaction*, in press.

Milner, N. (1988). A review of human performance and preferences with different input devices to computer systems., in D. Jones & R. Winder (Eds.). *People and Computers IV, Proceedings of*

*the Fourth Conference of the British Computer Society Human-Computer Interaction Specialist Group*. Cambridge: Cambridge University Press, 341-362.

Newman, W.M. (1968b). A System for Interactive Graphical Programming. *Proceedings of the AFIPS Spring Joint Computer Conference*, 47-54.

Potter, R., Shneiderman, B. & Weldon, L. (1988). Improving the accuracy of touch screens: an experimental evaluation of three strategies. *Proceedings of CHI'88,* 27-32.

Sears, A. & Shneiderman, B. (1989). High precision touchscreens: design strategies and comparisons with a mouse, *CAR-TR-450*. College Park, Maryland: Center for Automation Research, University of Maryland.

Sherr, S. (Ed.)(1988). *Input Devices*. Boston: Academic Press.

Zimmerman, T.G., Lanier, J., Blanchard, C., Bryson, S. & Harvill, Y. (1987). A Hand Gesture Interface Device, *Proceedings of CHI+GI '87*, 189-192.