New Human-Computer Interaction Techniques

Robert J.K. Jacob

Human-Computer Interaction Lab, Naval Research Laboratory, Washington, D.C., U.S.A.

Abstract. This chapter describes the area of human-computer interaction technique research in general and then describes research in several new types of interaction techniques under way at the Human-Computer Interaction Laboratory of the U.S. Naval Research Laboratory: eye movement-based interaction techniques, three-dimensional pointing, and, finally, using dialogue properties in interaction techniques.

Keywords. human-computer interaction, interaction techniques, eye movements, gesture, pointing, dialogue

1 Introduction

Tufte [9] has described human-computer interaction as two powerful information processors (human and computer) attempting to communicate with each other via a narrow-bandwidth, highly constrained interface. A fundamental goal of research in human-computer interaction is, therefore, to increase the useful bandwidth across that interface. A significant bottleneck in the effectiveness of educational systems as well as other interactive systems is this communication path between the user and the computer. Since the user side of this path is difficult to modify, it is the computer side that provides fertile ground for research in human-computer interaction. This chapter describes interaction technique research in general and then describes research in several new types of interaction techniques under way at the Human-Computer Interaction Laboratory of the U.S. Naval Research Laboratory (NRL).

Interaction techniques provide a useful focus for human-computer interaction research because they are specific, yet not bound to a single application. An interaction technique is a way of using a physical input/output device to perform a generic task in a human-computer dialogue. It represents an abstraction of some common class of interactive task, for example, choosing one of several objects shown on a display screen. Research in this area studies the primitive elements of human-computer dialogues, which apply across a wide variety of individual applications. The basic approach is to study new modes of communication that could be used for human-computer communication and develop devices and techniques to use such modes. The goal is to add new, high-bandwidth methods to the available store of input/output devices, interaction techniques, and generic dialogue components. Ideally, research in interaction techniques starts with studies of the characteristics of human communication channels and skills and then works toward developing devices and techniques that communicate effectively to and from those channels. Often, though, the hardware developments come first, people simply attempt to build "whatever can be built," and then HCI researchers try to find uses for the resulting artifacts.

2 Eye Movement-Based Interaction Techniques

One of the principal thrusts in interaction technique research at NRL has been eye movements [3, 5]. We have been interested in developing interaction techniques based on eye movements as an input from user to computer. That is, the computer will identify the point on its display screen at which the user is looking and use that information as a part of its dialogue with the user. For example, if a display showed several icons, the user might request additional information about one of them. Instead of requiring the user to indicate which icon is desired by pointing at it with a mouse or by entering its name with a keyboard, the computer can determine which icon the user is looking at and give the information on it immediately.

Our approach to this interaction medium is to try to make use of natural eye movements. This work begins by studying the characteristics of natural eye movements and then attempts to recognize appropriate patterns in the raw data obtainable from an oculometer, turn them into tokens with higher-level meaning, and design interaction techniques for them around the known characteristics of eye movements. A user interface based on eye movement inputs has the potential for faster and more effortless interaction than current interfaces, because people can move their eyes extremely rapidly and with little conscious effort. A simple thought experiment suggests the speed advantage: Before you operate any mechanical pointing device, you usually look at the destination to which you wish to move. Thus the eye movement is available as an indication of your goal before you could actuate any other input device.

However, people are not accustomed to operating devices in the world simply by moving their eyes. Our experience is that, at first, it is empowering to be able simply to look at what you want and have it happen, rather than having to look at it and then point and click it with the mouse. Before long, though, it becomes like the Midas Touch. Everywhere you look, another command is activated; you cannot look anywhere without issuing a command. The challenge in building a useful eye movement interface is to avoid this Midas Touch problem. Carefully designed new interaction techniques are thus necessary to ensure that they are not only fast but that use eye input in a natural and unobtrusive way. Our approach is to try to think of eye position more as a piece of information available to a user-computer dialogue involving a variety of input devices than as the intentional actuation of the principal input device.

A further problem arises because people do not normally move their eyes in the same slow and deliberate way they operate conventional computer input devices. Eyes continually dart from point to point, in rapid and sudden ‘‘saccades.’’ Even when the user thinks he or she is viewing a single object, the eyes do not remain still for long. It would therefore be inappropriate simply to plug in an eye tracker as a direct replacement for a mouse. Wherever possible, we therefore attempt to obtain information from the natural movements of the user’s eye while viewing the display, rather than requiring the user to make specific trained eye movements to actuate the system.

We partition the problem of using eye movement data into two stages. First we process the raw data from the eye tracker in order to filter noise, recognize fixations, compensate for local calibration errors, and generally try to reconstruct the user’s more conscious intentions from the available information. This processing stage uses a model of eye motions (fixations separated by saccades) to drive a fixation recognition algorithm that converts the continuous, somewhat noisy stream of raw eye position reports into discrete tokens that represent the user’s intentional fixations. The tokens are passed to our user interface management system, along with tokens generated by other input devices being used simultaneously, such as the keyboard or mouse.

Next, we design generic interaction techniques based on these tokens as inputs. The first interaction technique we have developed is for object selection. The task is to select one object from among several displayed on the screen, for example, one of several file icons on a desktop. With a mouse, this is usually done by pointing at the object and then pressing a button. With the eye tracker, there is no natural counterpart of the button press. We reject using a blink for a signal because it detracts from the naturalness possible with an eye movement-based dialogue by requiring the user to think about when he or she blinks. We tested two alternatives. In one, the user looks at the desired object then presses a button on a keypad to indicate his or her choice. The second alternative uses dwell time—if the user continues to look at the object for a sufficiently long time, it is selected without further operations.

At first this seemed like a good combination. In practice, however, the dwell time approach proved much more convenient. While a long dwell time might be used to ensure that an inadvertent selection will not be made by simply "looking around" on the display, this mitigates the speed advantage of using eye movements for input and also reduces the responsiveness of the interface. To reduce dwell time, we make a further distinction. If the result of selecting the wrong object can be undone trivially (selection of a wrong object followed by a selection of the right object causes no adverse effect—the second selection instantaneously overrides the first), then a very short dwell time can be used. For example, if selecting an object causes a display of information about that object to appear and the information display can be changed instantaneously, then the effect of selecting wrong objects is immediately undone as long as the user eventually reaches the right one. This approach, using a 150-250 ms. dwell time gives excellent results. The lag between eye movement and system response (required to reach the dwell time) is hardly detectable to the user, yet long enough to accumulate sufficient data for our fixation recognition and processing. The subjective feeling is of a highly responsive system, almost as though the system is executing the user's intentions before he or she expresses them. For situations where selecting an object is more difficult to undo, button confirmation is used rather than a longer dwell time.

Other interaction techniques we have developed and are studying in our laboratory include: continuous display of attributes of eye-selected object (instead of explicit user commands to request display); moving object by eye selection, then press button down, "drag" object by moving eye, release button to stop dragging; moving object by eye selection, then drag with mouse; pull-down menu commands using dwell time to select or look away to cancel menu, plus optional accelerator button; forward and backward eye-controlled text scrolling.

Eye movement-based interaction techniques exemplify an emerging new style of interaction, called non-command-based [7]. Previous interaction styles all await, receive, and respond to explicit commands from the user to the computer. In the non-command style, the computer passively monitors the user and responds as appropriate, rather than waiting for the user to issue specific commands. Because the inputs in this style of interface are often non-intentional, they must be interpreted carefully to avoid annoying users with unwanted responses to inadvertent actions. Our research with eye movements provides an example of how these problems can be attacked.

3 Three-Dimensional Interaction

Another area of interaction technique research at NRL has been an investigation of three degree of freedom input [4]. In studying interaction techniques, each new piece of hardware that appears raises the question ‘‘What tasks is this device good for, and how should it be incorporated into interface designs?’’ Such questions are typically answered specifically for each new device, based on the intuition and judgment of designers and, perhaps, on empirical studies of that device. Our work in three degree-of-freedom input provides an example of how greater leverage can be achieved by answering such questions by reasoning from a more general predictive theoretical framework, rather than in an *ad hoc* way.

We begin by posing the question for the three-dimensional position tracker, such as the Polhemus 3SPACE or Ascension Bird trackers. While directly answering the question ‘‘What is a three-dimensional tracker good for?’’ we also try to shed light on the next level question, i.e., ‘‘How should you answer questions like *What is a three-dimensional tracker good for?*’’ Concepts such as the logical input device provide descriptive models for understanding input devices, but they tend to ignore the crucial pragmatic aspects of haptic input by treating devices that output the same information as equivalent, despite the different subjective qualities they present to the user. Taxonomies and other frameworks for understanding input devices have tended to hide these pragmatic qualities or else relegate them to a ‘‘miscellaneous’’ category, without further structure.

Instead, we draw on the theory of processing of perceptual structure in multidimensional space [1]. The attributes of objects in multidimensional spaces can have different dominant perceptual structures. The nature of that structure, that is, the way in which the dimensions of the space combine perceptually, affects how an observer perceives an object. We posit that this distinction between perceptual structures provides a key to understanding performance of multidimensional input devices on multidimensional tasks. Hence two three-dimensional tasks may seem equivalent, but if they involve different types of perceptual spaces, they should be assigned to correspondingly different input devices.

The three-dimensional position tracker can be viewed as a three-dimensional absolute-position mouse or data tablet; it provides continuous reports of its position in three-space relative to a user-defined origin. The device thus allows the user to input three coordinates or data values simultaneously and to input changes that cut across all three coordinate axes in a single operation. (A mouse or trackball allows this in only two dimensions.) Such a device is obviously useful for pointing in three-space, but it is also applicable in many other situations that involve changing three values simultaneously. We considered two tasks that both involve three degrees of freedom, i.e., that require adjusting three variables. For comparison with the three-dimensional tracker, we used a conventional mouse (for two of the three variables in the tasks) and then provided a mode change button to turn the mouse temporarily into a one-dimensional slider for the third variable.

A naive view of these two alternatives suggests that the three-dimensional tracker is a superset of the two-dimensional mouse, since it provides the same two outputs plus a third. Thus the three-dimensional tracker should always be used in place of a mouse (assuming ideal devices with equal cost and equal accuracy), since it is always at least as good and sometimes better. Our intuition tells us that this is unlikely—but why? The goal of this research is to develop a firmer foundation from which to draw such judgments. To do this, we extend Garner’s theory of processing of perceptual structure [1], first developed with fixed images, to interactive graphical manipulation tasks and thereby use it to shed light on the selection of multidimensional input devices. Garner observed that relationships between the attributes of an object can be perceived in two ways that differ in how well the component attributes remain identifiable. Some attributes are *integrally* related to one another—the values of these attributes combine to form a single composite perception in the observer’s mind, and each object is seen as a unitary whole; while other attributes are *separably* related—the attributes remain distinct, and the observer does not integrate them, but sees an object as a collection of attributes.

Our hypothesis is that the structure of the perceptual space of an interaction task should mirror that of the control space of its input device. To examine it, we considered two interactive tasks, one set within an integral space and one in a separable one, and two input devices, one with integral dimensions and one, separable. This yields a two by two experiment, with four conditions. We expect performance on each task to be superior in the condition where the device matches that task in integrality/separability. That is, the interaction effect between choice of task and choice of device should far exceed the main effects of task or device alone.

For the integral three-attribute task in the experiment, the user manipulates the *x*-*y* location and the size of an object to match a target, since location and size tend to be perceived as integral attributes; for the separable task, the user manipulates the *x*-*y* location and color (lightness or darkness of greyscale) of an object to match a target, since location and color are perceived separably. The difference in perceptual structure between these two tasks is in the relationship of the third dimension (size or greyscale) to the first two (*x* and *y* location); in all cases, the *x* and *y* attributes are integral.

For the integral device condition, we use a Polhemus tracker, which permits input of three integral values. For the separable condition, we use a conventional mouse, which permits two integral values, to which we added a mode change to enable input of a third—separable—value. Our hypothesis predicts that the three degree of freedom input device will be superior to the two degree of freedom (plus mode change) device only when the task involves three integral values, rather than in all cases, as with the naive hypothesis mentioned above.

Our experimental results strongly supported this hypothesis. We found that neither device is uniformly superior to the other in performance. Instead, we find significantly better performance in the experimental conditions where the task and device are both integral or both separable and inferior performance in the other two conditions. These results support our extension of the theory of perceptual space to interaction techniques, which predicts that the integral task (size) will be performed better with the integral device (Polhemus) and that the separable task (greyscale) will be performed better with the separable device (mouse).

4 Dialogue Interaction Techniques

Another direction in our research is the notion of dialogue interaction techniques [6, 8]. In a direct manipulation or graphical interface, each command or brief transaction exists as a nearly independent utterance, unconnected to previous and future ones from the same user. Real human communication rarely consists of such individual, unconnected utterances, but rather each utterance can draw on previous ones for its meaning. It may do so implicitly, embodied in a conversational focus, state, or mode, or explicitly. Most research on the processes needed to conduct such dialogues has concentrated on natural language, but some of them can be applied to any human-computer dialogue conducted in any language. A direct manipulation dialogue is conducted in a rich graphical language using powerful and natural input and output modalities.

The user's side of the dialogue may consist almost entirely of pointing, gesturing, and pressing buttons, and the computer's, of animated pictorial analogues of real-world objects. A dialogue in such a language could nevertheless exhibit useful dialogue properties, such as following focus. For example, a precise meaning can often be gleaned by combining imprecise actions in several modes, each of which would be ambiguous in isolation. We thus attempt to broaden the notion of interaction techniques in these two dimensions (multiple transactions and multiple modes).

A useful property of dialogue that can be applied to a graphical interface is focus [2]. The graphical user interface could keep a history of the user's current focus, tracking brief digressions, meta-conversations, major topic shifts, and other changes in focus. Unlike a linguistic interface, the graphical interface would use inputs from a combination of graphical or manipulative modes to determine focus. Pointing and dragging of displayed objects, user gestures and gazes as well as the objects of explicit queries or commands all provide input to determine and track focus.

Human dialogue often combines inputs from several modes. Deixis often involves a pointing gesture that does not precisely specify its object; the listener deduces the correct object from the context of the dialogue and, possibly, from integrating information from the hand gesture, the direction of the user's head, tone of his or her voice, and the like. The user could, similarly, give a command and point in a general direction to indicate its object. The interface would disambiguate the pointing gesture based on the recent history of its dialogue with the user and, possibly, by combining other information about the user from physical sensors. An imprecise pointing gesture in the general direction of a displayed region of a map could be combined with the knowledge that the user's recent commands within that region referred principally to one of three specific locations (say, river *R*, island *I*, and hill *H*) within the region and the knowledge that the user had previously been looking primarily at islands displayed all over the map. By combining these three imprecise inputs, the interface could narrow the choice down so that (in this example) island *I* is the most likely object of the user's new command.

We call these higher-level interaction elements dialogue interaction techniques, and we have begun designing and testing them in our laboratory. We are also developing a software architecture for handling these properties that span more than one transaction. It treats them as orthogonal to the usual lexical, syntactic, and semantic partitioning of user interface software. Our first system demonstrates the use of a focus stack in an interactive graphics editor. In the future, we will expand to a richer representation of dialogue than a stack, to support a wider range of dialogue interaction techniques.

5 Conclusions

This chapter has provided an overview of a variety of new human-computer interaction techniques we are studying and building at NRL. Interaction techniques like these, when applied to the design of specific interfaces, increase the useful bandwidth between user and computer. This seems to be the key bottleneck in improving the usefulness of all types of interactive computer systems, and particularly educational systems, which depend heavily on dialogues with their users.

Acknowledgments

I want to thank my colleagues at the Naval Research Laboratory, Dan McFarlane, Preston Mullen, Manuel Perez, Linda Sibert, and Jim Templeman, who have worked on much of the research I have described in this chapter. This work was sponsored by the Office of Naval Research.

References

1. W.R. Garner, *The Processing of Information and Structure*, Lawrence Erlbaum, Potomac, Md., 1974.
2. B.J. Grosz, ‘‘Discourse,’’ in *Understanding Spoken Language*, ed. by D.E. Walker, pp. 229-284, Elsevier North-Holland, New York, 1978.
3. R.J.K. Jacob, ‘‘The Use of Eye Movements in Human-Computer Interaction Techniques: What You Look At is What You Get,’’ *ACM Transactions on Information Systems*, vol. 9, no. 3, pp. 152-169, April 1991.
4. R.J.K. Jacob and L.E. Sibert, ‘‘The Perceptual Structure of Multidimensional Input Device Selection,’’ *Proc. ACM CHI’92 Human Factors in Computing Systems Conference*, pp. 211-218, Addison-Wesley/ACM Press, 1992.
5. R.J.K. Jacob, ‘‘Eye Movement-Based Human-Computer Interaction Techniques: Toward Non-Command Interfaces,’’ in *Advances in Human-Computer Interaction, Vol. 4*, ed. by H.R. Hartson and D. Hix, pp. 151-190, Ablex Publishing Co., Norwood, N.J., 1993. http://www.eecs.tufts.edu/˜jacob/papers/hartson.txt [ASCII]; http://www.eecs.tufts.edu/˜jacob/papers/hartson.pdf [PDF].
6. R.J.K. Jacob, ‘‘Natural Dialogue in Modes other than Natural Language,’’ in *Dialogue and Instruction*, ed. by R.-J. Beun, M. Baker, and M. Reiner, pp. 289-301, Springer-Verlag, Berlin, 1995. http://www.eecs.tufts.edu/˜jacob/papers/como.html [HTML]; http://www.eecs.tufts.edu/˜jacob/papers/como.pdf [PDF].
7. J. Nielsen, ‘‘Noncommand User Interfaces,’’ *Comm. ACM*, vol. 36, no. 4, pp. 83-99, April 1993.
8. M.A. Perez and J.L. Sibert, ‘‘Focus in Graphical User Interfaces,’’ *Proc. ACM International Workshop on Intelligent User Interfaces*, Addison-

Wesley/ACM Press, Orlando, Fla., 1993.

9. E.R. Tufte, ‘‘Visual Design of the User Interface,’’ IBM Corporation, Armonk, N.Y., 1989.