Attentive Interfaces for Multiple Monitors

Mark Ashdown and Yoichi Sato
Institute of Industrial Science, University of Tokyo
4-6-1 Komaba, Meguro-ku, Tokyo 153-8505, Japan
mark@ashdown.name, ysato@iis.u-tokyo.ac.jp

ABSTRACT
The use of multiple monitors is becoming popular but it creates problems for pointer movement, window management, and control of applications. We suggest combining the standard input devices of keyboard and mouse with attentive user interface techniques. A user's work with conventional input devices will naturally be based around a focal region, but that focus can be rapidly transferred in response to attentive input.

Author Keywords
Gaze-contingent display, attentive user interface, head pose estimation, multiple monitors.

ACM Classification Keywords
H.5.2 [Information interfaces and presentation]: User Interfaces - Interaction styles.

FOCUS PLUS CONTEXT
The size and shape of a display affects how it will be perceived and used. Large displays and distributed display environments are quantitatively different from a standard monitor, being larger and containing more pixels, but they are also qualitatively different, affording different styles of working.

During physical tasks in large workspaces people naturally make use of focus and context regions: human vision only senses high resolution in the centre with the periphery being in much lower resolution, and people place primary objects and information in a small area in front of them while using the surrounding area for supporting items.

As displays become larger it becomes important to consider focus plus context techniques in computer interfaces. As a display gets larger and has more pixels navigating around it with a mouse and selecting graphical items becomes more difficult, thus techniques to make it easier to move the mouse pointer to items, or items to the mouse pointer, become useful. Czerwinski *et al.* [1] found that users of a large display were surprised when the completely unobscured window they were looking at did not receive keyboard events—actually a different window was active but was positioned outside the user's field of view. This shows that users were only concentrating on the windows in their focal region, and did not consider any that might be in the periphery until they turned to look at them. In a distributed display environment made of multiple monitors the partitioning of the display is also significant. Users tend to think of the displays as separate thus, for instance, they position a window within a particular monitor frame and avoid placing it across a boundary between two monitors [2].

COMPLEMENTARY INPUTS
The conventional computer input devices are the keyboard and mouse. These devices support explicit inputs—the user consciously performs every button press or mouse movement. Attentive interfaces use sensing techniques like computer vision to generate implicit inputs from measurements of the user's attention—for instance, where the user is looking.

Figure 1. We track the user's head pose using cameras, shown here above and below the centre monitor.

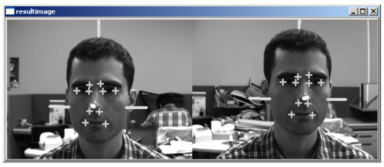

Figure 2. Feature points on the user's face are tracked in stereo, and the position and orientation of the head is continually estimated.

For a single monitor, eye tracking can be combined with the mouse to reduce the amount of mouse movement, as in Magic Pointing [7] which combined eye tracking and fixation detection with normal mouse movement to move the mouse pointer close to the location of a fixation. However, with several monitors the head is moved through a large range of positions and angles which makes eye tracking difficult.

Rather than using eye tracking, we have used a head tracking system developed in our lab [6] to create a system to move the mouse pointer between monitors and switch the active application. We have previously experimented with head tracking for switching between windows on a single-monitor system and for zooming and scrolling a map [3], and for a prototype focus plus context system on a large projected display [5] In our multi-monitor system we use head tracking to detect which monitor the user is looking at. When the focal monitor changes, the mouse pointer jumps to the new monitor, and the top window on that monitor is activated so it can receive keyboard events without any explicit action to switch applications. To move a window between monitors the user can start dragging it with the mouse, then look to another monitor to have it jump there. We tested eight users with a window management task and found that they required significantly less mouse movement than without the system, and preferred using it to the conventional one, although task time actually increased. The main problem was moving between nearby points on adjacent monitors, in which case the system did not always select the correct monitor. To address this problem we are improving the processing of head motion by implementing fixation detection for head motion in an analogous way to that which is used for eye motion.

We intend this combination of input mechanisms to match the user's perceptual process so that problems such as keyboard events going to invisible windows are avoided, and to aid pointer movement and window management tasks which become cumbersome as the size, resolution, and number of separate parts of the display increases. We also believe it will make it easier to use side monitors for more than purely passive information display—if we reduce the effort required to switch to applications that are placed in the periphery it should become easier to intersperse the central task with brief interactions with the peripheral information.

DESIGN & EVALUATION

Three important criteria that should be considered during the design of interfaces based on sensing techniques such as eye or head tracking are:

- the distinction between implicit and explicit inputs
- whether tracking data are treated as continuous or discrete
- the cost of mistakes

Implicit inputs are generated automatically in response to sensing of the user's inadvertent movements; explicit ones are consciously performed. The keyboard and mouse are used for explicit input, but we would like to use data such as head pose implicitly so the user does not have to think about it. The continuous stream of data generated by a tracking algorithm can either be used to continuously adjust parameters of the application, or it can be converted to a series of discrete events by, for example, applying a threshold, or using it in a hidden Markov model. In the map scrolling application the head pose was coupled to a continuous variable—the scroll and zoom settings of the map—but in the screen switching application we have chosen to discretize it so that at any moment one of three monitors is selected. An abrupt change of focus occurs at the point when the user is deemed to have moved to a new monitor, but other schemes are possible, and the feedback provided by a continually reacting display could mitigate inaccuracy in the system. Finally, the cost of mistakes should be kept low so that the user can easily recover from their own errors or errors of the tracking system, which is similar to the undo feature in many applications which avoids the need to worry about the possible outcome of every action.

Operating systems generally do not use knowledge about the division of the display into multiple monitors to modify the interface, but this should be exploited. For instance, window management would be aided by snapping windows to monitor edges, and operations like maximizing should work on an individual monitor basis. Attentive input based on head tracking could add further enhancements: pop-up windows or newly started programs could appear on the monitor the user is currently looking at, or the sound volume of multiple programs could treated as continuous variables and scaled based on the distance between a program's screen location and the user's point of regard. Seemingly passive displays could be updated in response to the user's attention—for instance, an email program, stock ticker, or news application could display new messages and information when the user looked at it, rather than distracting him/her from another task. The use of multiple displays increases multitasking, thus the user's attention is spread more thinly among applications, and the tradeoff between attention and utility [4] becomes a principle design concern for any program that must notify the user of new information.

ACKNOWLEDGEMENTS

This work on the head tracking interfaces was funded by a postdoctoral fellowship from the Japan Society for the Promotion of Science.

REFERENCES

1. M. Czerwinski, G. Smith, T. Regan, B. Meyers, G. Robertson, and G. Starkweather. Toward Characterizing the Productivity Benefits of Very Large Displays. In *Proceedings of IFIP Interact 2003*, pages 9–16, 2003.
2. J. Grudin. Partitioning Digital Worlds: Focal and Peripheral Awareness in Multiple Monitor Use. In *Proceedings of CHI 2001*, pages 458–465, 2001.
3. K. Kitajima, Y. Sato, and H. Koike. Vision-Based Face Tracking System for Window Interface: Prototype Application and Emperical Studies. In *Extended Abstracts of CHI 2001*, pages 359–360, 2001.
4. D. S. McCrickard and C. M. Chewar. Attuning Notification Design to User Goals and Attention Costs. *Communications of the ACM*, 46(3):67–72, 2003.
5. Y. Nakanishi, T. Fujii, K. Kiatjima, Y. Sato, and H. Koike. Vision-Based Face Tracking System for Large Displays. In *Proceedings of Ubicomp 2002*, 2002.
6. Y. Sato, K. Oka, H. Koike, and Y. Nakanishi. Video-Based Tracking of User's Motion and Its Use for Augmented Desk Interface. In *IEEE International Conference on Automatic Face and Gesture Recognition (FG 2004), May 2004.* , 2004.
7. S. Zhai, C. Morimoto, and S. Ihde. Manual And Gaze Input Cascaded (MAGIC) Pointing. In *Proceedings of CHI 99*, pages 246–253, 1999.