

Peek and Look: Accessing Off-Screen Targets Using Head Tracking

Clayton Epp, Miguel Nacenta, Carl Gutwin, Regan Mandryk
University of Saskatchewan, Department of Computer Science
176 Thorvaldson Building
110 Science Place
Saskatoon, SK S7N 5C9, CANADA
Tel: 1-306-966-4886

cce381@mail.usask.ca, nacenta@cs.usask.ca, gutwin@cs.usask.ca, regan@cs.usask.ca

ABSTRACT

Although screen displays are getting larger in size and more cost effective, data sets are becoming more complex and demand increasing amounts of space on the screen. Toolbars and menus occupy considerable space and compete with the data or document being viewed. To maximize the screen real estate available for data manipulation, we propose storing the tools off-screen. To access these off-screen tools we present two interaction techniques that allow the position of the head to change the viewport state (e.g. by panning or zooming), leaving the hands free for other tasks. In this preliminary work, we explore variations of these interaction techniques and challenges that arise in their implementation and use.

ACM Classification: H5.m. [User Interfaces]: Interaction styles.

General terms: Design, Human Factors

Keywords: navigation, off-screen targets, head tracking.

INTRODUCTION

In many graphical applications such as geographic information systems (GIS) and fine-grained illustration, the workspace is drastically larger than the available viewport. Competing demands put a premium on screen space, in particular, between the data view and the space for the tools required. Tools occupy a lot of room; for example, in the image-editor application shown in Figure 1, the window layout allocates 23% of the screen for tools, leaving only 77% for data.

A number of solutions for this problem exist: shortcut keys, transparent toolbars, configurable UI elements, virtual desktops, and focus+context techniques. However, these techniques are not without their shortcomings; they require expert knowledge (shortcuts), context switching (virtual desktops), obstruct interaction (transparent toolbars), or use valued screen space (configurable UI / focus + context techniques).

We suggest that tools be stored off-screen, around the edges of the visible workspace. For seldom-used tools, off-screen storage may be simpler to access than hunting through hierarchical menu systems. The idea of storing tools off-screen means that applications can make use of a much larger workspace than the monitor provides. For example, extending the workspace in Figure 1 by 10% in all directions results in an addition that is almost half (44%) of the original workspace size. In order to access this content we propose two interaction techniques where viewport adjustments are achieved via head-tracking.

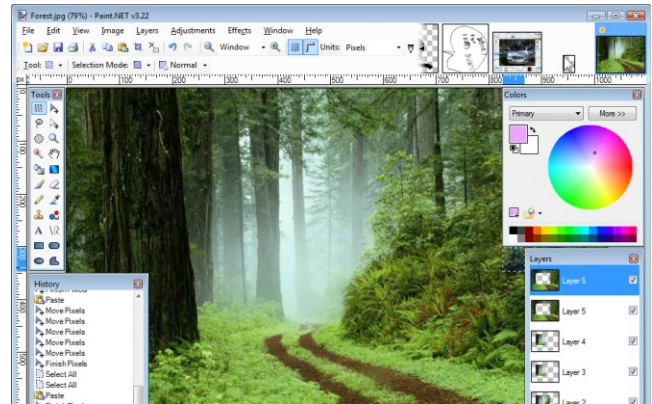


Figure 1: Typical data / tool space relationship.

RELATED WORK

Some techniques use arrows and halos around the periphery of the screen to visualize off-screen targets [1]. Hop [4] combines numerous techniques to improve selection of off-screen targets.

There has also been research in using just-off-screen eye tracking to facilitate reading tasks [5]; however the research focuses mainly on automatic scrolling of the viewport. They use eye-gaze and dwell time to access off-screen buttons, whereas we use head movement.

Recent work has used physical proximity to control graphic enlargement as a method of natural interaction to improve readability on monitors [2]. However, this method focuses on on-screen content with user proximity awareness

whereas we focus on head gaze in conjunction with mouse interactions while acquiring off-screen targets.

HEAD-TRACKING INTERACTION

We propose off-screen interaction through a combination of two types of user input: mouse interaction for targeting, and head-tracking for viewport adjustments. As the user moves their head, the viewport adjusts, revealing areas of off-screen content.

Two main variations were developed. The first technique, called *Peek*, is similar to the experience one gets when looking through a window frame. If the user wishes to see more of the outside area to the right, they would move their body to the left slightly while looking out the right edge of the window. The middle image in Figure 2 illustrates this scenario. In *Peek* we substitute the monitor bezel for the window edge. Using this same metaphor, when a person walks towards a window, more of the outside becomes visible to the user. We implemented this in the prototype by scaling the display to reveal the outer surrounding area.

The *Look* technique works similarly to *Peek* but in the reverse direction and without leaning the body. In this technique the user looks at the edge of the display that they wish to see more of, and the viewport moves to reveal that edge. The right image in Figure 2 illustrates the *Look* technique. A zooming version was also implemented for *Look*, where the screen zooms out revealing the outer edges when the user moves away from the screen.

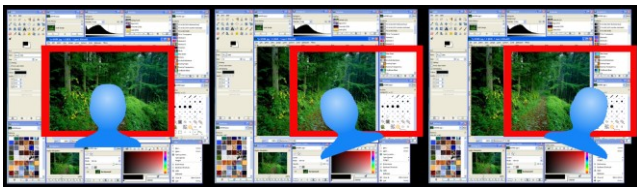


Figure 2: Rest (left), Peek (middle), Look (right)

DESIGN ISSUES

A number of issues became apparent throughout the design process of the prototype.

Sensitive head input: Small movements of the head could possibly invoke inadvertent movements of the screen leading to an uncontrolled and disruptive user experience. Adjustable input filtering as well as threshold boundaries were used to ensure that these small movements did not engage movement in the display.

Movement type: Two basic types of movement were incorporated into our prototype: discrete and continuous. With discrete movement, the screen is automatically animated revealing all the content as long as the technique is engaged. In contrast, continuous movement maps the head coordinates directly while constantly updating the screen. This follows the window metaphor more closely than the discrete method as peeking through a window is a continuous motion in reality.

Movement direction: Restrictions can also be configured on each edge and axis of the display. Such restrictions may be helpful in moving the screen in a more controlled fashion.

Movement speed: There are two considerations that need to be taken into account here: the animation speed for the discrete movement and the amount of physical movement to virtual movement in the continuous version (C:D ratio). We considered speed variations as an important aspect of the interaction. For example, someone could like the discrete movement but dislike the animation speed.

PROTOTYPE

The prototype application for this interaction utilizes Lee's method of head-tracking using the Nintendo Wii controller [3]. In this case, the Wii controller is situated below the monitor, pointing at the user's head. Head coordinates are then tracked via infrared lights worn by the user which are translated into viewport movements.

A modified version of the open source Virtual Network Computing (VNC) software, TightVNC [6], was used as the viewport for our prototype. This software allows us to display a remote computer desktop while adjusting the viewport via head coordinates, making it possible to present the user with real applications that run on the remote machine.

FUTURE WORK

This new way of interacting with off-screen objects shows promise of improving interaction in many ways. We suspect that this type of interaction will allow people to better remember the location of off-screen objects due to both the spatial layout of the tools and through the development of haptic memory. In future studies we will focus on investigating the impact of spatial memory, studying targeting performance access methods for off-screen targets, and exploring the costs associated with having targets off-screen.

REFERENCES

1. Baudisch, P. and Rosenholtz, R. (2003). Halo: a technique for visualizing off-screen objects. *Proc. CHI'03*, 481-488.
2. Harris, C. and Dey, A.K. (2008). Lean and zoom: proximity-aware user interface and content magnification. *Proc. CHI'08*, 507-510.
3. Head tracking for desktop VR displays using the Wii remote. <http://www.cs.cmu.edu/~johnny/projects/wii/>.
4. Irani, P., Gutwin, C., Yang, X. D. (2006). Improving selection of off-screen targets with hopping. *Proc. CHI'06*, 299-308.
5. Kumar, M., Winograd, T. (2007). Gaze-enhanced scrolling techniques. *Proc. UIST'07*, 213-216.
6. TightVNC Software. <http://www.tightvnc.com>