Exploring the Design of Accessible Goal Crossing Desktop Widgets

Eun Kyoung Choe[1]
Kristen Shinohara[1]
Parmit K. Chilana[1]
Morgan Dixon[2]
Jacob O. Wobbrock[1]

[1]The Information School
University of Washington
{eunky, kshino, pchilana, wobbrock}@u.washington.edu

[2]Computer Science & Engineering
University of Washington
mdixon@cs.washington.edu

Abstract

Prior work has shown that goal crossing may be a more accessible interaction technique than conventional pointing-and-clicking for motor-impaired users. Although goal crossing with pen-based input devices has been studied, pen-based designs have limited applicability on the desktop because the pen can "fly in," cross, and "fly out," whereas a persistent mouse cursor cannot. We therefore explore possible designs for accessible mouse-based goal crossing widgets that avoid triggering unwanted goals by using secondary goals, gestures, and corners and edges. We identify four design principles for accessible desktop goal crossing widgets: ease of use for motor-impaired users, safety from false selections, efficiency, and scalability.

Keywords: Crossing-based interfaces, input, human performance, mouse cursor, desktop accessibility, motor impairments, computer access.

ACM Classification Keywords

H5.2. [Information interfaces and presentation]: User interfaces – *input devices and strategies*.

Introduction

The mouse and trackball are popular devices for interacting with computers. Yet, for motor-impaired users, pointing, clicking, and dragging with these conventional devices can be difficult and error-prone

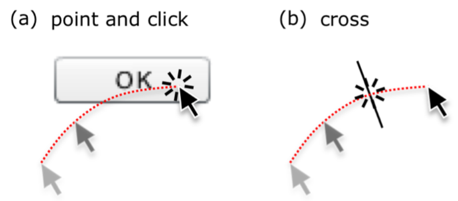

Figure 1. Two different ways of acquiring a target: (a) pointing to a target followed by clicking, (b) goal crossing by passing over a goal line.

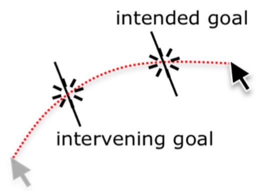

Figure 2. The occlusion problem

[7]. Accessibility research has not yet adequately addressed the challenges motor-impaired users face when using desktop computers with everyday input devices.

Users with motor impairments may have tremor or poor coordination, which affects their use of mice or trackballs during conventional point-and-click tasks (Figure 1a). Users may experience difficulty in moving the mouse cursor to within a target, and they may slip outside a target upon executing a click [6].

As an alternative to pointing-and-clicking, we propose to develop desktop user interfaces based on *goal crossing*. Goal crossing does not involve positioning a cursor inside a confined area or clicking a target. Instead, a target is acquired by passing over a goal line (Figure 1b), which our prior work has shown offers better performance for motor-impaired users [8].

While crossing seems simple enough, a major challenge is the *occlusion problem*, where nearby goals interfere with the intended target (Figure 2). A pen or stylus affords the ability to “fly in” to the target before crossing, and to “fly out” after crossing, but a persistent mouse cursor on the desktop has no such option. As a result, surrounding targets may occlude the desired target in two-dimensions, making target acquisition with crossing prone to unwanted selections. A seemingly obvious solution is to hold down the mouse button when crossing, but this requires users to *drag* across goals, which is known to be particularly difficult for users with motor impairments, especially when using trackballs [7]. We therefore are addressing this problem by designing crossing widgets that require specific and deliberate actions to select a target, and yet remain easy for motor-impaired people to perform.

In this work, we explore designs for accessible goal crossing widgets that solve the occlusion problem in our ongoing effort to create more accessible crossing-based desktop applications. The mouse and trackball are popular with users with motor impairments [8]. Therefore, instead of replacing these ubiquitous and inexpensive devices, we are investigating how to make the software they control more accessible.

Related work

Goal crossing was introduced to the field of human-computer interaction by Accot & Zhai [1]. They showed that simple goal crossing followed Fitts’ law and that it can be as efficient as pointing-and-clicking [1,2]. *CrossY* [3], a pen-based drawing application, showed the practical aspects of goal crossing, such as fluid composition of commands enabling users to select multiple goals in a single cross. However, the work of Accot & Zhai and CrossY assumes users are able-bodied and using pen-based input devices. In comparison, crossing rarely appears on the desktop. One exception is *DontClick.It* [4], a Flash-based Web design project that can be navigated without clicking. Our recent experimental study of crossing on the desktop [8] found that motor-impaired users were able to perform faster with goal crossing than pointing, and also greatly preferred it. Since in many cases, crossing can be just as efficient for able-bodied users [2], it may be a viable alternative for desktop user interfaces in general, not just for improved accessibility.

Design principles

In this section, we highlight four design principles that we identify as essential for creating accessible goal crossing widgets for the desktop.

- *Ease of motor-impaired performance:* Our goal crossing widgets probably should avoid mouse clicking, dragging, or use of complicated gestures. Simple gestures may be used.
- *Safety*: Unwanted targets should not be accidentally triggered despite being casually crossed as the cursor traverses the interface. This must be held in balance with the first design principle, as goals that trigger easily may also be triggered accidentally. Finding the "sweet spot" between these two principles is a key challenge.
- *Efficiency*: Along with ease of performance, the number of steps to acquire a goal must be minimized. Even if each step is easy to perform, having too many steps will result in inefficient designs.
- *Scalability*: In real user interfaces, there are possibly hundreds of menu items, links, buttons, and icons. Goal crossing widgets should be able to handle high-density layouts of the kind found in such user interfaces.

Certainly there are other design considerations such as ease of implementation or ease of canceling an unwanted selection. However, we have found the four design principles above to be the most useful in highlighting the strengths and weaknesses of our designs.

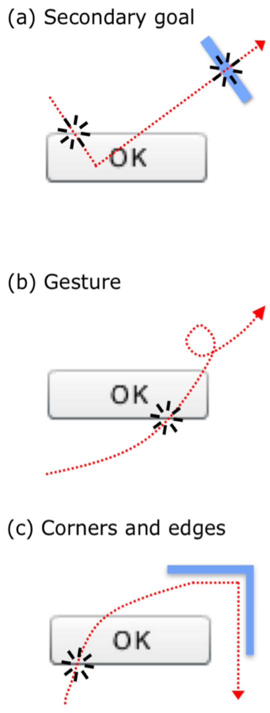

Figure 3. Three new types of goal crossing ideas: (a) crossing secondary goal, (b) gesturing, (c) utilizing corners and edges.

Design set

In this section, we introduce three new types of goal crossing designs that solve the occlusion problem. Most of the following crossing ideas are made up of two steps: (1) *activation*, which is the initial crossing event, and (2) *confirmation*, which then acquires the target. For completeness, we also mention four "basic" crossing ideas that are quick and simple solutions but require difficult motor movements, such as clicking and dragging across goals.

We have placed our design ideas into three groups based on how they address the occlusion problem. The three groups are (1) crossing a secondary goal, (2) gestures, and (3) utilizing corners and edges (Figure 3). The first two were briefly mentioned in our prior work [8], but were not fully developed there.

Crossing a secondary goal

Crossing the initial target activates the goal, and then a second goal appears. The second goal is crossed to confirm the selection. Many design variations exist depending on the parameters of the second goal: location of where the second goal appears, distance between the first and the second goal, and shape of the second goal. Figure 4a shows an idea where the position of a secondary arc depends on the approach angle of the mouse. This makes orthogonal crossing of the second goal easier. As seen in Figure 4b, it may be better in some cases for a second goal to appear in a fixed location for certain layouts.

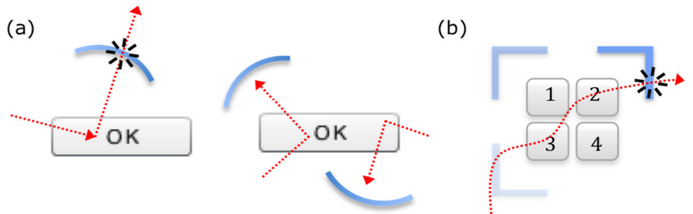

Figure 4. Secondary goal crossing: (a) re-positioning secondary goal: arc, (b) fixed location-secondary goal in case of a four-item cluster

Simple gestures

Simple mouse gestures may be a useful way to disambiguate intentional crosses from unintentional ones. Gestures can be made right before crossing a goal to activate it, right after crossing a goal to confirm it, or while crossing a goal to indicate confirmation. Different gestures must be explored and tested for their feasibility for people with motor-impairments. Making a 90° turn after crossing a goal is one example of a very simple gesture. Other gestures that may be more difficult are encircling a target, making a pig-tail, or scribbling on or near the target.

One of our most promising ideas is an adaptation of *Hover Widgets* [5], originally a pen-based input technique where the pen makes a 90° turn (an L-shaped gesture) when in the hover state of a Tablet PC. As in Figure 4, this idea can be applied to mouse-based crossing interfaces where turning 90° activates the most recently crossed goal. When the mouse crosses the end of the tunnel, the crossed goal is confirmed. Further study needs to be done to determine if this is feasible for motor-impaired people, especially with a trackball. However, the Hover Widgets design has ample flexibility built in—for instance, one can move varied lengths before turning 90°, and the tolerance for "straightness" can be adjusted (Figure 5b). The need to make a 90° turn after an initial cross should prevent unwanted selections, increasing safety.

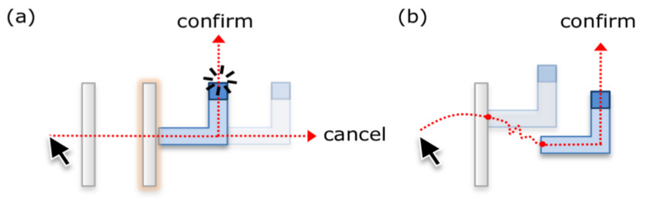

Figure 5. Gesture: (a) Hover Widget: cross and turn 90°, and cross again until hitting the end of the tunnel, (b) the tunnel repositions itself whenever the cursor moves out of it.

Utilizing corners and edges

Here, we take advantage of impenetrable or semi-penetrable corners and edges, which are easy to acquire. The idea behind Figure 6a and 6b is to place a confirmation box or hard edge right after crossing a goal. Each corner of the box represents a context menu customizable with different commands. If a goal is crossed, a box with a context menu appears, and if the corner is hit from inside the box, then it will be selected. Figure 6b is another variation. These ideas may make motor-impaired performance easy, but also may obstruct a free-moving mouse cursor. This limitation may be reduced by rendering edges behind or orthogonal to the goal, requiring a change in direction, or by using semi-penetrable edges that are passable with some persistence or speed.

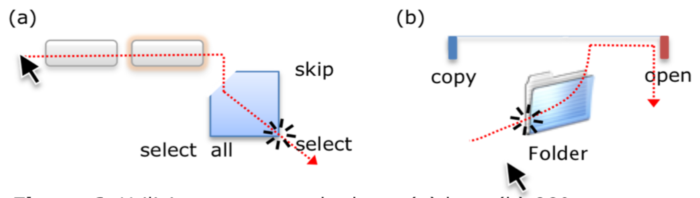

Figure 6. Utilizing corners and edges: (a) box, (b) 90° turn.

For comparisons: Basic crossing ideas

For comparisons, we include four "basic" crossing ideas that are simple and straightforward, but probably difficult for motor-impaired people to perform. "Cross and click anywhere" uses a mouse button like conventional pointing-and-clicking, but does not require specific targeting—after a goal is crossed, it is confirmed by a click anywhere within a certain distance or time of the initial cross. While this may work, it still requires a click, which we would like to avoid because of its reliance on finger dexterity. "Drag and cross" is more difficult because it requires holding down the mouse button *while* crossing. “Cross and press a keyboard key” would utilize the keyboard instead of the mouse button, and could be performed with the aid of a second hand. However, this design might be awkward for people with only one hand or who are unable to hold both hands on the desk surface. Finally, “cross and cross back” removes the need for buttons, but violates the follow-through advantage of goal crossing, and may require users to effectively “point.” We have built crossing widget prototypes for each of these four basic crossing ideas for use in controlled studies.

Design space
We locate the above crossing designs on a plane where *x*-axis is *Safety* and *y*-axis is *Ease of motor impaired performance*. Our goal is to create a crossing widget design scheme that is high on both axes, such that it provides the best compromise between target acquisition performance and reducing selection errors.

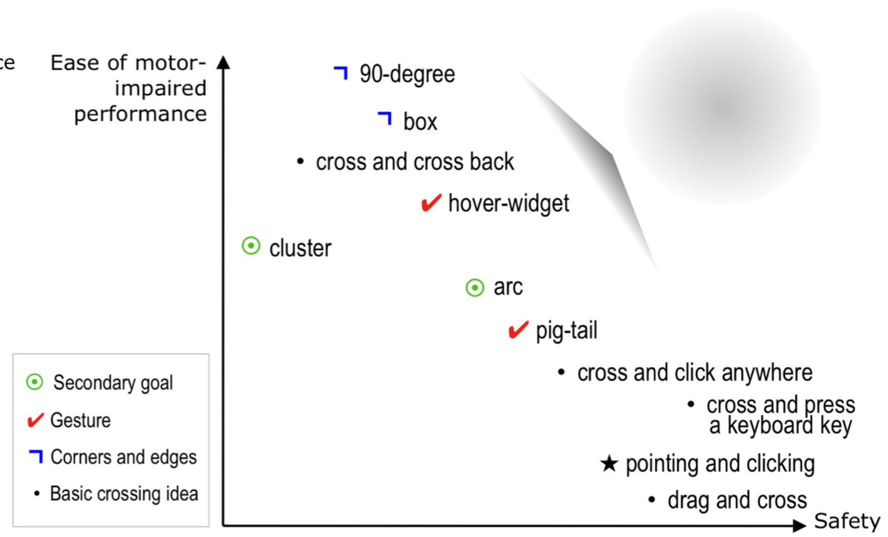

Figure 7. Design space of the accessible goal crossing widgets

Architectural challenges
The architecture underlying current windowing systems assumes that widgets (e.g., buttons, menus, scroll bars) are only responsible for the area they encompass and the actions that occur *inside* it. These controls receive mouse input messages only from within their on-screen boundaries. By doing so, such controls fundamentally prohibit interactions that require input from outside their boundaries, such as receiving mouse input from the areas beyond the widget. Also, they are only responsible for painting themselves within their boundaries. Both of these assumptions are challenged by our crossing widget designs, which may require crossing widgets to have an awareness of the mouse cursor at all times and to draw outside their narrow boundaries.

In designing crossing-based interfaces, we have an opportunity to question some of the fundamental architectural assumptions of the point-and-click paradigm, and to thereby extend the flexibility of graphical user interface toolkits. Our future work includes the creation of a goal crossing toolkit that will enable rapid development of accessible goal crossing interfaces for the desktop.

Ongoing work
We are in the process of creating prototypes of the above designs and others that uphold our design principles. The most promising prototyped designs will be tested with motor impaired users. These user studies will evaluate the efficiency and satisfaction of target acquisition time, ease of use, and the effectiveness of overcoming the occlusion problem, especially when targets are clustered densely. Successful designs will be implemented as widgets, and

we will expand our designs to include not just buttons, but also scroll bars and menus. Exemplar desktop crossing applications will follow, enabling us to fully assess the accessibility of this new click-free interaction paradigm. Also, we will explore the usefulness of this paradigm for other domains in which a mouse click is absent, such as for eye-tracking, laser pointing, or voice-controlled user interfaces.

Conclusion

We have presented design principles for accessible goal crossing desktop widgets, and explored different design ideas to address the occlusion problem. Secondary goal crossing, gesturing, and utilizing corners and edges are three categories of our goal crossing widget designs that do not require mouse clicking or dragging. We also presented a design space of these schemes, along with four "basic" schemes, defined by axes of *Safety* and *Ease of motor-impaired performance*. This design space has helped us to identify the qualities of the ideal design, which should score high on both axes. Our next step is to create prototypes, evaluate them, and improve them iteratively with motor-impaired users. Ultimately, this work lowers the barriers to desktop computer access by giving motor-impaired users more effective use of everyday mice and trackballs.

Acknowledgements

This work was supported by the National Science Foundation under grant IIS-0811063. Any opinions, findings, conclusions, or recommendations expressed in this work are those of the authors and do not necessarily reflect those of the NSF.

References

[1] Accot, J. and Zhai, S. (1997) Beyond Fitts' law: Models for trajectory-based HCI tasks. Proc. ACM CHI'97. New York: ACM Press, 295-302.

[2] Accot, J. and Zhai, S. (2002) More than dotting the i's—Foundations for crossing-based interfaces. Proc. ACM CHI '02. New York: ACM Press, 73-80.

[3] Apitz, G. and Guimbretière, F. (2004) CrossY: A crossing-based drawing application. Proc. ACM UIST '04. New York: ACM Press, 3-12.

[4] Frank, A. (2005) DontClick.It Diploma project in Communication Design, University of Essen-Duisburg, Essen, Germany. http://www.dontclick.it

[5] Grossman, T., Hinckley, K., Baudisch, P., Agrawala, M. and Balakrishnan, R. (2006) Hover Widgets: Using the tracking state to extend the capabilities of pen-operated devices. Proc. ACM CHI '06. New York: ACM Press, 861-870.

[6] Hwang, F., Keates, S., Langdon, P. and Clarkson, J. (2004) Mouse movements of motion-impaired users: A submovement analysis. Proc. ACM ASSETS '04. New York: ACM Press, 102-109.

[7] Trewin, S. and Pain, H. (1999) Keyboard and mouse errors due to motor disabilities. International Journal of Human-Computer Studies, 50 (2), 109-144.

[8] Wobbrock, J.O. and Gajos, K.Z. (2008) Goal crossing with mice and trackballs for people with motor impairments: Performance, submovements, and design directions. ACM Transactions on Accessible Computing 1 (1), 4:1-4:37.