The Nudging Technique: Input Method without Fine-Grained Pointing by Pushing a Segment

Shota Yamanaka
Meiji University
4-21-1, Nakano, Tokyo, Japan
stymnk@meiji.ac.jp

Homei Miyashita
Meiji University and JST CREST
4-21-1, Nakano, Tokyo, Japan
homei@homei.com

ABSTRACT
The *Nudging Technique* is a new manipulation paradigm for GUIs. With traditional techniques, the user sometimes has to perform a fine-grained operation (e.g., pointing at the edge of a window to resize). When the user makes a mistake in the pointing, problems may arise such as an accidental switching of the foreground window. The *nudging technique* relieves the user from the fine pointing before dragging; the user just moves the cursor to a target then pushes it. Visual and acoustic feedbacks also help the user's operation. We describe two application examples: window resizing and spreadsheet cell resizing systems.

Author Keywords
Nudging technique; graphical user interfaces (GUIs); pointing technique; drag-and-drop; mouse cursor operation.

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

INTRODUCTION
Researchers aiming to improve pointing speed and accuracy in GUIs have made many proposals on the basis of Fitts' law [3]. Some input paradigms that differ from traditional pointing, such as *crossing-based interfaces* [2], have also been proposed. In this paper, we describe a new interaction technique called the *nudging technique*. With this technique, when the user wants to drag an object, (s)he only has to move the cursor to the target and then push it; there is no need for fine-grained pointing before dragging.

Figure 1 illustrates some traditional input techniques and our approach. The *nudging technique* can be used with rougher manipulation than the *pointing technique*, and it reduces error because the user does not need to move the cursor onto a narrow target, as in the c*rossing-based interface* [2]. The *crossing technique* is triggered when the cursor passes a goal; conversely, the *nudging technique* focuses on maintaining a push on a line segment. Nevertheless, the condition before the cursor reaches targets is the same in both cases; therefore, our technique can be

UIST'13-Adjunct, October 8–11, 2013, St. Andrews, United Kingdom.
ACM 978-1-4503-2406-9/13/10.
http://dx.doi.org/10.1145/2508468.2514927

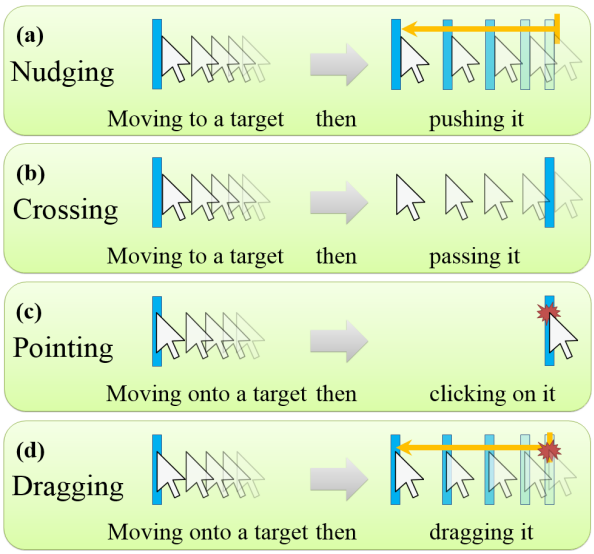

Figure 1. Cursor manipulation techniques for a narrow target. (a) Our proposed technique: keep on nudging a target. (b) Crossing-based interface [2]: a trigger is activated when the cursor crosses a target. (c) Traditional pointing. (d) Dragging is divided into two steps: pointing at a target, as in (c), and then dragging it.

modeled in the same manner as the *crossing technique*. Until targets are reached, the *nudging technique* has lower error rates than the *pointing technique* according to "pointing narrow targets vs. crossing lines" in the results of [2]. After reaching targets, the *nudging technique* follows the rule of the steering task [1] because the cursor should be within the line length during pushing; so the *nudging technique* is modeled according to the *steering law* [1]. Therefore, the movement times of nudging targets of width W that lie at a distance D for length L are given by

$$T_c = a_c + b_c \log_2\left(\frac{D}{W} + 1\right), T_s = a_s + b_s\left(\frac{L}{W} + 1\right)$$

where T_c is the movement time to reach targets (formulated in *crossing-based interfaces* [2]), T_s is the time during nudging (formulated in the *steering law* [1]), and a_c, a_s, b_c, and b_s are constants. The total movement time in the *nudging technique* is $T_c + T_s$.

There are some steering tasks such as movement in hierarchical menus, but there is no method that actively uses a steering operation in existing PC applications. The *nudging technique* is the first such instance.

APPLICATION EXAMPLES

We implemented two application examples to demonstrate the *nudging technique*: window resizing system and spreadsheet cell resizing system. To use the nudging function, the user activates a trigger by, for example, pressing key(s) (e.g. Ctrl) or holding a side button of a mouse that the user can configure optionally. When activated, in order to provide visual feedback, the cursor changes into a fist shape in the direction of the movement. A hitting sound is produced for acoustic feedback when the cursor pushes a line, and the fist then turns red from white.

When the fist cursor touches a frame of windows or cells, the system changes their sizes to keep the edge a certain distance away from the fist cursor. Thus the user can resize them as if (s)he pushes their border lines.

Window resizing system

When a user wants to resize a window, (s)he sometimes misses pointing at an edge of the window, which then causes a problem. For example, clicking inside the right edge may cause unintended scrolling; clicking outside that edge may cause switching of the foreground window. In these cases, the user has to perform an additional operation to recover the condition: scrolling back, or switching the foreground window again, respectively. After that, the user then tries again to point at an edge.

We developed a window resizing system using Win32API. The user can resize a window by pushing the border lines from the outside and from the inside without being concerned about the narrowness of the draggable area because the process of pointing at narrow targets is eliminated. Therefore, not only can errors in operation be reduced, but also the user can manipulate with an easy mind—"*I just push it*." Figure 2-left shows the process of pushing the right edge of a window.

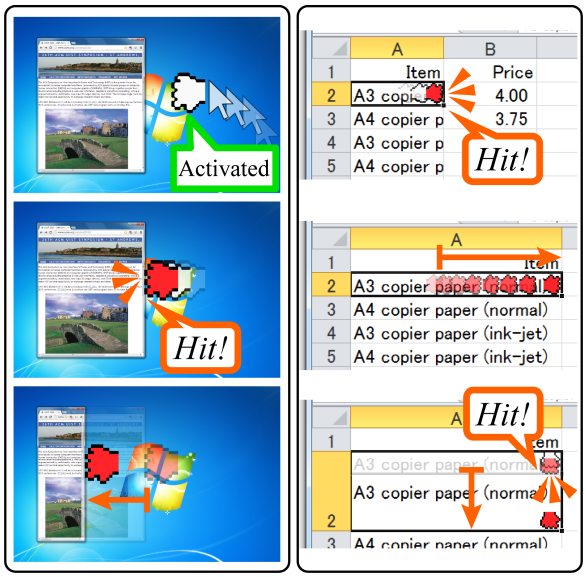

Figure 2. Images of prototype systems. (left) Resizing a window. (right) Resizing a cell horizontally then vertically.

Cell resizing system on spreadsheets

Resizing cells in spreadsheets also requires the accuracy. We developed a cell resizing system using VBA. The user can resize cells by pushing the border lines where column and row numbers are written (alphabets and digits). Additionally, the user also can resize cells by pushing their frames, so the user does not need to move the cursor to the top or left of the spreadsheet; Figure 2-right shows the process of pushing the right and then bottom edge of a cell.

DISCUSSION

In demonstrations at our university, many people understood the advantages of the *nudging technique*. Some said that they usually used the *point-and-drag* method because there is no other choice. Other comments were expressed: (1) ease of use, with only a mouse without side buttons being sufficient, (2) the usefulness of visual feedback or sound feedback, since with the normal arrow cursor it is hard to know whether the current mode is "nudging." Feedbacks, fist cursor, and sound, were implemented after the demonstrations, but the trigger activating the nudging function is our current work. We would like to investigate the best way to use our method.

In our future work, we plan to investigate the effectiveness of the *nudging technique* through formal experiments, such as evaluating operation times and error rates in the window and cell resizing tasks and making corresponding comparisons with traditional techniques. Additionally, it is also important to compare the *time from the start of the task trial to when resizing begins*, because we expect that it will be easier than with the traditional *point-and-drag*.

We implemented two other modes. The sticky mode is pushing and pulling of a line segment after the cursor hits it. The other mode is moving a window by pushing it from the outside. The evaluations of these modes too will be included in our future work.

Is it enough to solve the accuracy problem by expanding a "draggable" area? The user sometimes scrolls in a web browser or a long document by clicking on the right-scroll region and often selects multiple line units by dragging the column and row numbered area. These existing operations are important for the user, so another solution for a comfortable, speedy, and accurate operation is needed. We believe the *nudging technique* may be one answer.

REFERENCES

1. Accot, J. and Zhai, S. Performance Evaluation of Input Devices in Trajectory-Based Tasks: An Application of the Steering Law. *Proc. CHI 1999*, 466-472.
2. Accot, J. and Zhai, S. More than Dotting the i's --- Foundations for Crossing-Based Interfaces. *Proc. CHI 2002*, 73-80.
3. Fitts, P.M. The Information Capacity of the Human Motor System in Controlling the Amplitude of Movement. *Journal of Experimental Psychology*, vol.47, no.6, 1954, 381-391.