Rake Cursor: Improving Pointing Performance with Concurrent Input Channels

Renaud Blanch & Michaël Ortega
Laboratoire d'Informatique de Grenoble
Université Grenoble I / CNRS
385, rue de la Bibliothèque, B.P. 53
F-38041 Grenoble cedex 9, France
blanch@imag.fr, ortega@imag.fr

ABSTRACT

We investigate the use of two concurrent input channels to perform a pointing task. The first channel is the traditional mouse input device whereas the second one is the gaze position. The *rake cursor* interaction technique combines a grid of cursors controlled by the mouse and the selection of the active cursor by the gaze. A controlled experiment shows that rake cursor pointing drastically outperforms mouse-only pointing and also significantly outperforms the state of the art of pointing techniques mixing gaze and mouse input. A theory explaining the improvement is proposed: the global difficulty of a task is split between those two channels, and the sub-tasks could partly be performed concurrently.

Author Keywords

Fitts' law, multi-channel pointing, rake cursor.

ACM Classification Keywords

H.5.2 [**Information Interfaces and Presentation (e.g., HCI)**]: User Interfaces – *Graphical user interfaces, Input devices and strategies.*

INTRODUCTION

Pointing is a fundamental task in graphical user interfaces (GUIs). Many interaction techniques have been proposed to reduce pointing time. This paper explores a new approach: using two concurrent input channels to perform a pointing task. The first channel is the regular mouse. The mouse movements move the cursor in a standard manner. The only modification is that the standard cursor is replaced by a *grid of cursors*, all moving together. The second input channel is the gaze. The eye movements do not move the cursors, they have no motor effect. The gaze position is only used to select which cursor is *active*, i.e. where the traditional mouse events are send to the system. Figure 1 illustrates this principle with a hexagonal (hex) grid of cursors, the active one being the closest to the gaze position (figured by a red disc), the others being semi-transparent. We call this technique *rake cursor*.

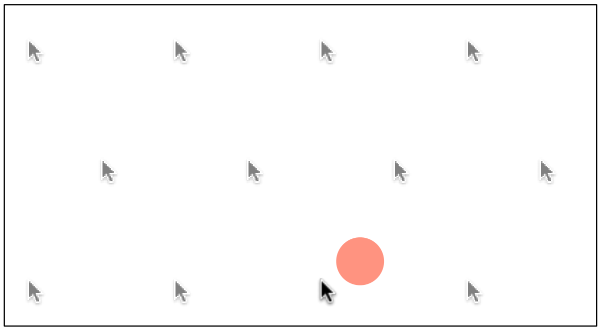

Figure 1. Rake of cursors. The gaze (figured by the red disc) selects the active cursor, other cursors are semi-transparent.

Rake cursor can be seen as a merge of several previously proposed interaction techniques. Gaze position to select object in GUIs has been proposed and evaluated for a while [5, 9]. However, using the gaze to interact is still under investigation (e.g., [7]) and not very much widespread, perhaps because most interaction techniques use the gaze as a motor input which is unnatural as noted by Zhai et al. [10]. The MAGIC interaction technique they proposed to circumvent this problem has a limitation that could explain that rake cursor performs better: the mouse and gaze input are sequential and the cursor jumps near the target but at a position difficult to predict for the user.

Using a grid of cursors has been recently investigated with the Ninja Cursors technique [6]. The problem that impairs this technique is the possible ambiguity occurring when multiple cursors hover over different potential targets at the same time. Instead of requiring a suplemental interaction to resolve this ambiguity, as Ninja Cursors does, rake cursor prevents it by using a suplemental input channel which states explicitly at any time which cursor is the active one.

After describing the rake cursor technique and its implementation, we describe a controlled experiment that compares our technique to normal pointing and to the state of the art of gaze-enhanced pointing. We then propose a theory to explain the observed benefit of the rake cursor technique. Finally, we discuss potential extensions of this technique.

RAKE CURSOR

Figure 2 illustrates rake cursor in action during a drag-and-drop interaction: the user wants to put a folder in the trash.

http://doi.acm.org/10.1145/1518701.1518914

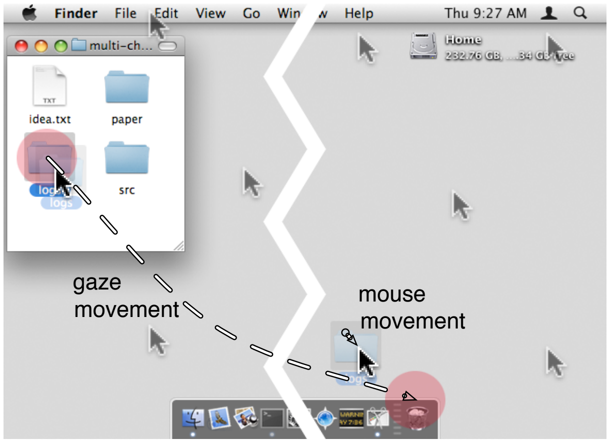

Figure 2. Multiple cursors drag- (left) and-drop (right). The gaze selects the active cursor (red discs figure the gaze position added, cursors magnified).

The normal interaction implies to traverse the whole screen while holding the mouse button depressed, which is a tedious task. With the rake cursor, the cursor grabbing the folder is active when the user starts the drag (left). Since the goal is the trash, it becomes naturally the gaze focus, which makes the closest cursor active (right). Since the cursor was already at that location the change does not introduce visual discontinuity and thus does not perturb the eye. The movement remaining to complete the task is easy: its amplitude is bounded by the distance between the cursors (D_R).

Implementation

Given the system cursor position, we compute the possible positions for the cursor on a hex grid. We choose this grid because of its regularity and because it is known to be the densest plane lattice packing[1] thus giving the best tradeoff between cursor density and D_R.

Those rake positions are used each time a mouse movement is detected as illustrated by Algorithm 1:

- the position of the gaze is monitored and recorded in order to be used by the mouse movement handler;
- the motion of the mouse is also monitored, it triggers the recording of the current cursor position, and then schedules an immediate redisplay;
- when redisplaying, a semi-transparent cursor is drawn at each rake position, and the position closest to the last known gaze position is recorded (best position);
- finally, if the current system cursor position is not the best, it is wrapped to this position (triggering a mouse motion event, which reenters the process described here).

Two implementation details should be noticed. First, after each redisplay the gaze position is updated to the computed best position. We do this to reduce the impact of a lack of gaze tracking: the cursor is not trapped in the neighborhood of the last known gaze position if it is not updated. This enables a graceful degradation of the technique. Second, the gaze does not produce any modification of the state of the GUI by itself. The active cursor can change only when a mouse motion event occurs. Thus, when the user does not move the mouse, the display is totally stable. A reference implementation for Mac OS X is made freely available[2].

Algorithm 1 Rake cursor – handling events

Variables starting with p are 2D positions on the screen.

```
global $p_g$, $p_c$                                  gaze and cursor positions
procedure ON_GAZE_MOVE($p$)
    $p_g \leftarrow p$                               record gaze position
procedure ON_MOUSE_MOVE($p$)
    $p_c \leftarrow p$                               record cursor position
    REQUEST_REDISPLAY( )
procedure ON_REDISPLAY( )
    $p_b \leftarrow p_c$                             compute best cursor position
    $d_b \leftarrow \|p_c - p_g\|$                   (i.e. minimum distance to gaze)
    for all $p \in$ COMPUTE_RAKE_POSITIONS($p_c$) do
        $d \leftarrow \|p - p_g\|$
        if $d < d_b$ then
            $p_b \leftarrow p$
            $d_b \leftarrow d$
        if $p \neq p_c$ then                         draw supplemental cursors
            DRAW_CURSOR($p$)
    if $p_b \neq p_c$ then                           wrap system cursor
        MOVE_SYSTEM_CURSOR($p_b$)
    $p_g \leftarrow p_b$
```

Applications

The rake cursor can be used directly on any GUI. It only requires to have a gaze tracker. This requirement may seem high given the cost of current eye tracking solutions, but the tracking our technique needs does not have to be very precise. Since the gaze is not used to point but to select the active cursor, the requirement on the precision of the tracking is low: it only needs to disambiguate between cursors that are D_R (typically ≈ 400) pixels distant from each others. This precision is much less than the one expected for standard eye trackers. A pure software tracking using a webcam could thus be sufficient for the rake cursor technique.

Rake cursor would be of particular interest for disabled people with limited movement capacity. It reduces the amplitude of movements, thus limiting the effort needed. In contrast to other eye tracking based input techniques, it does not use the gaze as a motor channel, and thus it does not stress the user.

Another good property of the rake cursor is that the cursor is literally *anywhere* at *anytime*. The little (but annoying) trouble of loosing the cursor is totally suppressed by the rake cursor technique: the cursor can not be loosed because it is where you look. More seriously, extending the rake to multiple displays would generalize the Multi-Monitor Mouse technique [2]. This requires an eye trackers spanning the displays or multiple eye trackers but also solves other issues such as drag-and-drop spanning displays [1].

[1] E. W. Weisstein. Circle Packing. 2008.
http://mathworld.wolfram.com/CirclePacking.html

[2] The source code is available at:
http://iihm.imag.fr/blanch/projects/rake-cursor/.

EXPERIMENTAL VALIDATION

To test the rake cursor idea, we ran a controlled experiment. Since we just wanted to show that the idea is worth exploring, we choose to use the well established, minimalist, *à la* Fitts, 1D protocol. In such a setup where there is only one target, there is no possible ambiguity for the Ninja Cursors technique, so it would at least be as good as the rake cursor technique. Comparing the two techniques would require a more elaborated setup (e.g., a 2D setup inspired by the ISO 9241-9) stressing the techniques with distractor targets.

Task

Participants had to perform successive 1D discrete pointing tasks. They had to move the cursor, represented by a one pixel thick vertical black line, to the start position marked by a gray rectangle on the left of the screen, rest there for 0.5 s, start moving to the target —a blue rectangle— as soon as it appeared on the right, and click it (Figure 3). After each block, their error rates were displayed and they were encouraged to conform to a nominal 4% error rate by speeding up or slowing down.

Conditions and Procedure

The control (C) condition used a mouse without acceleration. The magic (M) condition used a 1D implementation of the MAGIC technique: on a mouse motion, if the cursor is farther than 120 pixels from the gaze, it is wrapped to the gaze with a 120 pixels undershoot (distance suggested by [10]). The rake (R) condition used a 1D rake i.e. an array of 6 one pixel thick vertical black lines 200 pixels distant.

Four IDs (3, 4, 5, 6) and two sizes (D = 511 or 1023 pixels) were used, giving eight possible tasks. Ws are then given by Equation (1) that links ID, D and W. A pseudo-random series of 80 trials (10 times each possible task), balanced to minimize order effects, was build. This series was split into 2 blocks of 40 trials to allow a pause in the middle of the series. Those two blocks were repeated for each technique condition, making each participant perform 240 trials.

An order for the three conditions was chosen for each participant: the first pair of blocks were performed using the C condition, the second and third pairs were performed using the R then M conditions for half of the participants, and the M then R condition for the other half. Each series was preceded by 10 randomly-chosen tasks using the same condition to train the participants.

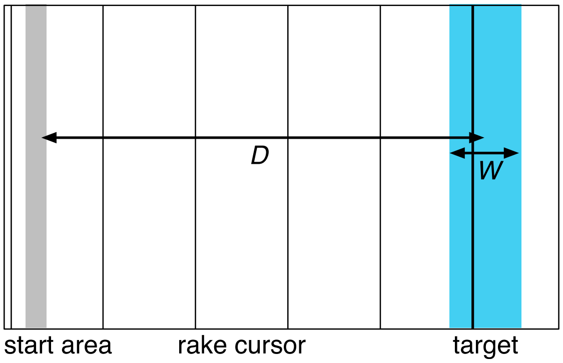

Figure 3. Experimental setup with a rake cursor (the active cursor on the target is the one which is looked at by the user).

Subjects and Apparatus

Sixteen unpaid adult volunteers, 5 female and 11 male, served in the experiment. We had to discard one of the them because the gaze tracker could not produce accurate gaze position for him (presumably due to reflects on his glasses). The experiment was conducted using a custom software and a Tobii ET-17 eye tracker (17-inch 1280 × 800 monitor).

Results

The effects of the technique were explored by analyzing four dependent variables: error rate (ER), reaction time (RT), movement time (MT), and total time ($TT = RT + MT$). Repeated measures analyses of variance were performed on these four variables. We analyzed the effects of the three factors (3 conditions, 4 indices of difficulty, and 2 sizes) in a within-participant full-factorial design.

Error rate

ER is 3.86% on average (slightly better than the 4% consign). The strongest effect on ER is the technique ($F_2 = 5.96, p = 0.0026$[3]), the second one is ID ($F_3 = 4.55, p = .0035$). **The rake condition gives the best error rate** ($2.25 \pm 2.2\%$[4] vs. $2.75 \pm 2.0\%$ for M; $6.58 \pm 4.1\%$ for C).

Movement time

The strongest effect on MT is the technique condition ($F_2 = 120.66*$[5]). **The rake condition gives the lowest movement times** ($0.64 \pm .14$s vs. $0.88 \pm .10$s for M; $1.39 \pm .22$s for C). As expected, ID has also a very strong effect ($F_3 = 45.57*$). We can also notice a significant effect of the interaction between D and the condition on MT which is probably due to the fact that for R, the difficulty of the pointing task is not ID but ID_S which depends on D and D_R.

Reaction time

The strongest effect ($F_2 = 131.82*$) on RT is the condition. This is not usual for pointing experiments. Further investigation shows that the RT are pair-wise significantly different (Student's t test), although less significantly for C and M. **The rake condition gives the slowest reaction time** ($0.40 \pm .06$s vs. $0.30 \pm .04$s for M; $0.28 \pm .03$s for C). A plausible explanation is that the multiple cursors of the rake adds a cognitive load: the selection of the active cursor is not fully parallelized with the pointing.

Total time

Since RT is longer for the R condition, and MT shorter, the best way to compare the techniques is to consider the total time TT. The main effects on TT are the condition ($F_2 = 88.18*$) and ID ($F_3 = 47.46*$). As for MT, the interaction between D and the condition is significant. A Student's t test shows that TT is significantly different for each pair of conditions with **R being the fastest technique, M the second fastest, both outperforming C** ($1.04 \pm .15$s vs. $1.18 \pm .09$s for M; $1.68 \pm .22$s for C, Figure 4).

[3] The total number of degree of freedom (371) is omitted for Fs.

[4] $\mu \pm \sigma$ gives the mean (μ) and standard deviation (σ) across users.

[5] $*$ denote $p < .0001$.

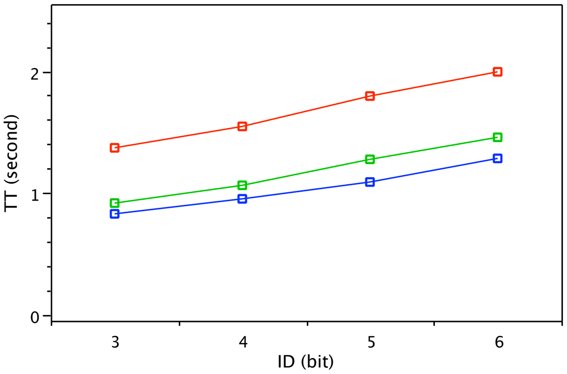

Figure 4. Total time vs. index of difficulty. From slowest (top) to fastest (bottom): control (C); magic (M); and rake (R) condition.

On average, the rake cursor interaction technique performs better (less errors, reduced pointing time), providing a 38.1% gain on the total pointing time, while the magic interaction technique provides a 29.8% benefit.

PROPOSED INTERPRETATION

In this section we propose a theory that could explain the gain observed in the experiment. For a normal pointing interaction, Fitts' law [3] is accurate at modeling the movement time (MT) as a linear function of the index of difficulty (ID), itself a function of the target distance (D) and width (W):

$$MT = a + b \times ID, \quad \text{with} \quad ID = \log_2\left(\frac{D}{W} + 1\right). \tag{1}$$

With the rake cursor technique, this expression of the difficulty is not relevant: the distance to the target depends upon the active cursor. If we consider that the user gaze is on the target, the active cursor is the one closest to the target. The distance to the target is then bounded by the distance between the cursors of the rake (D_R). The difficulty of the mouse pointing task (ID_M) is thus bounded:

$$ID_M \leq \log_2\left(\frac{D_R}{W} + 1\right). \tag{2}$$

The other sub-task is a selection performed by the gaze. Such a task is known to follow Hick's law [4]: the time to choose between n equally probable options is proportional to $\log_2(n+1)$. In our case, the number of choices is given by the number of cursors in the rake, which is inversely proportional to the distance between them (D_R) so the selection time should be proportional to the selection difficulty (ID_S):

$$ID_S = \log_2\left(\frac{k}{D_R} + 1\right). \tag{3}$$

ID can be interpreted as bits of information to transmit to the system. The "information theory" interpretation of Fitts' law [8] states that those bits have to be transmitted through a channel —the movement— with bounded throughput (b) thus leading to MT. The same bits have to be transmitted by our rake cursor technique so we have $ID_M + ID_S = ID$. The movement time and the selection time ($T_M = a + b \times ID_M$, and $T_S \propto ID_S$) could be overlapped if the movement and selection sub-tasks could be performed concurrently. In the experiment, the longer reaction time for rake cursor means that the overlap is not complete: the selection task retards the pointing. But the total time shows that the bandwidth between the user and the system is overall a bit better than with MAGIC.

CONCLUSION

We presented the *rake cursor* input technique, aimed at facilitating pointing in GUIs by using two concurrent input channels: the mouse motion to move a grid of cursors; and the gaze position to select the active cursor. We explained the details of the technique and provide a working implementation for the Mac OS X system. We have shown that the rake cursor technique outperforms the MAGIC technique while also not overloading the visual channel with a motor control. We have proposed an explanation for this improvement: the rake cursor technique allows to use the motor and visual channels concurrently. We expect that rake cursor will be valuable for any user but especially people having limited movement capacity. Rake cursor can also solve problem arising in multi-display setups.

In the future, we would like to compare the rake cursor technique to other techniques such as Ninja Cursors in a more realistic setup that could test the validity of our proposed interpretation. We also would like to study the impact of the form of the grid. We know that D_R impacts the performance: when it becomes larger than the screen, the rake cursor degenerates to the normal cursor. On the other hand, $D_R = 1$ pixel would mean that the pointing is done only by sole the gaze channel. Since both limit case are not efficient, an optimum D_R value must exists somewhere in between. The determination of this optimum, and more generally the impact of the form of the rake on its efficiency will be some of the next question we will investigate.

ACKNOWLEDGEMENTS

The authors would like to thank N. Mandran (MARVELIG) and B. Meillon (LIG-MULTICOM) for helping with the statistics and providing the gaze tracker. Part of this work is performed within the OpenInterface consortium (EU STREP FP6-35182).

REFERENCES

1. P. Baudisch, E. Cutrell, K. Hinckley, and R. Gruen. Mouse ether: accelerating the acquisition of targets across multi-monitor displays. In *proc. Ext. Abst. ACM CHI'04*, pages 1379–1382, 2004.
2. H. Benko and S. Feiner. Multi-monitor mouse. In *Ext. Abst. ACM CHI'05*, pages 1208–1211, 2005.
3. P. M. Fitts. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psycology*, 47:381–391, 1954.
4. W. E. Hick. On the rate of gain of information. *Quarterly Journal of Experimental Psychology*, (4):11–26, 1952.
5. R. J. K. Jacob. What you look at is what you get: eye movement-based interaction techniques. In *proc. ACM CHI'90*, pages 11–18, 1990.
6. M. Kobayashi and T. Igarashi. Ninja cursors: using multiple cursors to assist target acquisition on large screens. In *proc. ACM CHI'08*, pages 949–958, 2008.
7. M. Kumar, A. Paepcke, and T. Winograd. EyePoint: practical pointing and selection using gaze and keyboard. In *proc. ACM CHI'07*, pages 421–430, 2007.
8. I. S. MacKenzie. A note on the information-theoretic basis for Fitts' law. *Journal of Motor Behavior*, 21:323–330, 1989.
9. L. E. Sibert and R. J. K. Jacob. Evaluation of eye gaze interaction. In *proc. ACM CHI'00*, pages 281–288, 2000.
10. S. Zhai, C. Morimoto, and S. Ihde. Manual and gaze input cascaded (MAGIC) pointing. In *Proc. ACM CHI'99*, pages 246–253, 1999.