The Implicit Fan Cursor: A Velocity Dependent Area Cursor

Xiaojun Su
City University of Hong Kong
Hong Kong
xiaojun.su88@gmail.com

Oscar Kin-Chung Au
City University of Hong Kong
Hong Kong
kincau@cityu.edu.hk

Rynson W.H. Lau
City University of Hong Kong
Hong Kong
rynson.lau@cityu.edu.hk

ABSTRACT

We present the Implicit Fan Cursor (IFC) – a novel target pointing technique using a cursor with a fan-shape activation area. The IFC couples the cursor's activation area with its velocity, i.e., the speed and direction of the mouse motion, behaving like a 2D spotlight cursor at low speed and a circular area cursor at high speed. Thus, it enables the user to precisely acquire distant targets at low speed and easily acquire nearest targets at high speed, without explicit mode switching. This technique minimizes cursor movement, while taking into consideration of the precision of cursor movement at different speeds. It also ensures that only one target is captured at any time. The results of our controlled experiments show that the IFC outperforms the point cursor and the area cursor techniques, particularly in terms of cursor moving distance, and that its performance can be accurately modeled using the Fitts' law.

Author Keywords

Area cursor; implicit fan cursor; velocity-aware pointing; Fitts' law.

ACM Classification Keywords

H.5.2 User Interfaces: Graphical user interfaces (GUI), Interaction styles; I.3.6 Methodology and Techniques: Interaction techniques

General Terms

Human Factors; Design

INTRODUCTION

In graphical user interfaces (GUIs), target pointing is a fundamental task for acquiring interface components such as buttons, menu items and icons. With the increase in both size and resolution of computer displays, it becomes less efficient for a user to acquire small interface elements over the large display screen with the traditional point cursor. Several techniques have been proposed in recent years to address this problem and to improve pointing performance. One of the approaches is to reduce cursor movement by directly altering the cursor or target locations [1] [2] [12]. These techniques perform better than the point cursor in a sparse desktop environment. However, they are sensitive to the density and layout of the interface components, and their performance degrades if the target to be captured has multiple nearby objects. As it is common to have non-uniform target distributions and clusters of small targets in GUIs, these techniques may not always provide a performance improvement over the traditional point cursor.

Recently, there are some techniques proposed for dense target environments. They include techniques that are based on expanding the size of the targets [9] [10] [21], dynamic control-display ratio [5] and multiple cursors [17]. A promising technique developed from the area cursor [16] [25] is the Bubble cursor [11]. It dynamically adjusts the cursor's activation area such that only the closest target is captured. This is equivalent to expanding the boundary of each target to the Voronoi cell with the target center being the cell center, such that the Voronoi tessellation defined by all targets fills the whole desktop space. This maximizes the effective sizes of all targets. To allow the selection of the empty space between targets and reduce rapid changes of cursor size, another technique based on the area cursor, called Dynaspot [8], couples the cursor's activation area with its speed, behaving as a point cursor at low speed and a circular area cursor at high speed. However, this technique limits the maximum size of the circular area, and thus the cursor still needs to travel a long way to access distant targets even in a sparse desktop environment.

In this paper, we present the Implicit Fan Cursor (IFC), a new type of area cursor that couples the cursor's activation area with its velocity, i.e., its speed and direction of motion, as illustrated in Figures 1 and 2. It has a fan-shape activation area to allow the user to capture the target easily with minimal cursor motion (Figure 1(b)). The spanning angle of the cursor's activation area grows as a function of its speed to a maximum span of 180 degrees, and the orientation of the activation area is determined by the cursor's moving direction (Figure 2). At low speed, the fan cursor behaves as a spotlight cursor, enabling the user to precisely acquire distant targets (e.g., point B in Figure 2), while at high speed, it behaves as a semi-circular area cursor, allowing the user to easily approach the targets along the specified rough orientation (e.g., point D in Figure 2). To isolate the intended target from others, the spotting distance of the cursor is dynamically and automatically adjusted such that at any time only one target is captured.

After reviewing the related work, we discuss the design and implementation of the IFC, and evaluate its performance. Results show that the IFC outperforms the traditional point cursor and some popular area cursor techniques in both sparse and dense desktop environments. Finally, we show that the performance of the IFC can be modeled with the Fitts' law.

CHI 2014, April 26–May 1, 2014, Toronto, Ontario, Canada.
Copyright © 2014 ACM 978-1-4503-2473-1/14/04..$15.00.
http://dx.doi.org/10.1145/2556288.2557095

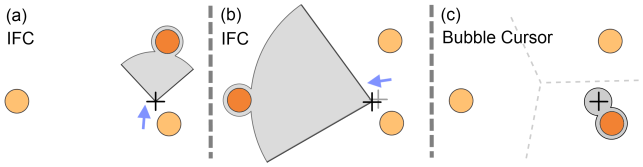

Figure 1. (a) The IFC's activation area is coupled to the cursor's velocity. (b) It allows the user to select a distant target with small cursor movement. (c) The Bubble Cursor always captures the closest target without considering the cursor's motion.

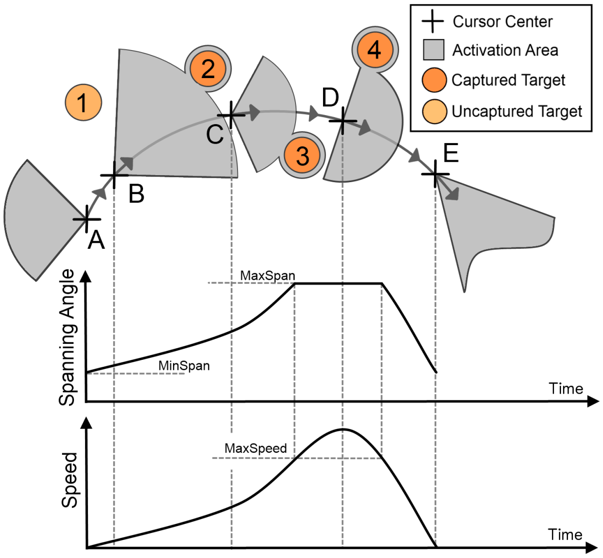

Figure 2. The spanning angle of the IFC is dynamically changing according to the cursor speed, and its orientation is determined by the cursor's moving direction.

RELATED WORK

The Fitts' law is typically used to study pointing in GUIs [20] [24]. It is a model for predicting the movement time MT in pointing tasks as:

$$MT = a + b \cdot \log_2(\frac{A}{W} + 1) \qquad (1)$$

where A is the distance (or amplitude) between the cursor and the target. W is the target width. a and b are two empirically determined constants, depending on hardware configuration and user behavior. Eq. 1 indicates that the movement time decreases if the size of the target increases or the distance needed to travel to acquire the target decreases. Hence, we may improve the efficiency of pointing tasks either by reducing the movement amplitude A, increasing the target width W, or a combination of both. In this section, we review these three approaches for improving pointing efficiency.

Reducing the Movement Amplitude

Jumping Cursors: These methods attempt to reduce the movement amplitude A by directly altering the cursor or target locations. The drag-and-pop technique [2] temporarily brings the potential targets closer to the cursor when an object is being dragged. The object pointing technique [12] ignores the empty spaces among the potential targets, and the cursor can jump from one selectable target to another. Another cursor jumping technique, the Delphian desktop [1], estimates the user's intention based on the movement direction and the peak velocity of the cursor and allows the cursor to jump over obstructive objects. All these techniques are sensitive to the target layout and density. Their performances are also heavily dependent on the accuracy of the target prediction algorithms used. In [18], an improved endpoint prediction method is proposed. It can predict target regions with an accuracy of up to 42%. Our method is similar to the jumping cursor techniques in that we also aim at reducing the cursor travel distance in a sparse desktop. However, instead of jumping among the potential targets, our method provides a smooth cursor movement with intuitive orientation control.

Multiple Cursors: These methods employ multiple cursors for target selection. The Ninja Cursor [17] consists of multiple cursor instances distributed over the screen and concurrently controlled by the same input device. This essentially reduces the movement distance, i.e., between the target and its nearest cursor. Another multi-cursor technique, the Rake Cursor [6], captures also the gaze position to help identify the current active cursor. The main limitation of the multi-cursor techniques is that they increase the total number of visual elements on the screen, leading to possible visual distraction. The need to search for the active cursor introduces extra focus switching and may increase the overall interaction time.

Increasing the Target Width

Increasing the target size: These methods attempt to speed up target pointing by directly increasing the size of the potential targets. The len-based cursors [13][23] magnify nearby objects in the visual space, i.e., screen space, where the information is displayed, but not in the motor space, i.e., device space, where the physical movement takes place. Thus, it still takes a similar amount of movement time for the cursor to reach the target. Other methods involve increasing the target size in both visual and motor spaces when the cursor approaches any potential targets. In [21], target expansion is shown to facilitate pointing even if it occurs as late as after 90% of the movement has been completed. Experiments in [9] show that for small targets, visual expansion in unaltered motor-space has similar performance gains as enlarged motor-space.

Area Cursors: These methods attempt to speed up target pointing by using an area cursor with a larger activation region [16], instead of a single-pixel hotspot as in the traditional point cursor. This is similar to increasing the effective target size. It makes pointing easier, but may capture multiple objects, leading to ambiguities. This problem can be solved by integrating a point cursor into the area cursor [25], or by interactively adjusting the cursor area on multi-touch input [22]. The Bubble cursor [11] improves the area cursor by dynamically resizing its activation area depending on the proximity of surrounding targets, such that only one target is captured at any time. This is equivalent to partitioning the empty space into a Voronoi tessellation defined by all targets, thus maximizing the activation area of each target. Starburst [3] uses a different partitioning method, better adapted to clustered targets with non-uniform distributions. Several variations of the Bubble cursor have been proposed [19] [14]. Recently a speed-dependent area cursor technique, named Dynaspot [8], is proposed. It behaves as a point cursor at low speed and an area cursor at high speed, and thus allows pointing anywhere in the empty space without requiring an explicit mode switching. Our approach is also based on a dynamic area cursor. However, unlike previous area cursor techniques, our technique utilizes both the cursor speed and orientation information to control the shape of the cursor's activation area. It allows overlapped (thus larger) effective target widths, and the user can seamlessly switch between the low-speed spot light mode and the high-speed area mode.

The Beam Cursor [26] is a pen-based technique, which exploits the sliding motion and allows selection of the nearby targets in the moving direction of the pen-tip. This is similar to our proposed approach in the sense that both techniques use the cursor moving direction as input. However, the Beam Cursor also requires the first landing position of the pen-tip to segment the screen into selectable regions. This information is typically not available with a traditional mouse.

The Combined Approach

There are several pointing techniques that are based on dynamically adjusting the control-display ratio, thus changing the underlying movement amplitude as well as target width. Sticky icons [25] and Semantic pointing [5] slow down the cursor when approaching a selectable target, thus increasing the underlying target width for easier acquisition. However both techniques are sensitive to the layout and density of the targets. While they work well in a sparse desktop environment, problems may arise when targets are clustered, as some targets located along the path to the intended target may slow down the cursor movement. The Vacuum [4] dynamically controls the positions of the prospective targets. Similar to IFC, the Vaccum uses a fan shaped area to determine the potential targets. The difference is that the Vaccum attempts to reduce the movement distance by drawing the targets towards the cursor. Animation is applied to smooth the visual state transition. This method significantly reduces the selection time of distant targets, while making no significant improvements when selecting nearby targets.

THE IMPLICIT FAN CURSOR (IFC)

The Implicit Fan Cursor is based on the area cursor technique [8] [11] [16] [25], and is designed to facilitate efficient target selection with minimum cursor movement. It dynamically adjusts the spanning angle and orientation of the cursor's activation area to model different selection styles, i.e., behaving as a spotlight to search for targets with precise orientation control or behaving as an area cursor to search for nearby targets with rough orientation control, without requiring any explicit mode switching. The IFC is inspired by a number of speed-dependent techniques that have successfully made use of the cursor's dynamic characteristics, including Dynaspot [8], interactive zooming [15] [13], and the sigma lenses [23]. While most area cursor techniques adopt a square-shape cursor [16] [25] or a circular cursor [11] [8], the IFC uses a fan-shape activation area to allow the user to specify its focusing orientation with the cursor's moving direction, and to ensure the nearest target along this moving direction to be captured first, as shown in Figure 1. To ensure that only one target is captured by the activation area at anytime, the radius of the fan is automatically set to the distance of the nearest target that the fan cursor reaches.

Previous area cursor techniques usually capture the nearest target, without considering the dynamic motion information of the cursor, e.g., the Bubble cursor as shown in Figure 1(c). The IFC allows the user to focus on targets along the movement direction and thus to select a target that may not be nearest to the cursor as shown in Figure 1(a). This means that the effective size of the targets can be larger than the corresponding Voronoi cells defined by all target centers. Such a Voronoi tessellation was previously considered as the "maximum" effective size of a given target set [11]. By utilizing the cursor's motion information, the effective areas of the targets could now be overlapped, allowing the user to pick a non-obscured target with minimum cursor movement and without the need to move the cursor to the Voronoi cell of the desired target, as shown in Figure 1(b). As a result, individual targets have extended effective widths that are much larger than their corresponding Voronoi cells. According to the Fitts' law, the IFC could facilitate more effective pointing performance than other area cursor techniques, which give relatively smaller effective widths.

To reduce visual distraction, we do not display the fan-shape activation area. We only highlight the captured target. We have conducted a preliminary user study on three display styles and confirmed that not displaying the activation area as shown in Figure 3(c) gives the best performance, compared to displaying a circular cursor (Figure 3(a)) or our fan cursor (Figure 3(b)). Hence, the fan-shape activation areas shown in Figures 2 and 1 are only for illustration. They are in fact invisible to the user. This is why we name the new method as the Implicit Fan Cursor. Note that it is easy for the user to identify the captured target as it is highlighted with an expanded contour, which has the same translucent color as the circular cursor shown in Figures 3(a) and 3(b). This is similar to the approach taken by the Bubble cursor [11].

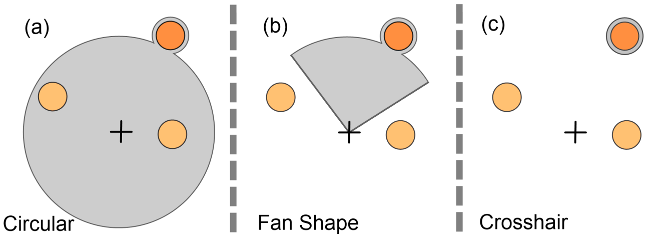

Figure 3. We have experimented with three display styles of the IFC: circular-shape, fan-shape and crosshair. Note that the captured target is always highlighted in all three display styles.

Velocity-dependent Behavior

As shown in Figure 2, the spanning angle of the activation area is set dynamically according to the cursor speed, bounded by the predefined minimum and maximum spanning angles ($MinSpan$, $MaxSpan$). The IFC facilitates efficient pointing in both sparse and dense target configurations. With a smaller spanning angle at low speed, users can have a more precise orientation control, reaching distant targets with only small cursor movements. For example, in Figure 2, the cursor at point A only needs to travel a short distance to point B to capture target 2. At high speed, the cursor behaves as a semi-circular area cursor and allows rough but effective target selection. For example, target 4 in Figure 2 can be captured by moving the cursor from point C to the target location roughly, as the spanning angle is maximized at high speed. We have conducted a preliminary user study to determine the most suitable minimum and maximum spanning angles, and found that setting $MinSpan$ and $MaxSpan$ to 90 degrees and 180 degrees, respectively, gives the best performance.

If the cursor speed is less than a speed threshold $MaxSpeed$, the spanning angle α of the fan-shape activation area is linearly proportional to the cursor speed s. When the cursor exceeds $MaxSpeed$, the fan reaches the maximum spanning angle $MaxSpan$. When the user stops moving the mouse, the spanning angle will linearly decrease to $MinSpan$ in one second if no targets are captured. Otherwise, the fan will keep its spanning angle unchanged to avoid undesirable deselection of the captured target. We have found that setting $MaxSpeed$ to 1200 pixels/s (or $33.2cm/s$) works fine in all our experiments.

Since the captured mouse positions can be noisy, in particular during mouse clicking, we apply a simple Exponentially Weighted Moving Average (EWMA) operation to smooth out the noisy samples collected. The EWMA operation averages recent velocity values with exponentially decreasing weights for past velocity values. Specifically, the smoothed velocity $\tilde{v}_t$ of the cursor at the current time t is computed as:

$$\tilde{v}_t = (1 - \lambda)\tilde{v}_{t-1} + \lambda v_t \tag{2}$$

where $0 \leq \lambda \leq 1$ is the weight. $\tilde{v}_{t-1}$ is the smoothed velocity value computed in the last time step, while v_t is the cursor's current velocity value. (In our experiment, we set $\lambda = 0.95$ and measured the velocity 30 times a second.) Note that the noise caused by mouse clicking typically has small magnitudes, which can be easily filtered out by the EWMA. With the smoothed velocity $\tilde{v}_t$, we may compute the current moving speed s_t, orientation o_t and the spanning angle α_t of the cursor as:

$$s_t = |\tilde{v}_t| \tag{3}$$

$$o_t = \tilde{v}_t / |\tilde{v}_t| \tag{4}$$

$$\alpha_t = min(MinSpan + \beta s_t, MaxSpan) \tag{5}$$

where β is a scaling factor for the spanning angle:

$$\beta = (MaxSpan - MinSpan)/MaxSpeed \tag{6}$$

Support for Pointing in the Empty Space

Pointing in the empty space is frequently needed in order to invoke many desktop operations such as region selection and deselection. Our IFC can naturally support pointing in the empty space without explicit mode switching using an approach similar to Dynaspot [8]. It is by setting the minimum spanning angle to zero and applying a minimum speed threshold such that the fan cursor is activated only when the cursor speed is higher than the given threshold. Hence, the spanning angle of the activation area grows from a point cursor when the user starts moving the mouse, and the user may select or click at the empty space when the cursor is not moving. Though it is an interesting topic to study the usability and performance of empty space pointing, the main focus of this paper is on the target acquisition performance of the IFC, and we leave this study as a future work.

PRELIMINARY STUDY 1: SPANNING ANGLE

Before comparing the proposed method with other pointing techniques, we have conducted a preliminary study to evaluate how different spanning angles may affect the performance of the proposed fan cursor technique.

Apparatus

This experiment was conducted on a PC with a Pentium Dual-Core 3.4GHz CPU, a 24" display screen of resolution 1920×1080 and an ordinary optical mouse. It was installed with MS Windows 7 and the default mouse configuration. The test program was written in C#.

Participants

6 participants (4 males and 2 females) of age 24 to 27 were recruited. All participants are experienced computer users and happened to be right-handed.

Procedure and Design

Sixteen circular candidate targets of 32 pixels in diameter were evenly distributed on a circle of 300 pixels in radius centered at the screen center, as shown in Figure 4. At the start of each pointing task, a pair of opposite start and goal targets drawn in yellow and green colors, respectively, were randomly selected from the candidate targets. Participants were asked to first click at the start target and then the goal target. When a target is captured, it changes to red color. In order to approximate a more realistic target acquisition scenario, four distracter targets were placed around the goal target. Two were placed along the direction of movement with one on each side of the goal target, while the other two were placed perpendicular to the direction of movement. The distracters were of the same size as the goal target and were 64 pixels away from it. Note that a task would be counted only if the user clicked at the goal target immediately after clicking at the start target. If he mistakenly clicked at a wrong target, e.g., one of the distracters, before clicking at the goal target, this task would be disqualified.

Our experiment was a one-factor within-participant design. Each participant performed the experiment in one session lasting for approximately half of an hour. The session is divided into groups of different spanning angle configurations. The participant was required to perform fifty pointing tasks in each configuration: ($MinSpan$, $MaxSpan$) $\in$ {(30, 30), (30, 90), (30, 180), (90, 90), (90, 180), (90, 270)}, and the configurations were ordered with balanced Latin-square. The movement time of each pointing task, defined as the time duration between the clicking of the start target and that of the

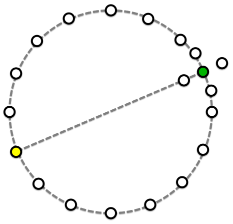

Figure 4. The screen layout for preliminary study 1. The yellow circle is the start target while the green circle is the goal target. The participant was asked to first select the start target and then the goal target.

(MinSpan, MaxSpan)	Mean Movement Time	Error Rate
(30, 30)	1052ms	9.7%
(30, 90)	992ms	13.6%
(30, 180)	907ms	14.5%
(90, 90)	904ms	9.4%
(90, 180)	851ms	9.7%
(90, 270)	862ms	10.0%

Table 1. Mean movement times and error rates for different spanning angle configurations.

goal target, and the overall error rate for each configuration were recorded for each participant. Before starting the experiment, each participant was given a single 5-minute warm-up session to get familiar with the interface. A total of 1,800 pointing tasks were performed in this experiment, with each participant performing a total of 300 tasks. Note that the implicit display style (i.e., crosshair only) was used in this preliminary study.

Results and Discussion

The effect of spanning angle configuration was explored by analyzing two dependent variables, movement time and error rate. Repeated measures analysis of variance shows that the use of different spanning angle configurations has a significant effect on the mean movement time ($F_{(5,25)}$=$65.32, p< 0.0001$) and the error rate ($F_{(5,25)}$=$2.31, p=0.0593$). Table 1 shows the mean movement time and the error rate of each configuration. We observe that the movement time is generally shorter for larger spanning angles in both $MinSpan$ and $MaxSpan$. This is due to the fact that a smaller spanning angle typically requires a more precise orientation control on the mouse, thus leading to longer manipulation time. However, some participants commented that a fan cursor with spanning angle greater than 180 degrees might capture targets behind the cursor, leading to non-intuitive jumping behavior. This explains that the pointing performance slightly degrades when the maximum spanning angle is higher than 180 degrees. As such, we suggest to limit the maximum spanning angle to 180 degrees, which ensures that only targets in front of the cursor will be captured. For the remaining experiments, we only show results from the (90, 180) configuration, unless specified otherwise.

Display Style	Mean Movement Time	Error Rate
Fan-shape	851ms	9.7%
Circular	844ms	13.6%
Implicit (Crosshair only)	819ms	7.3%

Table 2. Mean movement times and error rates for different display styles.

PRELIMINARY STUDY 2: DISPLAY STYLE

We have also conducted another preliminary study to evaluate how different display styles (i.e., how the cursor is visualized) may affect the performance of the IFC. Note that for all display styles we tested here, the underlying cursor activation area was always in fan shape, using the same spanning angle configuration of (90, 180); we only changed how the cursor is visualized on the screen.

Apparatus and Participants

Same as those in preliminary study 1.

Procedure and Design

We used the same pointing task as described in preliminary study 1. Each participant was asked to perform 50 pointing tasks in each of the three display styles: fan-shape, circular and implicit style, and they were ordered with balanced Latin-square. Before starting the experiment, each participant was given a single 5-minute warm-up session to get familiar with the interface. A total of 900 pointing tasks were performed in this experiment, with each participant performing a total of 150 tasks.

Results and Discussion

Table 2 shows the mean movement times and the error rates of different display styles. Some participants commented that both the fan-shape and circular cursors caused visual distraction, due to their continuous changing sizes. They drew user attention, causing unnecessary focus switching between the cursor and the targets. For the fan-shape cursor in particular, the fan-shape activation area distracted them from the selection of the desired target to the manipulation of the cursor orientation. Overall, we have found that the implicit cursor performs the best. Hence, we used this display style in the main experiment to compare our technique with other representative techniques.

MAIN EXPERIMENT: PERFORMANCE COMPARISON

After determining the spanning angles and the display style, we conducted the main experiment to quantitatively evaluate the performance of the IFC. We compared it with the point cursor, which served as a baseline, the Bubble cursor [11] and the Dynaspot [8] (with 32-pixel spotwidth), which are two of the most popular area cursors.

Apparatus

Same as those in preliminary studies 1 and 2.

Participants

Sixteen adult participants (8 males and 8 females) of age 23 to 28 were recruited. All participants are experienced computer users and happened to be right-handed.

Task and Procedure

We followed the same general procedure as the one used by [11] in testing the Bubble cursor – participants needed to select a goal target drawn in green color, which was surrounded by two groups of distracters, as shown in Figure 5. As in the preliminary study, when a target is captured by the cursor, it changes to red color. We created two groups of distracter targets. They were set to the same size as the targets and drawn as black outlined circles. One group of four main distracters are placed around the goal target to control the potential effective width of the goal target. We use a parameter RotaAngle to determine the locations of these four main distracter targets. If RotaAngle = 0^o, two are placed along the direction of movement with one on each side of the target, while the other two are placed perpendicular to the direction of movement, as in the preliminary study. If RotaAngle is not equal to 0^o, we simply rotate the four distracter targets about the goal target in clockwise direction by the corresponding angle. The second group of distracters are placed along the path from the start target to the goal target. They are distributed around a 20^o slice originating from the start target to the nearest main distracter so as to match a density condition DD. If DD = 1, we pack the slice with distracters such that the distracters do not overlap each other. If DD = 0.5, we place half of the number of distracters as in DD = 1 within the slice. If DD = 0, we place zero distracters. After placing the distracters within the slice, we then place similar density of distracters outside the slice. Figure 5 shows such an example.

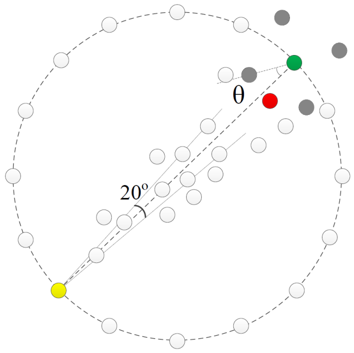

Figure 5. The screen layout of the main experiment. The yellow circle is the start target, while the green circle the goal target. The four grey circles are the main distracters, which are used to control the potential effective width. RotaAngle for controling the locations of the four main distracters is set to θ. Note that the grey color for the main distracters and all the lines shown in the diagram are for illustration only. In the experiments, there were no lines and all the distracter targets were black outlined circles.

Design

Our experiment was a 4×3×3×3×3×3 within-participant design with the following factors: (1) four techniques **TECH** for comparison: IFC, Dynaspot, Bubble cursor and Point cursor; (2) three amplitudes **A**: 256, 512 and 768 pixels; (3) three target widths **W**: 8, 16 and 32 pixels; (4)three **DistracterRatios**: 1.5, 3 and 5 (the ratio of the distance between each main distracter and the goal target to the target width); (5) three rotation angles **RotaAngle**: 0^o, 22.5^o and 45^o; (6) three distracter densities **DD**: 0, 0.5 and 1.

We note that some of the factors (i.e., DistracterRatio, RotaAngle, and DD) are related to the overall target density, and we could have used a general density factor in the experiment instead [7]. However, as the target effective width for different pointing techniques may be defined differently based on W, DistracterRatio and RotaAngle, we directly examined how these different factors affect the performance of each pointing technique here.

Factors	DF	DFDen	F	p
TECH	3	45	116.4	< 0.0001
W	2	30	394.9	< 0.0001
A	2	30	236.3	< 0.0001
DistracterRatio	2	30	132.3	< 0.0001
TECH×W	6	90	9.7	< 0.0001
TECH×A	6	90	3.6	< 0.005
TECH×DistracterRatio	6	90	13.2	< 0.0001
TECH×RotaAngle	6	90	3.7	< 0.005
TECH×A×DistracterRatio	12	180	5.2	< 0.0001
TECH×RotaAngle×DistracterRatio	12	180	4.6	< 0.0001

Table 3. The ANOVA results for the movement time of TECH×A×W×DistracterRatio×RotaAngle×DD. (DF is the degree of freedom, while DFDen is its denominator.)

Each participant performed the experiment in a single session, which was divided into groups of different pointing techniques. Each TECH group was further divided into 3 subgroups, one for each amplitude A. Each subgroup A was divided into 3 blocks, one for each target width W. In each block, participants completed a total of 27 combinations of DistracterRatio, RotaAngle and DD, presented in order. Each participant needed to perform 5 pointing tasks for each possible combination of all the factors. A total of 77,760 pointing tasks were thus included in the analysis. In the experiment, the TECH condition was counterbalanced with Latin-square, and A and W conditions were arranged in random order. Before using each technique, participants were given a single 5-minute warm-up session to get familiar with the technique and the tasks. They were given a 10-minute break after each TECH group. On average, each participant took approximately 3 hours to complete the whole experiment.

Results and Discussion

Movement Time and Covered Distance

Table 3 shows results of the repeated measures analysis of variance on movement time. We only show factors with significant effects of the ANOVA results and skip those insignificant ones. We can see that TECH, A, W and DistracterRatio have significant effects on the movement time. The average mean movement time was $1,148ms$ for IFC, $1,212ms$ for Bubble, $1,209ms$ for Dynaspot and $1,392ms$ for the point cursor. The IFC was the fastest among all four techniques. The following significant interactions are also observed: TECH×A, TECH×W, TECH×DistracterRatio and TECH×RotaAngle. This shows that different techniques were affected differently by the factors.

We have performed a Tukey HSD post-hoc test on the differences in mean movement time among the four techniques. Results show that IFC, Bubble and Dynaspot are significantly faster than the point cursor and there are no significant differences in movement time among IFC, Dynaspot and Bubble.

Figure 6 left shows the movement times of different techniques grouped by different target widths W. A Tukey HSD post-hoc test on the differences among the techniques TECH

W	Comparison
8	P < I, D, B
16	P < I, D, B
32	P < I, B

I = IFC, B = Bubble, D = Dynaspot, P = Point

Table 4. Significant differences for mean movement time MT among TECH by width W, where $a < b$ means TECH b is significantly faster than TECH a.

A	Comparison
256	P < I, B, D
512	P < I, B, D
768	P < I, B, D; D < I

I = IFC, B = Bubble, D = Dynaspot, P = Point

Table 5. Significant differences for mean movement time MT among TECH by Amplitude A, where $a < b$ means that TECH b is significantly faster than TECH a.

by width W (Table 4) shows that the movement time decreases significantly as the width increases for all four techniques. In addition, there are no significant differences in movement time among IFC, Dynaspot and Bubble for all W values, and the three techniques are significantly faster than the point cursor at most W values. However, when W=32, there is no significant difference between Dynaspot and the point cursor. This may be due to the fact that the ratio of potential effective width (potential EW) to target width W is relatively small for Dynaspot when the target width is large, and thus the advantage of Dynaspot over the point cursor becomes less obvious. Here, we have a hypothesis that the potential EW of the four techniques determines their performance ordering, i.e., a technique with a larger potential EW should perform better than the other techniques under the same amplitude A and width W.

Figure 6 middle shows the movement times of different techniques grouped by different movement amplitudes A. A Tukey HSD post-hoc test on the differences among the technqiues TECH by amplitude A (Table 5) shows that the movement time increases significantly as the amplitude increases for all four techniques. In addition, there are no significant differences among IFC, Dynaspot and Bubble for A = 256 or A = 512. The test reveals that IFC is significantly faster than Dynaspot when A = 768. The reason may be that Dynaspot has a fixed maximum activation area and thus a fixed potential EW (maximum activation area + W) for each W. For large amplitude, Dynaspot still has to move near to the goal target to capture it. This also indicates that the potential EW of each technique determines the performance ordering among the techniques.

Figure 6 right shows the movement times of different techniques grouped by different distracter ratios. A post-hoc test (Table 6) reveals that IFC, Dynaspot and Bubble are significantly faster than the point cursor for all distracter ratios, and that IFC is significantly faster than Dynaspot and Bubble for DistracterRatio = 5. This may be due to the fact that Dynaspot and Bubble have fixed potential efficient widths, although Bubble has a larger one than Dynaspot. However, the IFC can provide a larger potential EW than Bubble as a result of coupling movement-fan orientation.

DistracterRatio	Comparison
1.5	P < I, B, D
3	P < I, B, D; D< I
5	P < I, B, D; B, D < I

I = IFC, B = Bubble, D = Dynaspot, P = Point

Table 6. Significant differences for mean movement time MT among TECH by DistracterRatio, where $a < b$ means that TECH b is significantly faster than TECH a.

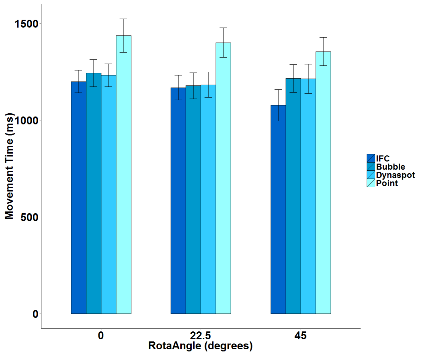

Figure 7. The movement times of different techniques grouped by RotaAngle.

Figure 7 shows the movement times of different techniques grouped by different RotaAngle values. The post-hoc test reveals that IFC, Dynaspot and Bubble are significantly faster than the point cursor for all RotaAngle values, and that there is no significant different among IFC, Dynaspot and Bubble (Table 7). To further explore the performance differences among the techniques by different factors, we have performed a post-hoc test of the techniques by both distracter ratios and RotaAngle values (Table 8). The test shows that for distracter ratio = 5 and RotaAngle = 45, IFC performs significantly faster than both Bubble and Dynaspot. Because in this situation, IFC can provide significantly larger potential EW than Bubble and Dynaspot and thus performs significantly better. We have also found that for Distracter Ratio = 3 and RotaAngle = 45, there was not any significant difference between IFC and Bubble. A possible explanation is that IFC and Bubble are likely to first capture either one of the two main distracter targets, which are located between the start target and the goal target. However, the potential EW of Dynaspot is independent of the cursor location.

Figure 8 shows the mean potential EW of different techniques grouped by the combination of W values and distracter ratios. For the point cursor, the potential EW is the same as the target width. For Bubble, the potential EW is determined by the multiplication of the target width and the distracter ratio. For Dynaspot, the potential EW is constrained by the

RotaAngle	Comparison
0	P < I, B, D
22.5	P < I, B, D
45	P < I, B, D; B, D < I

I = IFC, B = Bubble, D = Dynaspot, P = Point

Table 7. Significant differences for mean movement time MT among TECH by RotaAngle, where $a < b$ means that TECH b is significantly faster than TECH a.

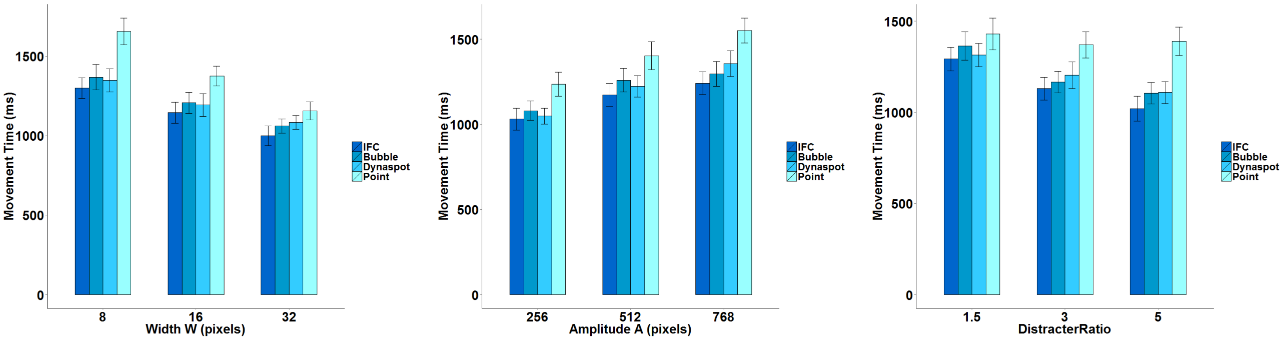

Figure 6. The movement times of different techniques grouped by Width (Left), Amplitude (Middle) and Distracter Ratio (Right). In this and all later charts, error bars represent intervals of one standard deviation.

DistracterRatio; RotaAngle	Comparison
1.5; 0	-
1.5; 22.5	-
1.5; 45	-
3; 0	P < I, B, D
3; 22.5	P < I, B, D
3; 45	P < I, B, D; D < I
5; 0	P < I, B, D; B, D < I
5; 22.5	P < I, B, D
5; 45	P < I, B, D; B, D < I

I = IFC, B = Bubble, D = Dynaspot, P = Point

Table 8. Significant differences for mean movement time MT among TECH by DistracterRatio and RotaAngle, where $a < b$ means that TECH b is significantly faster than TECH a.

target width and the size of the activation area. Unlike Bubble and Dynaspot, the potential EW of IFC is determined by both the distracter ratio and the cursor velocity, and is always larger than than of Bubble, which was previously considered as the maximum effective width. Since it is difficult to define the potential EW of IFC, we approximate it in each selection task as the distance between the cursor and the target at the time when a participant clicks and selects the target correctly. Note that this approximation is posteriori and is always smaller than the priori potential EW defined by the target configuration when IFC is being used. Nevertheless, we can see that for all W values and distracter ratios, the posteriori effective widths of IFC are still larger than the (priori) potential EW of all other techniques, and IFC always has the smallest movement time, as shown in Figure 9.

We have also computed the covered distance, which is defined as the physical distance travelled by the cursor to acquire the goal target, of the four techniques. The overall mean covered distance was 467 pixels for IFC, 525 pixels for Bubble, 538 pixels for Dynaspot and 648 pixels for the point cursor. Figure 10 shows the mean covered distances of different techniques grouped by the combination of W values and distracter ratios. We can see that for all W values and distracter ratios, IFC has the smallest covered distance.

Error Rate

Table 9 shows results of the repeated measures analysis of variance on error rate. There is a significant effect for TECH and W on the error rate. The following interactions are also detected: TECH×A, TECH×W and TECH×A×DistracterRatio. A Tukey HSD post-hoc test was performed and no significant differences among IFC, Bubble and Dynaspot were detected. However, IFC was the most accurate among all four techniques. The overall mean error rate was 5.6% for IFC, 6.7% for Bubble, 8.5% for Dynaspot and 9.3% for the point cursor.

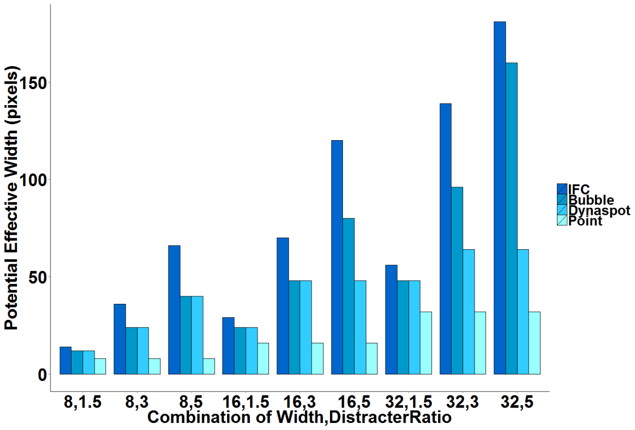

Figure 8. The mean potential effective width of different techniques grouped by the combination of Width and DistracterRatio.

Factors	DF	DFDen	F	p
TECH	3	45	6.82	<0.001
W	2	30	7.63	<0.001
TECH×A	6	90	3.83	< 0.001
TECH×W	6	90	2.96	< 0.05
TECH×A×DistracterRatio	12	180	2.550	< 0.005

Table 9. The ANOVA results for the error rates of TECH×A×W×DistracterRatio×RotaAngle×DD. (DF is the degree of freedom, while DFDen is its denominator.)

Qualitative Results

Participants were asked to rank the techniques by subjective preference in a post-hoc questionnaire. Thirteen participants ranked the IFC as their preferred technique, while three ranked Bubble as their preferred technique and IFC as second. Most participants complained about the visual distraction caused by the Bubble cursor as the cursor becomes extremely large in a sparse environment.

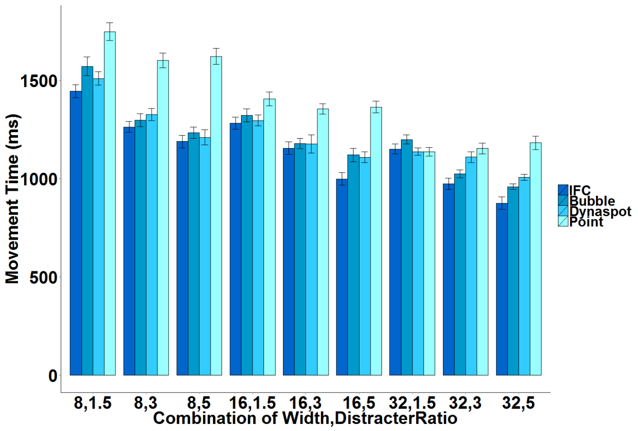

Figure 9. The mean movement times of different techniques grouped by the combination of Width and DistracterRatio.

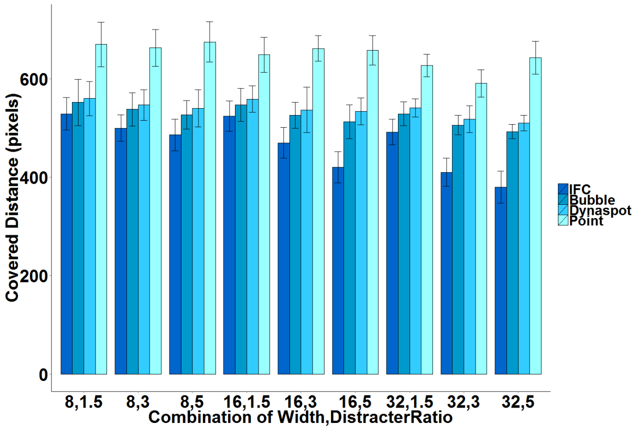

Figure 10. The covered distance of different techniques grouped by the combination of Width and DistracterRatio.

Technique	$MT = a + b.ID$		
	a	b	r^2
IFC	629	164	0.92
Bubble	609	170	0.89
Dynaspot	441	203	0.88
Point	222	239	0.90

Table 10. Linear fit: intercept, slope and r^2 values for different techniques.

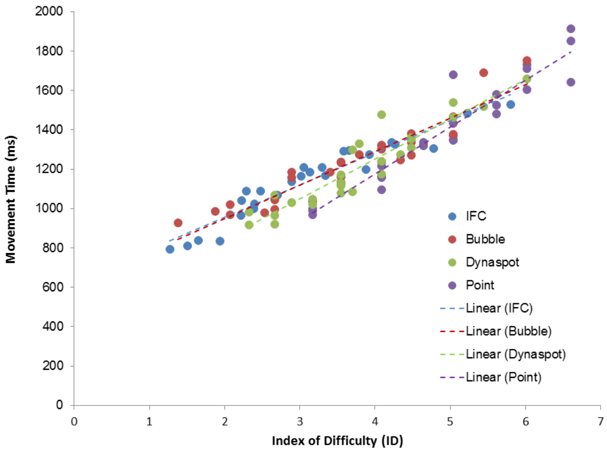

Figure 11. Linear regression of different techniques.

THE FITTS' LAW AND THE INDEX OF DIFFICULTY

Figure 11 plots the movement times of the four techniques as a function of the index of difficulty (ID). For IFC, Bubble and Dynaspot, we define the ID using the effective target width as $log_2(\frac{A}{EW} + 1)$. For Point, we define the ID using the target width as $log_2(\frac{A}{W} + 1)$. Note that the effective target widths, as defined in the previous section, are priori for Bubble and Dynaspot. However, the effective width for IFC was measured during the experiment, which is posteriori. We take the mean for each amplitude, width and distracter ratio, fitting 27 points to each technique as shown in Figure 11. Table 10 lists the intercept, slope and r^2 values for each technique. We can see that all techniques fit the linear model with resonable r^2 values. From Figure 11, we may conclude that the performance of the IFC can be modelled using Fitts' law.

CONCLUSION AND FUTURE WORK

This paper introduces a new target acquisition technique that is based on a velocity-dependent dynamic area cursor. Our key contribution is the utilization of the orientation and speed information of the cursor movement to dynamically update the cursor's activation area. This allows the effective width of the selectable targets to be larger than their corresponding Voronoi cells, which was previously considered as the maximum effective width in other area cursor techniques.

The implicit property of our fan cursor reduces unnecessary visual distraction and allows users to focus on the target being captured. During a typical target pointing operation, the user's focus may be jumping among the selectable targets. However, our technique provides a better control over the prediction-based jumping cursor techniques because our acquisition of targets is controlled by the underlying smooth cursor movement.

The results of our controlled experiments show that the IFC is a promising pointing technique with performance advantages over existing area cursor techniques, particularly in terms of cursor moving distance. In addition, the performance of our technique can be accurately modeled using Fitts' law.

As a future work, we would like to evaluate our method in broader application scenarios, e.g., empty space pointing and target selection in 3D virtual environment. We would also like to investigate the effectiveness of a content-aware fan cursor, which may adjust the spanning angle according to both cursor velocity and local target density.

ACKNOWLEDGEMENTS

We thank all the reviewers of this paper for the insightful comments and constructive suggestions. The work described in this paper was partially supported by two grants from City University of Hong Kong (CityU Reference No.: 7003056 and 7004155), and two GRF grants from the RGC of Hong Kong (RGC Reference No.: CityU 116010 and CityU 115112).

REFERENCES

1. Asano, T., Sharlin, E., Kitamura, Y., Takashima, K., and Kishino, F. Predictive interaction using the delphian desktop. In *Proc. ACM UIST* (2005), 133–141.

2. Baudisch, P., Cutrell, E., Robbins, D., Czerwinski, M., Tandler, P., Bederson, B., and Zierlinger, A. Drag-and-pop and drag-and-pick: Techniques for accessing remote screen content on touch-and pen-operated systems. In *Proc. INTERACT* (2003), 57–64.

3. Baudisch, P., Zotov, A., Cutrell, E., and Hinckley, K. Starburst: a target expansion algorithm for non-uniform target distributions. In *Proc. AVI* (2008), 129–137.

4. Bezerianos, A., and Balakrishnan, R. The vacuum: facilitating the manipulation of distant objects. In *Proc. ACM CHI* (2005), 361–370.

5. Blanch, R., Guiard, Y., and Beaudouin-Lafon, M. Semantic pointing: improving target acquisition with control-display ratio adaptation. In *Proc. ACM CHI* (2004), 519–526.

6. Blanch, R., and Ortega, M. Rake cursor: improving pointing performance with concurrent input channels. In *Proc. ACM CHI* (2009), 1415–1418.

7. Blanch, R., and Ortega, M. Benchmarking pointing techniques with distractors: Adding a density factor to fitts' pointing paradigm. In *Proc. ACM CHI* (2011), 1629–1638.

8. Chapuis, O., Labrune, J.-B., and Pietriga, E. Dynaspot: speed-dependent area cursor. In *Proc. ACM CHI* (2009), 1391–1400.

9. Cockburn, A., and Brock, P. Human on-line response to visual and motor target expansion. In *Proc. GI* (2006), 81–87.

10. Cockburn, A., and Firth, A. Improving the acquisition of small targets. In *Proc. People and Computers XVII: BCS Conf. on Human Computer Interaction* (2003), 181–196.

11. Grossman, T., and Balakrishnan, R. The bubble cursor: enhancing target acquisition by dynamic resizing of the cursor's activation area. In *Proc. ACM CHI* (2005), 281–290.

12. Guiard, Y., Blanch, R., and Beaudouin-Lafon, M. Object pointing: a complement to bitmap pointing in guis. In *Proc. GI* (2004), 9–16.

13. Gutwin, C. Improving focus targeting in interactive fisheye views. In *Proc. ACM CHI* (2002), 267–274.

14. Hertzum, M., and Hornbaek, K. Input techniques that dynamically change their cursor activation area: A comparison of bubble and cell cursors. *International Journal of Human-Computer Studies 65*, 10 (2007), 833–851.

15. Igarashi, T., and Hinckley, K. Speed-dependent automatic zooming for browsing large documents. In *Proc. ACM UIST* (2000), 139–148.

16. Kabbash, P., and Buxton, W. The "prince" technique: Fitts' law and selection using area cursors. In *Proc. CHI* (1995), 273–279.

17. Kobayashi, M., and Igarashi, T. Ninja cursors: using multiple cursors to assist target acquisition on large screens. In *Proc. ACM CHI* (2008), 949–958.

18. Lank, E., Cheng, Y., and Ruiz, J. Endpoint prediction using motion kinematics. In *Proc. ACM CHI* (2007), 637–646.

19. Laukkanen, J., Isokoski, P., and Räihä, K. The cone and the lazy bubble: two efficient alternatives between the point cursor and the bubble cursor. In *Proc. ACM CHI* (2008), 309–312.

20. MacKenzie, I. Fitts' law as a research and design tool in human-computer interaction. *Human Computer Interaction 7*, 1 (1992), 91–139.

21. McGuffin, M., and Balakrishnan, R. Fitts' law and expanding targets: Experimental studies and designs for user interfaces. *ACM Trans. on Computer Human Interaction 12*, 4 (Dec. 2005), 388–422.

22. Moscovich, T., and Hughes, J. Multi-finger cursor techniques. In *Proc. GI* (2006), 1–7.

23. Pietriga, E., and Appert, C. Sigma lenses: focus-context transitions combining space, time and translucence. In *Proc. ACM CHI* (2008), 1343–1352.

24. Soukoreff, R., and MacKenzie, I. Towards a standard for pointing device evaluation, perspectives on 27 years of fitts' law research in hci. *International Journal of Human-Computer Studies 61*, 6 (Dec. 2004), 751–789.

25. Worden, A., Walker, N., Bharat, K., and Hudson, S. Making computers easier for older adults to use: area cursors and sticky icons. In *Proc. ACM CHI* (1997), 266–271.

26. Yin, J., and Ren, X. The beam cursor: A pen-based technique for enhancing target acquisition. In *Proc. HCI* (2006), 119–134.