Additive Voronoi Cursor: Dynamic Effective Areas Using Additively Weighted Voronoi Diagrams

Jacky Kit Cheung, Oscar Kin-Chung Au, Kening Zhu

Additive Voronoi Cursor: Dynamic Effective Areas using Additively Weighted Voronoi Diagrams

Jacky Kit Cheung[1] and Oscar Kin-Chung Au[2] and Kening Zhu[2]

[12] School of Creative Media, City University of Hong Kong, Hong Kong
[1] ckitsa@gmail.com
[2] {kincau, keninzhu}@cityu.edu.hk

Abstract. We present Additive Voronoi Cursor (AVC) – a new cursor technique for target selection by dynamically resizing the area cursor based on the analysis of the two different phases of mouse movement: the ballistic and the correction phases during target selection. On-screen Targets can be divided into respective areas dynamically based on both target distribution and cursor velocity. We assumed that to select a target, a user will first perform ballistic/fast cursor movement aiming to the target roughly, then correct the cursor position with slower movement towards the desired target. Therefore, after the ballistic movement, the desired target would locate within the local region closed to the cursor. We defined Additive Weighted Voronoi Diagrams with selectable targets by assigning larger weights to the nearby objects right after the ballistic cursor movement. Therefore, the effective areas of the nearby objects are enlarged, and they can be selected more easily and quickly. We had compared our cursor technique with recent developed area-cursor methods. The results showed that our method performed significantly better on certain configurations.

Keywords: Area cursor, additively weighted Voronoi diagram, movement-phase-aware cursor, Fitt's law

1 Introduction

Target selection is a fundamental task for acquiring graphical-user-interface (GUI) components such as buttons, icons and menu options. Most screen-based applications, such as gaming and information visualization, still require frequent use of mouse selection. While manipulation tasks in design applications increase the frequency of using shortcuts, they still need to cooperate with mouse operations. With the increment in both size and resolution of computer displays, it becomes less efficient for a user to acquire small on-screen elements on the large display with the traditional cursor technique. In recent years, several techniques have been proposed to address this problem and to improve selection performance. One of the approaches is to reduce the cursor movement by directly changing the locations of the cursor or targets [1, 2, 12, 19, 21]. These techniques performed better than the traditional cursor technique in a sparse desktop environment. However, they are sensitive to the density and layout of

the on-screen selectable components, and their performance would degrade if the target to be captured was surrounded by 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 improve the target-selection performance over the traditional technique.

Some techniques for dense target environments have been proposed in recent researches, such as expanding the targets' size [9, 10, 23], dynamically controlling the display ratio [5], and applying multiple cursors [6, 18]. One promising technique developed from the area cursor [17, 29] is the Bubble cursor [11] and its variations [14]. The Bubble cursor dynamically adjusted the cursor's activation area such that only the closest target would be captured. This is equivalent to expanding the boundary of each target to the Voronoi region with the target center being the region center, so that the Voronoi diagram defined by all targets fills the whole screen space. The definition of targets' selectable regions in most of these area-cursor techniques only relies on Standard Voronoi diagram, and ignores the cursor motion. We believe that there is a potential design space for designing a new area cursor technique by considering targets' effective areas as dynamic Voronoi diagram which is defined by both target distribution and cursor motion.

In this paper, we present the Additive Voronoi Cursor (AVC), a new target-selection technique by dynamically resizing on-screen targets' effective areas based on the analysis of the two different phases of mouse movement: the Ballistic and the Correction phases. During the the process of target selection, we assume the user will perform a ballistic movement with a fast speed to bring the cursor to the desired target roughly, then adjust the cursor position with a slower correction movement towards the desired target. With AVC, the screen space was treated as an Additively Weighted Voronoi Diagrams (AWVD), and divided into respective areas dynamically, according to the target distribution and the cursor location at the time of entering the correction phase. In general, we assumed that the desired target should be located within the local region near to the cursor during the correction phase. We defined AWVD with selectable targets as Voronoi sites by assigning larger weights to the targets closer to the cursor in the correction phase, such that the effective area of the nearby targets would be enlarged as shown in Fig. 3, allowing easier and faster selection than ordinary Bubble-cursor-like methods. In this paper, we focus on the selection task in desktop computer setting. We believe our pointing techniques could be also applicable to VR/AR domains, as target selection by pointing operations is the common UI in VR/AR systems, and the bubble-like area cursor techniques are commonly used in these systems.

The rest of the paper is organized as follows: after reviewing the related work, we will present the design and implementation of two types of AVC - Additive Voronoi Cursor (AVC) and Additive Voronoi Cursor with Manhattan distance (AVC+M), and evaluate their performance with existing area cursor methods. Experiment results showed that our method performed significantly better among the tested area cursor methods. Finally, we will report that the performance of the AVC and AVC+M can be modeled with the Fitts' law.

2 Related Work

The Fitts' law is typically used to study target selection in GUIs [22, 27]. It is a model for predicting the movement time MT in target selection tasks as:

$$MT=a+b \cdot \log2(AW + 1) \quad (1)$$

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 selection technique, hardware configuration and user behavior. The equation indicates that the efficiency of selection tasks can be improved either by reducing the movement amplitude A, increasing the target width W, or a combination of both.

In early work, selection techniques based on jumping cursors/objects [1, 2, 12, 19] or multiple cursors [6, 18] were proposed to reduce the movement amplitude of target-selection tasks. However, these techniques are sensitive to the target layout and density. In addition, the jumpy interface and increased visual elements would lead to visual distraction, thus degrade user performance.

Area-Cursor techniques used an area cursor with a large activation region [17], instead of a single-pixel hotspot in the traditional point cursor to increase the efficiency of target selection. This is equivalent to enlarging the effective target size. They made selection easier, but may capture multiple objects at the same time, leading to ambiguities. This problem can be solved by integrating a point cursor into the area cursor [29], or by interactively adjusting the cursor area on multi-touch input [25]. The Bubble cursor [11] tackled this problem by dynamically resizing the activation area of a particular target based on the proximity of surrounding targets, by which only one target would be captured at a time. This is equivalent to partitioning the screen space into a Voronoi diagram which is defined by all targets, thus maximizing the overall activation area of all targets. Starburst [3] used a different partitioning method, which adapted to clustered targets with non-uniform distributions. Several variations of the Bubble cursor have been proposed [16, 20]. Recently, speed-dependent area-cursor techniques [8, 28] were proposed. Chapuis et al. [8] developed DynaSpot, in which the cursor behaved as a point cursor at low speed and an area cursor at high speed, and thus allowed pointing anywhere in the empty space without requiring an explicit mode switching. Note that in all these area-cursor techniques the targets' potential effective areas were defined by standard Voronoi diagram with Euclidean distance function. In this paper we consider to explore the design space of area cursor techniques by define the effective areas as dynamic Voronoi diagram with a different distance function.

More recently, an area-cursor technique with dynamic target effective areas, named IFC, was proposed [28]. IFC adopted a dynamic fan-shape area cursor which was based on both the speed and moving direction of the cursor movement, such that only the targets in front of the moving direction of the cursor would be selectable, potentially allowing larger targets' effective widths and shorter cursor moving distance. However, the response of target capturing of IFC is different from Bubble-like cursor techniques, and it may be difficult for users to control the cursor orientation smoothly without sufficient practice.

Guillon et al. [14] studied the impact of display styles on the performance with Bubble-like area cursors. The experiment results showed that showing the effective areas of all targets led to the best performance. However, the study only focused on static effective areas, thus the results may not apply to area cursors with dynamic effective areas such as IFC and our proposed technique. Another interesting finding by Guillon et al. [14] is that the area cursor with Manhattan distance metric has a similar performance as the original Euclidean-distance-based area cursor. Therefore, we also considered the Manhattan distance metric in our area cursor design, and studied the performance difference between our Manhattan distance-based and Euclidean distance-based cursor techniques.

Several target selection techniques were based on dynamically adjusting the control-display ratio, thus changing the underlying movement amplitude as well as target width. Sticky icons [29] and Semantic pointing [5] slowed down the cursor when approaching a selectable target, thus increasing the underlying target width for easier acquisition. However, both techniques were sensitive to the layout and density of the targets. While they may 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 brought prospective targets closed to the cursor. This method significantly reduced the moving distance of the cursor for selecting distant targets, but it required additional visual aids and complex cursor operations to control the scanning region.

3 Additive Voronoi Cursor

The Additive Voronoi Cursor is based on the area-cursor technique and is designed to support faster and easier target selection during the correction phase. We assumed that in a general selection task, the user will first perform the fast ballistic cursor movement to bring the cursor close to the desired target, followed by the slow correction movement to precisely select the desired target. We repartitioned the screen space immediately after the cursor movement changing from the ballistic phase to the correction phase, so that targets closer to the cursor will have relative bigger effective areas than those further away from the cursor. Different from the standard Voronoi partition used in the ordinary Bubble area cursor, we hypothesized that the dynamic partition of the additively weighted Voronoi diagram (AWVD) in AVC would improve the efficiency of the correction cursor movement.

Note that other literature [15, 24] considered that the whole target-selection task involves extra phases besides the ballistic and correction phases, such as the searching phase before the starting of mouse movement, and the verifying phase after the correction movement and before the mouse clicking occurs. However, in this work, we do not focus on these extra phases, as the time spending on them would not significantly affect by area cursor techniques. Therefore, we designed AVC to mainly determine the ballistic and the correction phases based on the cursor moving speed, and repartition the effective areas as an AWVD immediately after the change from the ballistic phase to the correction phase.

We defined the cursor movement enters the ballistic phase once the cursor speed is faster than a speed threshold s_B, and the cursor movement enters the correction phase once its speed is slower than another speed threshold $\boldsymbol{s_B}$, such that $\boldsymbol{s_B} > \boldsymbol{s_C}$. An example of such phase switching is demonstrated in Fig. 1. We have conducted a preliminary study to optimize the thresholds values for AVC, which is described in the next section.

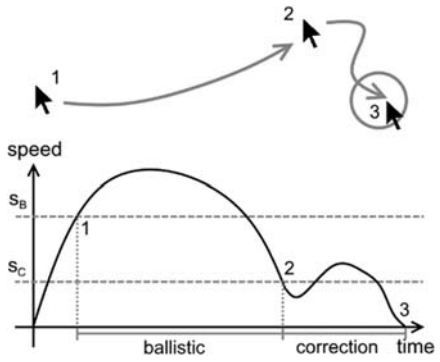


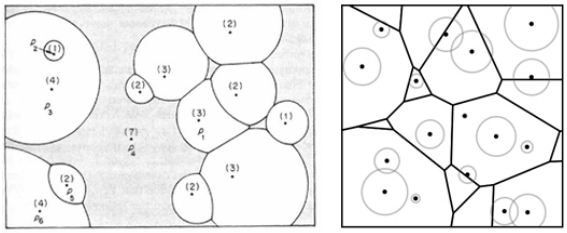
Fig. 1. Target selection example and phase detection. A fast ballistic movement is performed first to roughly bring the cursor close to the desired target (1 -> 2), followed by the slow correction step to exactly select the desired target (2 -> 3).

3.1 Selecting Distance Functions for Voronoi Diagrams

We have considered different methods of 2D partition for better partitioning the screen space with the centers of selectable targets as the set of sites. In particular, we focused on the definition of the distance function $\mathrm{d}(p, q)$ that used in the computation of Voronoi Diagram, where p is a 2D point on the screen and q is the center of one of the selectable targets. The distance function mainly controlling the sizes of the partitioned regions which would be the effective area of the corresponding target. Standard Voronoi diagram directly uses the Euclidean distance function, by altering this we could control the target effective width to a certain extent. One possible way to alter the original Euclidean distance function used in standard Voronoi diagram computation is to apply weights to each site multiplicatively:

$$d_m(p, q_i) = \frac{\|p - q_i\|}{w_i} \tag{2}$$

where w_i is the weight assigned to the target q_i. In general, a bigger weight will result in a bigger corresponding Voronoi region. However, a multiplicatively weighted Voronoi diagram (MWVD) [7] often contains non-convex regions, as well as enclave and exclave regions (Fig. 2a). Therefore, using MWVD for the effective areas in target selection might make the selection process unpredictable and unintuitive, because the enclave and exclave shapes might introduce unexpectable switching of captured targets during the selection process.



(a) **(b)**

Fig. 2. (a) Example of multiplicatively weighted Voronoi diagram, the numbers next to the sites are the corresponding assigned weights. **(b)** Example of a power diagram, the circle sites are displayed in grayed color.

Another possible weighting method is the power diagram [7] which replaces the point sites with circle sites. The circle radius r_i of a circle site could be considered as the site weight for the distance function.

$$d_p(p, q_i) = \|p - q_i\|^2 - r_i^2 \tag{3}$$

While the power diagrams could ensure straight region boundaries and convex Voronoi regions, one main drawback is that the site center could be outside the corresponding Voronoi regions (Fig. 2b). This could result in unselectable targets even when the cursor is close to them, particularly when the targets locations or the circle radius r_i are unevenly distributed. Therefore, we abandoned this weighting method.

Instead we decided to use the AWVD, to define the effective areas of the targets. The AWVD uses the weighted distance function given by

$$d_a(p, q_i) = \|p - q_i\| - w_i \tag{4}$$

Even some regions could be non-convex in the AWVD, every non-convex region is star-shaped with respect to its site (target center). This means that from the center q_i we can draw a line to any point in the region, and the line will be contained entirely within the region, ensuring the selectability of the target [7]. Also, since AWVD only contains connected convex and star-shaped regions, there will be no undesired enclave and exclave regions.

We adopted a gradient weighting function centered at the cursor to define the weight values. Therefore, the targets close to the cursor will receive relatively larger weights than distant ones, so their corresponding regions will be elongated radiationally from the cursor position (Fig. 3), allowing a larger effective width along with the possible moving directions from the cursor to the nearby targets. Specifically, we used a linear gradient function (Fig. 4), centered at the cursor position when the cursor movement just enters the correction phase, to define the additive weights for repartitioning the screen space. Fig. 3 shows the different effective areas defined by the unweighted Voronoi diagram and the AWVD, notice that in the latter case how longer effective areas are defined in the directions from the cursor to nearby targets. This enlarged effective area allows users to select nearby targets easier and faster.

To provide the best performance, the parameters of this linear gradient function should be selected based on the screen resolution and other hardware configurations, as well as the user preferences. The weight assignment would affect the shape changes of the effective regions of selectable targets. If the scaling factor is too large, then two nearby targets have large differences in their weights, the target effective area with larger weight would dominance the nearby region and the region of the other target will disappear. Therefore, we select a scaling factor as shown in Fig. 4 that would never cause such undesired effects and produce enough elongation of effective areas of targets near to the cursor position when the cursor entering the correction phase. We adopted these settings in our empirical experiment, resulting in the desired Voronoi-diagram partition with enlarged effective areas for nearby targets. Therefore we used these empirical settings in our controlled user study.

It may be possible that the user's desired target is not located close to the cursor right after switching to the correction phase. The user can use another ballistic movement to move the cursor to another screen region roughly, or slowly move the cursor and search the desired targets at another region. In either case, the previous "repartitioned" effective areas will not affect the next selection operation, as only the local region near to the old cursor position (at the time switching to correction phase) was repartitioned by the additively weighted Voronoi diagram.

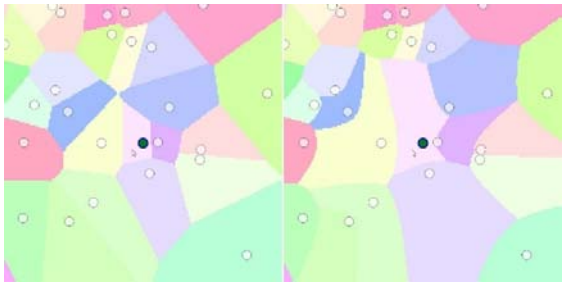


Fig. 3. Different effective areas defined by the unweighted Voronoi diagram **(left)** and the additive weighted Voronoi diagram **(right)**.

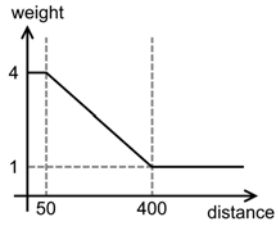


Fig. 4. The weighting function used for our cursor technique. The x-axis is the distance from the cursor position to the target center (in pixels), and the y-axis is the weight values assigned to the target.

3.2 Speed Computation

To detect the switching between the ballistic and correction phases, one needs to compute the current ongoing cursor speed. Since the captured mouse positions can be noisy, we apply a simple Exponentially Weighted Moving Average operation to filter the noisy samples, similar to the velocity smoothing method adopted in [28]. Specifically, the smoothed ongoing cursor velocity $\tilde{v}_t$ at current time t is computed as

$$\tilde{v}_t = (1 - \lambda)\tilde{v}_{t-1} + \lambda\tilde{v}_t \tag{5}$$

where $0 \leq \lambda \leq 1$ is the weight, $\tilde{v}_{t-1}$ is the smoothed velocity in the last time step, while $\tilde{v}_t$ is the current unsmoothed velocity value. We need a smoothed cursor speed computation to remove noisy speed samples that caused by the raw mouse input (particularly the zero speed values produced when two successive cursor samples have the same position), therefore cursor speed smoothing is necessary for our cursor technique. We have tested with both 30 and 60 times per second settings for cursor speed computation with moving average smoothing, and found that both settings can produce satisfied and smooth speed estimation, therefore we choice 30 fps for our experiments. The weight λ is set to 0.95 and the velocity was sampled 30 times per second. With the smoothed velocity, the speed value is just the magnitude of the velocity vector $s_t = |\tilde{v}_t|$.

3.3 Additional Voronoi Cursor with Manhattan Distance Metric (AVC+M)

As reported in [14] that the area cursor technique with the Manhattan distance-based Voronoi diagram (MTE) gives comparable performance than the Euclidean distance-based area cursor (VTE), it is interesting that what would the performance of AVC be if adopting the MTE. Therefore, in our experiment, we also involved AVC with the Manhattan distance metric (AVC+M). The distance function of AVC+M is given as

$$d_a(p, q_i) = \|p - q_i\|_M - w_i \tag{6}$$

where $\|\cdot\|_M$ is the Manhattan distance metric. Notice that different from AVC, AVC+M always generates straight line region boundaries in vertical, horizontal and 45°-oriented directions (Fig. 5), which may result in different performance when comparing with AVC.

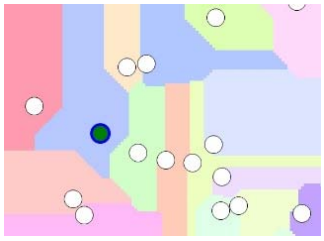


Fig. 5. AVC+M always generates straight line region boundaries in vertical, horizontal and 45°-oriented directions.

3.4 Implicit Display Style

The results in [14] showed that explicitly showing the all static target effective areas gives better performance than other displaying styles. However, this does not apply to AVC since the target effective areas are dynamically updated according to the user's interaction. Moreover, displaying effective areas of all targets would overwhelm selectable targets and other UI elements, thus would not be appropriate for many applications. According to the findings of recent pieces of literature [15, 28], an area cursor with dynamic target effective areas should hide the effective areas for better performance, as displaying the effective areas would distract users' attention and degrade target selection speed. Therefore, we adopted the implicit display style and hid the dynamic target effective areas, so that users will not aware of the update of the underlying Voronoi diagram when the cursor movement is switched from ballistic phase to correction phase. Also, for a fairer comparison, we hide the target effective areas for all cursor techniques being tested in our experiment. The Voronoi regions displayed in the figures and the accompanying video were only for visualization purposes only.

4 Experiment

4.1 Preliminary Study: Ballistic and Correction movement speed

Since AVC dynamically resizes the effective areas of the on-screen objects right after the Ballistic phase, it is important to robustly detect the phase change without affecting the user performance. We have conducted a preliminary study to optimize the threshold values of the mouse speed for detecting Ballistic and Correction phases in our proposed AVC technique. Note that the speed thresholds found in this preliminary study were only optimized for the specified screen resolution and mouse configuration. For other screen/mouse configurations, the optimized threshold values may be different.

Apparatus. This study was conducted on a PC Notebook with a Quad-Core 2.7GHz CPU, a 17" display screen of resolution 1920×1080 and a Logitech optical mouse (DPI = 1000). It was installed with MS Windows 10 and the default mouse speed setting. The "Enhance Pointer Precision" function is turned off, thus no cursor acceleration was used in the experiment.

Participants. We recruited 9 participants (6 males and 3 females) of 21 to 30 (Mean = 27.2, SD = 2.86) within the university. All participants are right-handed frequent computer users, and they use their dominant hand to control the mouse.

Procedure and Design. Previous research [7] has proved that the distractors along the start and goal target would not affect the performance of a task selection task, but only the distractors close to the goal targets would affect. Therefore this study followed the settings from [8, 28], with the aim of studying how different threshold

speeds for classifying the Ballistic and Correction movements would affect user performance of target selection using AVC under various configurations of the moving distance/amplitude and the crowdedness. Sixteen circular candidate targets of 8 pixels in radius were evenly distributed on a large circle with the possible diameter of 256, 512, and 768 pixels which defined the movement amplitude. The large circle was placed at the screen center, and the centers of the candidate targets lied on the circumference of the large circle, as shown in Fig. 6a. At the start of every selection task, the software randomly marked a pair of opposite targets as start and goal targets, and drew the start target in yellow, and the goal target in green color, respectively. When a target was captured by the cursor, it was surrounded by a blue stroke. Participants were instructed to click the start target first and then select the goal targets as quickly and as accurately as possible using AVC. In order to mimic a more realistic target-selection scenario, another four targets of the same size were placed around the goal target as distractors. Two of them were placed along the direction of movement (one before and one after the goal target), and the other two distractors were placed perpendicular to the direction of movement (one on the left and one on the right). The distractors were all equally away from the goal targets with a distance of three times the target's diameter (48 pixels).

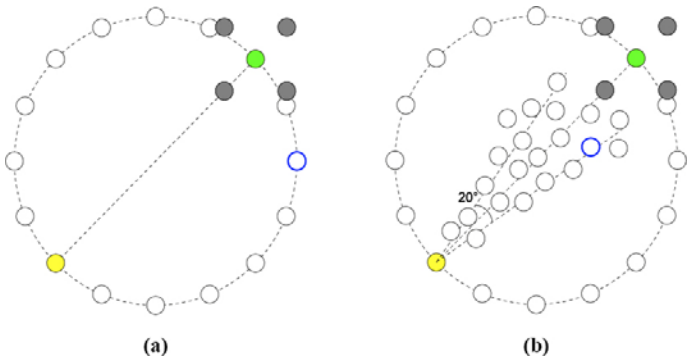


Fig. 6. (a) The screen layout for the preliminary study. The yellow circle is the start target while the green circle is the goal target. The target being captured would be surrounded by a blue stroke. The four grey circles are the main distractors to control the effective width of the goal target. Note that the color and the dashed line shown above is for illustration purpose only. Except for the start target, goal target and the selected target as shown above, other targets are 1-pixel black outlined circles in white color without any lines. **(b)** The study interface of the main experiment, in addition to Figure 6a. The second group distractors are distributed a 20° slice beginning from start target to the nearest main distractor.

We designed our experiment as a 3×3×3 within-participant study with the following factors: (1) three amplitudes (in pixels), **A**: 256, 512, 768; (2) three speed thresholds for identifying the ballistic phase (in pixels/s), $\boldsymbol{s_B}$: 80, 100, 120; and (3) three speed thresholds for identifying the correction phase (in pixels/s), $\boldsymbol{s_C}$: 30, 50, 70. Based on the screen resolution and mouse configuration, we achieved the weighting function (Fig. 4) and $\{\boldsymbol{s_B}, \boldsymbol{s_C}\}$ as {100, 50}. We offset a positive and negative value to the $\{\boldsymbol{s_B}$,

s_C} to provide those factors with 3 different values, to form the predefined factor values of the experiment. The experiment was divided into groups of different combinations of the s_B and s_C, which were ordered with a Latin-square-based counterbalance. In each group, participants performed the selection tasks in different values of **A**. Each **A** value repeated 16 times with the 16 pairs of opposite start and goal targets appearing in a randomized order. The whole experiment lasted for approximately 30 minutes for each participant. While we noted that the movement distance/amplitude might potentially affect the selection performance, in this study we mainly focused on how different speed thresholds would affect the user performance of target selection.

The time duration between clicking the start target and selecting the goal target was measured as task-completion time, and the overall error rate of each participant (i.e. the percentage of failed tasks) was recorded. A task would be marked as failed if the participant selected an incorrect goal target before selecting the correct one. If a participant selected an incorrect target, he/she still needed to select the correct target to complete the task. Before starting the selection tasks, the participants were given a 5-minute training session to get familiar with the user interface. A total of 3,888 selecting tasks were performed, with each of the users performing a total of 432 tasks.

Results and Discussion. With the **A**, s_B, and s_C as the independent factors, we statistically analyzed their effects on the task completion time and the error rate. Table 1 shows the variant result of mean movement time and error rate for each configuration. We have performed Bonferroni correction for multiple pairwise comparisons and checked the normality of the data for the experiments in this paper.

Task-Completion Time. Three-way ANOVA showed that the **A** ($F_{(2,3861)} = 327.2$, $p < 0.0005$) and s_C ($F_{(2,3861)} = 3.72$, $p < 0.05$) had a significant effect on the task-completion time, while there was a marginal effect of s_B on the task-completion time ($F_{(2,3861)} = 2.85$, $p = 0.058$). There was no significant interaction effect of these three factors on the task completion time. It was obvious that the task-completion time should increase as the amplitude increases, so we mainly focused on optimizing the combination of the s_B and s_C.

Table 1. Mean task completion times and error rates for different s_B and s_C.

(s_B, s_C)	Mean Task Completion Time				Error Rate			
	A256	A512	A768	Avg	A256	A512	A768	Avg
(80, 30)	725	847	1028	867	12.5%	6.9%	14.6%	11.3%
(80, 50)	740	845	1031	872	7.0%	6.4%	20.4%	11.3%
(80, 70)	741	903	1029	891	9.7%	9.7%	15.3%	11.6%
(100, 30)	738	860	985	861	6.9%	4.9%	9.7%	7.2%
(100, 50)	739	879	991	869	5.6%	6.3%	9.8%	7.2%
(100, 70)	761	874	1007	881	9.7%	9.0%	9.0%	9.3%
(120, 30)	777	857	990	875	13.9%	7.0%	7.0%	9.3%
(120, 50)	748	880	1021	882	9.0%	5.6%	11.3%	8.6%
(120, 70)	769	902	1037	903	5.6%	9.9%	9.0%	8.2%

Post-hoc pairwise comparison showed that across all three different amplitudes, s_C = 30 yielded significantly shorter task-completion time than s_C = 50 ($p < 0.05$) and was marginally faster than s_C = 70 ($p = 0.052$). Configurations with s_B = 100 was marginally faster than 120 ($p = 0.052$), and slightly faster than 80.

Error Rate. Three-way ANOVA showed that s_B ($F_{(2,3861)} = 2.85$, $p < 0.005$) and A ($F_{(2,3861)} = 4.47$, $p < 0.05$) had a significant effect on the error rate, while there was no significant or marginal effect of the Correction Speed Threshold s_C on the error rate. There was no significant interaction effect of these three factors on the error rate.

Post-hoc pairwise comparison showed that condition s_B = 100 yielded significantly less error than s_B = 80 ($p < 0.05$), and slightly but not significantly less error than s_B = 120. The higher error rate occurred when s_B = 80 is mainly due to the incorrect movement mode switching during the correction phase, as the threshold is too low to trigger, thus the effective areas of targets were updated unexpectedly, and the selection tasks became more difficult to complete.

Considering the effect of both thresholds on the task-completion time and the error rate, we finally chose s_B = 100 and s_C = 30 for the main experiment.

4.2 MAIN EXPERIMENT: PERFORMANCE COMPARISON

After determining the speed thresholds for the ballistic and correction phases, we conducted the main experiment to evaluate the performance of the proposed AVC and AVC+M with comparing to other area cursor methods, including Bubble Cursor [11], Manhattan Target Expansion (MTE) [14] and Implicit Fan Cursor (IFC) [28].

Apparatus. Same as those in the preliminary study.

Participants. 10 participants (7 males and 3 females) of age 19 to 38 (Mean = 27.6, SD = 5.27) were recruited within the university. All participants are right-handed frequent computer users, and they use their dominant hand to control the mouse. These participants did not participate in the preliminary study.

Task and Procedure. Similar to the preliminary study, we followed the general procedure adopted in Bubble [11] and IFC [28] for comparing cursor techniques. The study interface was similar to the one that we used in the preliminary study, but with different sets of testing factors. Participants are instructed to select the start and goal targets alternatively in each trial, which was drawn in yellow and green color respectively. The target would be highlighted by a blue surrounding stroke when the target is being captured. Participants were instructed to first click the start target and then select the goal target as fast as and as accurate as possible. When an incorrect target was selected (after the start target is clicked) in an individual task, the task would be marked as failed and participants needed to re-select the correct goal target (without re-clicking the start target).

We observed that some participants in the preliminary study would perform multiple clicks after an incorrect selection, trying to complete the task with random clicking positions near the correct goal target. This usually causes unnecessary long task completion time that we would like to avoid. Therefore, we instructed the participants not to perform random clicks in the trial, and we observed no such random clicking behavior among all participants.

In this experiment, we included two sets of distractors of the same size as the goal target. Four main distractors were placed around the goal target as introduced in the previous study to control the maximum effective width of the goal target. Another set of distractors were placed along the path from the start target to the goal target. The number of these distractors was determined by the Amplitude. They were distributed in a 20° slice beginning from the start target to the nearest main distractor (Fig. 6b).

Design. Our study was a 5×3×3×3 within-participant design with the following factors: (1) five area-cursor techniques **TECH** for comparison: AVC, AVC+M, Bubble Cursor, IFC, MTE; (2) three target radius **R**: 8, 16, 32 pixels; (3) three amplitudes **A**: 256, 512, 768 pixels; (4) three distractor ratio **DRATIO**: 1.5, 3 and 5 (the ratio of the distance between the goal target and each main distractor to the target width).

For AVC and AVC+M, we adopted the $\boldsymbol{s_B}$ (100 pixels/s) and the $\boldsymbol{s_C}$ (30 pixels/s) we selected in the preliminary study. IFC introduced additional factors such as "RotaAngle" and "DistractorDensities" to study the task performance of different cursor techniques. These factors would mainly affect the target selection if the cursor technique is velocity dependent (i.e. depends on both cursor speed and moving direction). Since our proposed technique is speed dependent only, these factors would not affect the size of target effective areas much, and therefore we abandoned these factors. We also followed the concern proposed in IFC [28], that different pointing methods may have different defined target effective width, to examine directly how the **DRATIO** would affect the performance of each area cursor techniques instead of using a constant density factor.

Each participant performed the study in a single session consisting of 5 groups of area cursor techniques in a counterbalanced order by Latin Square. In each group, the participants needed to perform 384 trials of selection tasks, with each trial to click the start target and then select the goal target with the assigned area cursor technique. The task completion time, and the overall error rate were recorded. In each group, participants performed the tasks under different combinations of **R**×**A**×**DRATIO**. Each combination repeated 16 times with the 16 pairs of opposite start and goal targets appearing in a randomized order. We have dropped the trials with **R** = 32 and **A** = 256 as the targets are too close among each other. Thus, a total of 19,200 selection tasks were conducted in the experiment. In this analysis, **R**, **A**, and **DRATIO** were given in random order. Participants were informed about the type of **TECH** that was going to be used, and were educated about the technique before the trails of the **TECH** group is started. A 5-minute warm-up session was given to allow participants to get familiar with the techniques and tasks. They were given a 10-minute break between each **TECH** group. Each participant took approximately 1.5 hours to complete the whole experiment.

Results and Discussion. *Task Completion Time.* Table 2 shows the results of the four-way ANOVA analysis of variance on task completion time. We only show factors with significant effects and skip those insignificant ones. We can observe that **TECH** ($F_{(4,1080)} = 61.6, \mathrm{p} < 0.0005$), **A** ($F_{(2,1080)} = 74.9,\ \mathrm{p} < 0.0005$), **R** ($F_{(2,1080)} = 147.4, \mathrm{p} < 0.0005$) and **DRATIO** ($F_{(2,1080)} = 17.4,\ \mathrm{p} < 0.0005$) have significant effects on the task-completion time. A significant interaction effect can also be observed in **TECH** × **R** and **R** × **DRATIO**. These results show that different **TECHs** were affected differently by the factors.

Post-hoc test showed that the AVC and AVC+M were significantly faster than Bubble (AVC vs Bubble: $p < 0.0005$; AVC+M vs Bubble: $p < 0.05$) and IFC (AVC vs IFC & AVC+M vs IFC: $p < 0.0005$). The average mean task completion time was 730.2ms for AVC, 752.1ms for AVC+M, 815.3ms for Bubble, 994.9ms for IFC, 769.8ms for MTE as shown in Fig. 7a. In general, AVC gives 5.1% improvement in term of the task-completion time over MTE, which was the best cursor method with implicit display style in Guillon et al's research [14]; while both Manhattan distance-based methods (AVC+M and MTE) provided the similar performance. The significant difference between AVC and Manhattan distance-based methods suggested that our new dynamic Voronoi diagrams can better utilize the local screen region for faster target selection than the original bubble and Manhattan distance-based methods.

There was a significant interaction effect of **R** and **TECH** ($F_{(8,1080)} = 2.810$, $p < 0.005$). Fig. 7b shows the mean task completion time of different techniques **TECH** grouped by **R**. Post-hoc pairwise comparison (Table 3) showed that AVC (with all **R**), and AVC+M (with **R** = 8, 16) are significantly faster than Bubble and IFC, while AVC+M was only significantly faster than IFC with **R** = 32. The results showed that AVC (and even AVC+M) were efficient in acquiring smaller targets. On the other hand, both AVC and AVC+M did not have a significant difference to MTE with all **R**.

Table 2. The significant ANOVA results for the task completion time of **TECH** × **A** × **R** × **DRATIO**. (DF is the degree of freedom.)

Factors	DF	F	p
TECH	4	61.608	< 0.0005
A	2	74.945	< 0.0005
R	2	147.405	< 0.0005
DRATIO	2	17.415	< 0.0005
TECH × R	8	2.810	< 0.005
R × DRATIO	4	3.963	< 0.005

Table 3. Significant differences for mean task completion time among **TECH** grouped by target radius **R**, where a < b means **TECH** b was significantly faster than **TECH** a. (A = AVC, Am = AVC+M, B = Bubble, I = IFC, M = MTE)

R	**Comparison**
8, 16	I < B < A, Am I < M
32	I < B < A I < Am, M

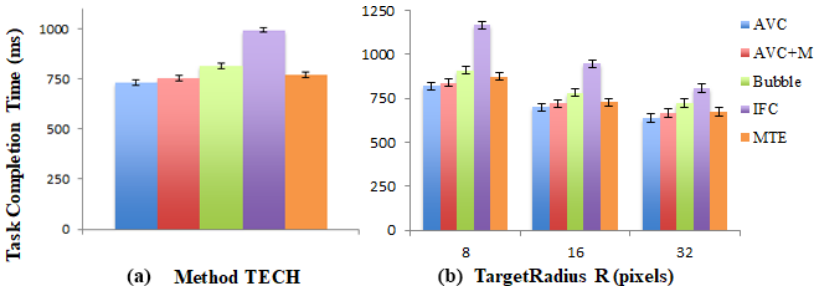


Fig. 7. (a) The mean task completion time of different cursor techniques **TECH (b)** and that grouped by target radius **R**. In this and all later charts, error bars represent standard error.

Error Rate. Four-way ANOVA showed that both **TECH** ($F_{(4,1080)} = 11.4,\ p < 0.0005$), **A** ($F_{(2,1080)} = 27.4,\ p < 0.0005$), **R** ($F_{(2,1080)} = 14.4,\ p < 0.0005$) and **DRATIO** ($F_{(2,1080)} = 13.0,\ p < 0.0005$) placed a significant effect on the error rate, as shown in Table 4. Fig. 8a shows the mean error rate of different TECH. Post-hoc pairwise comparison revealed that AVC yielded significantly less error than the IFC cursor ($p < 0.0005$) and the AVC+M method ($p < 0.05$), and there was no significant difference among the AVC, MTE, and Bubble cursor, in terms of the error rate. In addition, the AVC+M yielded significantly more error than the MTE did in general ($p < 0.005$). IFC had a significantly higher error rate than other techniques, this may due to the orientation-based control style of IFC, which is different from other tested cursor techniques and could have a steeper learning curve and require longer training to master.

TECH and **A** had a significant interaction effect on the error rate ($F_{(8,1080)} = 2.27$, $p < 0.05$). Post-hoc pairwise comparison (Table 5a) showed that the AVC and AVC+M were significantly more accurate than the IFC with **A** = 256 (AVC vs IFC: $p < 0.0005$; AVC+M vs IFC: $p < 0.005$) and **A** = 512 (AVC vs IFC & AVC+M vs IFC: $p < 0.0005$). However, both AVC ($p < 0.05$) and AVC+M ($p < 0.0005$) were significantly less accurate than the MTE, and AVC+M was significantly less accurate than Bubble while **A** = 768 ($p < 0.05$)_. AVC+M and MTE produced straight line region boundaries as shown in Fig. 5, which caused the selection process became unpredictable. While selecting a target along with a long enough amplitude, the immediate change of the target area might confuse the users, and hence the error rate was increased for larger amplitude. Fig. 8b shows the mean error rate of different techniques **TECH** grouped by **A**. All techniques had a higher error rate with **A** = 256, comparative to higher amplitude. This might due to the fact that the participants had shorter overall cursor-moving time when the goal target was closer, and hence they would be easier to perform incorrect selections due to short response time. While Bubble, IFC, and MTE each had similar error rates with **A** = 512 and **A** = 768, AVC and AVC+M captured a relatively higher error rate with **A** = 768 than those when **A** = 512. A possible explanation is that the accuracy of ballistic motion dropped when the amplitude increased. This increased the distances between the cursor and the goal target when entering the correction phase, causing the goal target received a relatively smaller

weight and a smaller effective area, thus increased the overall error rate. As the higher-speed or consecutive ballistic motions were usually used for selecting targets with larger amplitudes, a sophisticated speed-based weighting scheme, other than a fixed weighting function, could be used to define the AWVD, in order to improve the performance and error rates in large amplitude selection tasks.

TECH and **DRATIO** also had a significant interaction effect on the error rate ($F_{(8,1080)} = 2.06$, $p < 0.05$). Post-hoc pairwise comparison (Table 5b) showed that the five methods did not have significant error rate differences among others while the **DRATIO** was small enough (**DRATIO** = 1.5). Fig. 8c had explained that the error rate of both methods was high due to the smaller effective width. AVC and AVC+M were both significantly more accurate than the IFC cursor with **DRATIO** = 3.0 (AVC: $p < 0.005$; AVC+M: $p < 0.05$) and **DRATIO** = 5.0 ($p < 0.0005$). Fig. 8c shows the mean error rate of different **TECH** grouped by **DRATIO**. We observed that **DRATIO** had less effect for IFC on error rate. Except for IFC, the error rate of other **TECH** was gradually decreased as the distractor ratio increases, mainly owing to the effective area become larger.

Lastly, we could also observed that the method of IFC was significantly slower than the other four methods ($p < 0.0005$), and in general IFC had significantly higher error rate than the other four methods except for the cases of **A** = 768 and **DRATIO** = 1.5. IFC had a better performance than other tested area cursor techniques in their paper, this might be owing to IFC was the only technique that used implicit display style in their experiment, while in our paper, all tested area cursor techniques used implicit display style and the effect of different display styles on performance, as reported in [15, 28], could be minimized.

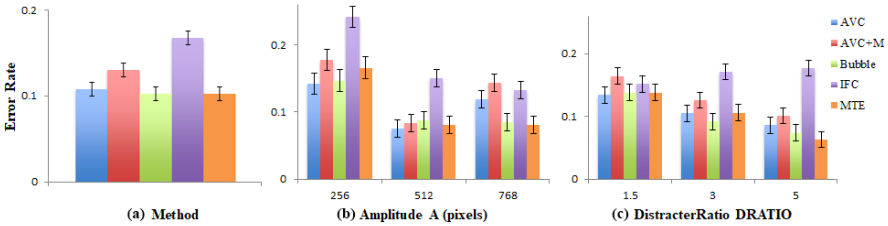


Fig. 8. (a) The mean error rate of different cursor techniques **TECHs**, and that of grouped by **(b)** Amplitude **A**, **(c)** DistractorRatio **DRATIO**

Table 4. The significant ANOVA results for the error rate of **TECH** × **A** × **R** × **DRATIO**. (DF is the degree of freedom.)

Factors	**DF**	**F**	**p**
TECH	4	11.365	< 0.0005
A	2	27.382	< 0.0005
R	2	14.411	< 0.0005
DRATIO	2	13.016	< 0.0005
TECH × A	8	2.272	<0.05
TECH × DRATIO	8	2.056	<0.05

Table 5. Significant differences for mean error rate among **TECHs** grouped by **(a)** amplitude **A**, **(b)** DistractorRatio **DRATIO**, where a < b means **TECH** b had less error rate than **TECH** a. (A = AVC, Am = AVC+M, B = Bubble, I = IFC, M = MTE)

(a)

A	Comparison
256	I < A, Am, B, M
512	I < A, Am, B, M
768	Am, I < B, M A < M

(b)

DRATIO	Comparison
1.5	-
3.0	I < A, Am, B, M
5.0	I < A, Am, B, M Am < M

5 FITTS' LAW INDEX OF DIFFICULTY

Fig. 9 illustrates the index of performance of the five **TECHs** by plotting the relationship between task completion time and index of difficulty (ID). We define the ID for each selection trials using the effective target width (EW) of the goal target.

Since AVC, AVC+M. and IFC define effective areas non-statically, the effective widths of targets are unknown before the selection is performed when using these techniques. Therefore, for a fair comparison, we consider the EW as clicking distance (the distance between goal target and the clicking position) for all techniques. Specifically, we take the average clicking distance for each combination of amplitude, target radius and distractor ratio for all tested cursor techniques, compute the IDs and linearly fit 24 points for each technique as shown in Fig. 9. Table 6 lists the intercept, slope and r^2 values for each technique. We can see that all techniques fit the linear model with reasonable r^2 values, thus the performance of the AVC and AVC+M can be modeled using Fitts' law. AVC had the smallest slope among all **TECHs**. indicating AVC is easier to master than other tested cursor techniques.

Table 6. Linear fit: intercept, slope and $\boldsymbol{r^2}$ values for different TECH. ($t = a + b \times ID$)

TECH	intercept a	slope b	r^2
AVC	130	136	0.764
AVC+M	86.5	148	0.814
Bubble	139	153	0.855
IFC	-339	310	0.938
MTE	76.6	156	0.820

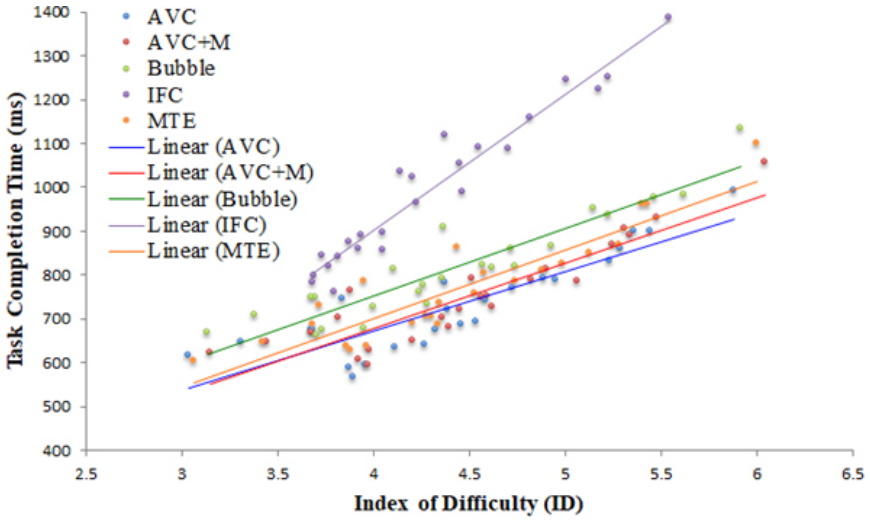


Fig. 9. Linear regression of different TECH.

6 CONCLUSION AND FUTURE WORK

This paper introduced a new target-selection technique that dynamically resizes the effective areas of targets by analyzing two different phases of mouse movement: the ballistic and correction phases during target selection. The key contribution of this paper is two area-cursor techniques: AVC and AVC+M, by defining the Additively Weighted Voronoi Diagram (AWVD) with the selectable targets such that the effective areas of the targets close to the cursor will be enlarged dynamically right after the cursor movement switched from ballistic to correction phase. The correction cursor movement becomes easier so the overall target selection time can be reduced. The result of our controlled experiment showed that our proposed technique achieved better performance over existing area-cursor techniques. In addition, the performance of AVC and AVC+M can be modeled using Fitts' law.

In the future, the relationship between Ballistic and Correction phases of cursor movement can be further evaluated. According to the result of our study, the error rates of AVC and AVC+M increased while the amplitude was larger. Introducing additional detection of consecutive Ballistic movements may resolve this issue. We would also like to investigate a content-aware weighting method for adjusting the weights of on-screen targets according to both the speed of the last ballistic movement and the local target density, to provide a better target selection performance. Besides, studying the performance of AVC in 3D systems would be one of our future work.

References

1. Asano, T., Sharlin, E., Kitamura, Y., Takashima, K., & Kishino, F. (2005, October). Predictive interaction using the delphian desktop. In Proceedings of the 18th annual ACM symposium on User interface software and technology (pp. 133-141). ACM.
2. Baudisch, P., Cutrell, E., Robbins, D., Czerwinski, M., Tandler, P., Bederson, B., & Zierlinger, A. (2003, August). Drag-and-pop and drag-and-pick: Techniques for accessing remote screen content on touch-and pen-operated systems. In Proceedings of INTERACT (Vol. 3, pp. 57-64).
3. Baudisch, P., Zotov, A., Cutrell, E., & Hinckley, K. (2008, May). Starburst: a target expansion algorithm for non-uniform target distributions. In Proceedings of the working conference on Advanced visual interfaces (pp. 129-137). ACM.
4. Bezerianos, A., & Balakrishnan, R. (2005, April). The vacuum: facilitating the manipulation of distant objects. In Proceedings of the SIGCHI conference on Human factors in computing systems(pp. 361-370). ACM.
5. Blanch, R., Guiard, Y., & Beaudouin-Lafon, M. (2004, April). Semantic pointing: improving target acquisition with control-display ratio adaptation. In Proceedings of the SIGCHI conference on Human factors in computing systems (pp. 519-526). ACM.
6. Blanch, R., & Ortega, M. (2009, April). Rake cursor: improving pointing performance with concurrent input channels. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (pp. 1415-1418). ACM.
7. Boots, B. N., Chiu, S. N., Okabe, A., & Sugihara, K. (2000). Spatial tessellations: concepts and applications of Voronoi diagrams, 2nd edition.
8. Chapuis, O., Labrune, J. B., & Pietriga, E. (2009, April). DynaSpot: speed-dependent area cursor. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems(pp. 1391-1400). ACM.
9. Cockburn, A., & Brock, P. (2006, June). Human on-line response to visual and motor target expansion. In Proceedings of Graphics Interface 2006 (pp. 81-87). Canadian Information Processing Society.
10. Cockburn, A., & Firth, A. (2004). Improving the acquisition of small targets. In People and Computers XVII—Designing for Society (pp. 181-196). Springer, London.
11. Grossman, T., & Balakrishnan, R. (2005, April). The bubble cursor: enhancing target acquisition by dynamic resizing of the cursor's activation area. In Proceedings of the SIGCHI conference on Human factors in computing systems (pp. 281-290). ACM.
12. Guiard, Y., Blanch, R., & Beaudouin-Lafon, M. (2004, May). Object pointing: a complement to bitmap pointing in GUIs. In Proceedings of Graphics Interface 2004 (pp. 9-16). Canadian Human-Computer Communications Society.
13. Gutwin, C. (2002, April). Improving focus targeting in interactive fisheye views. In Proceedings of the SIGCHI conference on Human factors in computing systems (pp. 267-274). ACM.
14. Guillon, M., Leitner, F., & Nigay, L. (2015, April). Investigating Visual Feedforward for Target Expansion Techniques. In Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (pp. 2777-2786). ACM.
15. Guillon, M., Leitner, F., & Nigay, L. (2016, June). Target Expansion Lens: It is Not the More Visual Feedback the Better!. In Proceedings of the International Working Conference on Advanced Visual Interfaces (pp. 52-59). ACM.
16. Hertzum, M., & Hornbæk, K. (2007). Input techniques that dynamically change their cursor activation area: A comparison of bubble and cell cursors. International Journal of Human-Computer Studies, 65(10), 833-851.

17. Kabbash, P., & Buxton, W. A. (1995, May). The “prince” technique: Fitts' law and selection using area cursors. In Proceedings of the SIGCHI conference on Human factors in computing systems (pp. 273-279). ACM Press/Addison-Wesley Publishing Co.
18. Kobayashi, M., & Igarashi, T. (2008, April). Ninja cursors: using multiple cursors to assist target acquisition on large screens. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (pp. 949-958). ACM.
19. Lank, E., Cheng, Y. C. N., & Ruiz, J. (2007, April). Endpoint prediction using motion kinematics. In Proceedings of the SIGCHI conference on Human Factors in Computing Systems(pp. 637-646). ACM.
20. Laukkanen, J., Isokoski, P., & Räihä, K. J. (2008, April). The cone and the lazy bubble: two efficient alternatives between the point cursor and the bubble cursor. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems(pp. 309-312). ACM.
21. Li, W. H. A., Fu, H., & Zhu, K. (2016). BezelCursor: bezel-initiated cursor for one-handed target acquisition on mobile touch screens. International Journal of Mobile Human Computer Interaction (IJMHCI), 8(1), 1-22.
22. MacKenzie, I. S. (1992). Fitts' law as a research and design tool in human-computer interaction. Human-computer interaction, 7(1), 91-139.
23. McGuffin, M. J., & Balakrishnan, R. (2005). Fitts' law and expanding targets: Experimental studies and designs for user interfaces. ACM Transactions on Computer-Human Interaction (TOCHI), 12(4), 388-422.
24. Meyer, D. E., Abrams, R. A., Kornblum, S., Wright, C. E., & Keith Smith, J. E. (1988). Optimality in human motor performance: ideal control of rapid aimed movements. Psychological review, 95(3), 340.
25. Moscovich, T., & Hughes, J. F. (2006, June). Multi-finger cursor techniques. In Proceedings of Graphics Interface 2006 (pp. 1-7). Canadian Information Processing Society.
26. Pietriga, E., & Appert, C. (2008, April). Sigma lenses: focus-context transitions combining space, time and translucence. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (pp. 1343-1352). ACM.
27. Soukoreff, R. W., & MacKenzie, I. S. (2004). 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), 751-789.
28. Su, X., Au, O. K. C., & Lau, R. W. (2014, April). The implicit fan cursor: a velocity dependent area cursor. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems(pp. 753-762). ACM.
29. Worden, A., Walker, N., Bharat, K., & Hudson, S. (1997, March). Making computers easier for older adults to use: area cursors and sticky icons. In Proceedings of the ACM SIGCHI Conference on Human factors in computing systems (pp. 266-271). ACM.