

# The Satellite Cursor: Achieving MAGIC Pointing without Gaze Tracking using Multiple Cursors

Chun Yu<sup>1</sup>, Yuanchun Shi<sup>1</sup>, Ravin Balakrishnan<sup>2</sup>,

Xiangliang Meng<sup>1</sup>, Yue Suo<sup>1</sup>, Mingming Fan<sup>1</sup>, Yongqiang Qin<sup>1</sup>

<sup>1</sup>Department of Computer Science  
Tsinghua University, Beijing, China  
{c-yu06, mxl08, fmm08, qyq08}@mails.thu.edu.cn  
{shiyc, suoyue}@mail.thu.edu.cn

<sup>2</sup>Department of Computer Science  
University of Toronto, Toronto, ON, Canada  
ravin@dgp.toronto.edu

## ABSTRACT

We present the satellite cursor – a novel technique that uses multiple cursors to improve pointing performance by reducing input movement. The satellite cursor associates every target with a separate cursor in its vicinity for pointing, which realizes the MAGIC (manual and gaze input cascade) pointing method without gaze tracking. We discuss the problem of visual clutter caused by multiple cursors and propose several designs to mitigate it. Two controlled experiments were conducted to evaluate satellite cursor performance in a simple reciprocal pointing task and a complex task with multiple targets of varying layout densities. Results show the satellite cursor can save significant mouse movement and consequently pointing time, especially for sparse target layouts, and that satellite cursor performance can be accurately modeled by Fitts' Law.

## Author Keywords

MAGIC pointing, multiple cursor, reducing A.

## ACM Classification Keywords

H.5.2 [User Interface]: Theory and methods, Interaction styles, Graphical user interface (GUI).

## General Terms

Human Factors, Performance, Theory.

## INTRODUCTION

Pointing is a fundamental task in GUI interfaces. However, increasing display sizes result in larger cursor movements, increasing fatigue and pointing times. Based on Fitts Law [7], plenty of researchers [2-4,8,10,13,18] have sought to improve pointing performance by manipulating cursor movement amplitude ( $A$ ), target width ( $W$ ) or both, achieving a more effective layout of targets in motor space than in visual space, with larger width and shorter movement distance for every target.

Although most previous techniques do reduce pointing time, they are subject to some distraction factors [16]: as the cur-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

UIST'10, October 3–6, 2010, New York, New York, USA.

Copyright 2010 ACM 978-1-4503-0271-5/10/10...\$10.00.

sor moves toward the intended target, it is inevitably perturbed by the bypassed targets either in the visual or motor domain, which affects users' planning of mouse movement and can negate the time saving gained from reorganizing target layouts. The distraction factor could be more severe for the cursor jumping techniques [1,8] based on reducing  $A$ , since the inaccurate prediction of cursor movement may defy users' expectation. We believe that the distraction factor is essentially related to the fact that  $N$  targets compete for the "favor" of a single cursor.

To achieve both an effective layout of targets in motor space and minimum distraction, we propose the "satellite cursor". Instead of employing a fixed number of cursors for all targets on screen as previous multi-cursor techniques [5,11,15] do, the satellite cursor associates every target with a separate cursor in its vicinity, which acts as the target's "satellite". All satellite cursors move synchronously with the mouse. Users can only select a target by its own satellite cursor. To avoid ambiguity, each satellite cursor's position is carefully calculated so that at most one target can be visually indicated by its satellite cursor. Pointing with a satellite cursor is done as if with a standard cursor locally.

Essentially, this approach uses multiple cursors to re-layout targets in motor space and improves pointing by reducing pointing distances. An interesting aspect of this technique is that it realizes the MAGIC (manual and gaze input cascade) [19] pointing method without requiring an additional gaze input channel as was previously done.

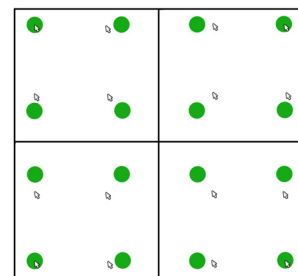


Figure 1: In this example, the 4 satellite cursors move synchronously and only one can be selected by its satellite cursor at one time.

We next review past research on improving pointing performance; discuss the design of satellite cursor, detailed in a two-phase algorithm, “aggregate and expand”, followed by several designs to mitigate visual clutter caused by multiple cursors; evaluate the performance of the satellite cursor in two studies; and conclude by discussing design implications for real applications and future work.

## RELATED WORK

Most previous research on improving pointing performance are target-aware approaches that attempt to beat Fitts’ Law by manipulating target width ( $W$ ) and cursor movement amplitude ( $A$ ). By awareness of target numbers, positions and sizes, these methods re-layout targets in motor space to reduce pointing time. Although target-agnostic approaches require no knowledge about on-screen targets, the degree to which they improve pointing is also limited as they are “blind to the very targets they seek to acquire” [16]. Other directions include gaze engaged approaches [19] and more recent multi-cursor techniques [5,11,15].

### Target-Aware Approaches

In this section, we review previous target-aware methods from two aspects: the re-layout strategy and the distraction factors. However, it is worth first asking “why targets in motor space can be re-laid out?” and “to what degree can pointing time be saved?” The answer lies in empty space [9], on-screen space that cannot be selected or is unclickable. Target-aware approaches vary in their particular strategy of making use of empty space, by either assigning it as the effective width of a target or eliminating it from the cursor movement distance. For a layout of targets with no empty space, such as a painting canvas, such target-aware techniques produce no benefit. So the re-layout strategy can be systematically investigated by questioning how empty space is taken advantage of and how much empty space is used.

**Area Cursor** [10]. By replacing the single pixel hotspot of a cursor with a region hotspot, an area cursor enlarges the effective width of targets. In this case, empty space is used to increase target width. However, it is not sufficiently leveraged because the enlarged targets may not necessarily occupy the whole motor space. Also, ambiguity arises when multiple targets fall inside the area cursor at one time.

**Bubble Cursor** [8]. As an improved version of area cursors, the bubble cursor defines the activation region of a target as the Voronoi cell of a target, which ensures both maximized use of empty space and removes ambiguity. However, due to Voronoi decomposition, the assignment of empty space to each target depends on the layout of targets and may be unequal. For example, the effective width of targets in dense regions is enlarged to a lesser degree than those in sparse regions. Besides that, the constantly changing size of the bubble cursor can be distracting and may potentially affect pointing performance.

**Drag-and-Pop** [2]. As the user starts dragging an icon, this technique moves potential target icons toward the user’s current cursor location and pointing is then done as if with

a standard cursor with no distraction effect. In motor space, all empty space between the targets and the cursor is eliminated to reduce pointing distance. However, this technique requires input activation (i.e., initiation of the drag) prior to actual pointing, which arguably makes it less suitable for non-dragging pointing tasks.

**Object Pointing** [8]. Object pointing uses a cursor that skips empty space. When the cursor is about to leave a target, it jumps to the first object located in that direction. However, predicting the intended target by a moving vector is subject to kinematic noise and can easily defy users’ expectations.

**Semantic Pointing** [4]. By increasing control-display (C-D) ratio within targets and decreasing C-D ratio in empty space, semantic pointing makes targets bigger and shrinks empty space in the motor domain. It only partially utilizes the empty space. As it preserves continuity of cursor movement, the distraction effect may not be as serious as in object pointing. But the variable C-D ratio across the screen may also hamper users’ planning of cursor movement.

Among those techniques, the area cursor and the bubble cursor are “increasing  $W$ ” approaches while drag-and-pop and object pointing are “decreasing  $A$ ”. An advantage of the “decreasing  $A$ ” approaches is that they can mitigate physical fatigue with less mouse movement. This is of practical value for daily computer users especially as displays get larger. Table 1 summarizes the usage of empty space and distraction types for each technique.

Technique	Empty Space Usage	Empty Space Usage Ratio	Distraction
Area Cursor	Width	Partial	Visual
Bubble Cursor	Width	Complete	Visual
Drag-and-pop	Distance	Complete	None
Object Pointing	Distance	Complete	Motor
Semantic Pointing	Distance	Partial	Motor

Table 1: Summary of previous technique

We believe the distraction effects presented in these works are essentially related to the fact that  $N$  targets compete for the “favor” of a single cursor. This is a key motivation for resorting to multiple cursors to improve pointing.

### Gaze Engaged Approaches

Gaze has long been used as a salient pointing mechanism. But “it is unnatural to overload a perceptual channel such as vision with a motor control task”. For this reason, Zhai [19] propose MAGIC pointing, by which a large portion of the cursor movement was eliminated by warping the cursor to the eye gaze area and subsequent fine manipulation and selection was done with a mouse.

### Multi-Cursor Techniques

A recent approach is to improve pointing by multiple cursors. In the Ninja Cursor [11], an array of cursors evenly distributed over the screen synchronously follow the movement of the mouse. Users point to a target with one of the cursors located nearest to the target. To resolve ambiguity when multiple cursors indicate different targets, Ninja

Cursor uses a queue-based strategy, in which to activate some cursor, all cursors that preceded it in the queue must be pushed out by moving the cursor away from their potential targets. This can be awkward and arguably too complex to be practical. Two later works [5,15] use gaze for indicating the active cursor, at the expense of requiring an additional input modality.

### THE SATELLITE CURSOR

The satellite cursor extends the multi-cursor technique to resolve the distracting effects resulting from multiple targets' competing for the favor of a single cursor. Instead of employing a fixed number of cursors for all targets on screen as previous multi-cursor techniques do, the satellite cursor technique associates every target with a separate satellite cursor in its vicinity. A target can only be selected by its satellite cursor. All satellite cursors move synchronously with a same constant C-D ratio. The trick here is that at most one target can be visually indicated by its satellite cursor at any moment. Pointing with a satellite cursor is done as if with a standard cursor locally with no concern that any other target might be below its satellite cursor. While ensuring no ambiguity when selecting, a satellite cursor should also be positioned as near as possible to its target to save cursor movement distance. These two requirements are critical when positioning satellite cursors.

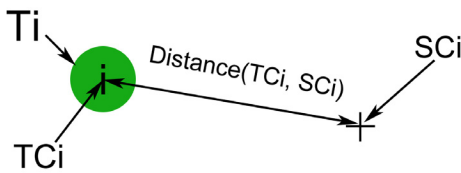


Figure 2: Definition of symbols

To aid discussion, we first define a set of symbols (Figure 2) and operations that help clarify the problem.

$T_i$ : the  $i^{\text{th}}$  target on the screen.

$SC_i(x, y)$ : position of  $T_i$ 's satellite cursor on screen.

$TC_i(x, y)$ : position of  $T_i$ 's center on screen.

$Distance(TC_i, SC_i)$ : Euclidean distance between  $TC_i$  and  $SC_i$ .

$Within(T_i, SC_i)$ : 1 if  $SC_i$  is within  $T_i$ ; 0 if not.

Calculating positions of all satellite cursors with no ambiguity and minimal movement distance can be described as an optimization problem as follows:

$$\min_{SC=\{SC_i|i=1\dots n\}} \sum_{i=1}^n Distance(T_i, SC_i)$$

$$\text{s. t. } \sum_{i=1}^n Within(T_i, SC_i) \leq 1$$

Although this problem can be resolved computationally, the result might be unpredictable to the user. For this reason, we propose a more intuitive solution - a two-step algorithm named "Aggregate and Expand". We introduce it using a 4-target layout example.

### Aggregate and Expand Algorithm

In the first step, all targets are "aggregated" to the main cursor to be more compact but with no overlap between each other. The resulting target at the new location is noted as an image of the original target, or an image target. Then we define the *pointing vector* for each image target as the Euclidian vector that takes the main cursor as the initial point and its center as the terminal point. As shown in Figure 3, the pointing vectors are  $v_1, v_2, v_3, v_4$  for each, suffixed with the target number.

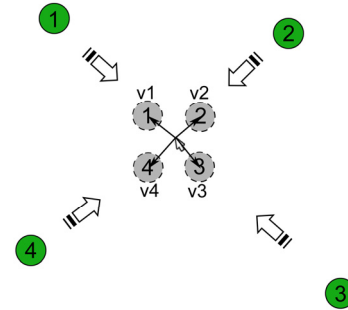


Figure 3: Image targets represented with gray circles; the pointing vector  $v_1$ - $v_4$  are represented with black arrows.

In the expand step, we calculate positions of all satellite cursors. Before that, we define the *pointing vector* for each target as the Euclidian vector that takes its satellite cursor as the initial point and its center as the terminal point. Finally, the satellite cursor of a target is positioned so that the target's *pointing vector* is equivalent to that of its image's *pointing vector*. Effectively, all image targets "expand" to the same position as their targets, trailed by its *pointing vector*. The satellite cursor is positioned at the terminal point of the new settled *pointing vector* (Figure 4).

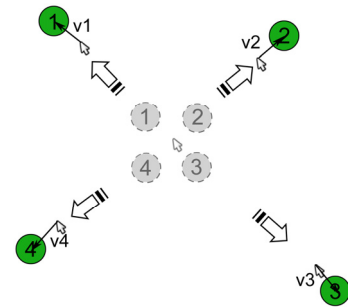


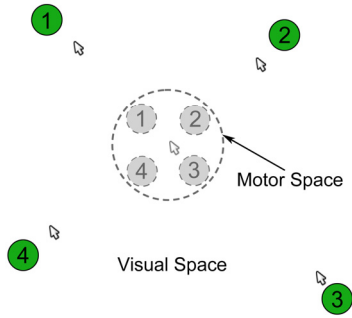
Figure 4: Satellite cursors are represented as black arrow at the terminal point of the pointing vector.

In this example, we use only circle targets. However, the satellite cursor supports targets of any shape, which is very important for applying it to real GUI tasks where targets can be of arbitrary shape. The main cursor is chosen to be the satellite cursor with which a selection has just been conducted for serial selections, representing users' memory of cursor position as single cursor users might expect.

### Avoid Ambiguity

As users control the mouse, all satellite cursors move synchronously in an identical CD ratio with the main cursor (Figure 5). Hence, at any moment, a target's *pointing vec-*

tor is equivalent to that of its image’s, which implies the equivalence between a satellite cursor selecting its target and the main cursor selecting its image. That is, if the main cursor selects an image target (within its region), then its original target must have been selected by its satellite cursor. Since there is no overlap between each image target, the main cursor can select at most one target at one time. It thus follows that only one satellite cursor can select its target a given time.



**Figure 5: Target layout in visual space (green) and motor space (white); selecting a target by its satellite cursor in visual space is equivalent to selecting its image target by the main cursor in motor space.**

### Empty Space Usage

The image targets and the main cursor visualize this pointing task in motor space (Figure 5). By aggregating targets to be as compact as possible, we can eliminate all the empty space to reduce movement distance, which is similar to the re-layout strategy of the drag-and-pop technique. For layouts of evenly distributed targets, the decreasing ratio of pointing distance can be estimated as  $\frac{S_{\text{targets}}}{S_{\text{screen}}}$ , where  $S_{\text{targets}}$  is the sum of areas of all targets and  $S_{\text{screen}}$  is the area of the screen. Thus, the sparser the layout of targets, the more effective the satellite cursor is at saving mouse movement.

### MAGIC Pointing Method

The MAGIC pointing method is potentially the fastest pointing technique because human eyes move much quicker than hands and target acquisition usually requires the user to look at the target first, before actuating cursor control [19]. With the satellite cursor, users first look at the target, then locate its satellite cursor in its vicinity, and finally point with the satellite cursor as if with a standard cursor locally. Much cursor movement is eliminated.

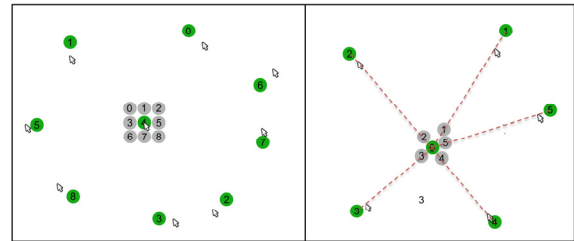
A difference between the satellite cursor and gaze-based MAGIC pointing is how the position of the vicinal cursor is determined. For gaze-based MAGIC pointing, the position is controlled by the accuracy of gaze tracking, while for the satellite cursor, the position is determined by the layout strategy in the aggregating step. A particularly interesting aspect of the satellite cursor is that it approximates MAGIC pointing without the need for gaze tracking.

### Re-Layout Strategy

The only constraint for the satellite cursor to work is that when re-laying out or “aggregating” targets in motor space,

there should be no overlap between each target. We propose two re-layout strategies in this section.

The first is called “tiled layout”, which eliminates empty space as much as possible. Figure 6(left) shows a square-like layout of image targets in motor space, with the main cursor at the center. The algorithm is simple and produces a resolution approximating the optimal one in terms of decreasing pointing distance. However, a problem with this layout strategy is that in order to take maximum advantage of empty space, the position of the image target is uncontrolled, which means the satellite cursor is positioned at an unpredictable direction from its target. This directional uncertainty will increase the time users spend on finding the satellite cursor.



**Figure 6: (left) Tiled. (right) Radial layout of image targets in motor space for difference numbers of targets.**

The second is called “radial layout”, which arranges the target, its image target and the main cursor in a line (Figure 6, right), so that a satellite cursor always stands on the segment connecting the main cursor and the target. This design plays a similar but better role than the “intelligent offset” of Zhai’s work to minimize directional uncertainty since the warped cursor’s position is partly dependent on gaze tracking in his work. The radial layout leverages users’ memory of cursor position and makes it possible for users to perform motors acts in parallel to visual search and to quickly locate the satellite cursor. The algorithm runs like this: 1. Calculate and sort the distances between the main cursor and every target in visual space. 2. Aggregate targets toward the main cursor one by one in ascending order of the distance as tiled as possible. The resulted layout tends to position targets visually more distant farther from the main cursor in motor space, which may improve users’ anticipation of the satellite cursor’s position.

The satellite cursor technique uses multiple cursors to bridge motor space and visual space, which makes re-layout strategies of targets in motor space as flexible as possible. This feature enables easily incorporating usage information into the re-layout strategy, for example, by positioning a frequently used target nearer to the main cursor while leaving a seldom used one farther.

### VISUAL CLUTTER AND DISTRACTION

The basic satellite cursor technique works well for sparse target layouts where a satellite cursor is always in the vicinity of its target and it is the only one in the vicinity. However, as target layouts get denser, the chances increase that more than one satellite cursor show up in a target’s vicinity, which requires users to determine which one to use for pointing. Although this distraction may make satellite cursor difficult to

use (Figure 7), it is different from the distraction effects of techniques based on a single cursor which can hardly be resolved. On the other hand, multiple cursors bring in visual clutter and may discomfort users.



**Figure 7: Visual clutter and distraction for dense clusters of targets.**

As current graphical user interfaces are usually filled with both sparse and dense target layouts, lack of support for dense target layouts limits the satellite cursor’s usage for real tasks. For this reason, we propose three techniques to mitigate the visual clutter and distraction.

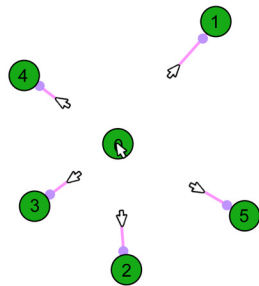
### Group Dense Clusters of Targets

A solution to address the distraction resulting from dense target layouts is to treat a dense cluster of targets as a single one and associate the cluster with a single satellite cursor. Thus, when aggregating and expanding, the relative distance between each target within the cluster is preserved. Also, pointing within the cluster is done as if pointing with a standard cursor. Although the empty space between targets is not sufficiently utilized in this approach, the sum of empty space in the cluster is limited for the high density of target layout. Such dense layout of targets may be found in palettes or ribbons on current GUI interfaces.

### Visual Decoration

Another direction for mitigating the distraction is to add visual decoration to indicate the correspondence between a target and its satellite cursor. A similar design can also be found in “drag-and-pop” where “a rubber band” is used to connect a tip icon and its original icon [2]. We propose three visualization elements (Figure 8).

- *Oriented Cursor.* The satellite cursor’s arrow points where its target resides.
- *Ear.* A circle sits on the rim of a target, indicating where its satellite cursor resides.
- *Line.* A line connects between the target center and its satellite cursor.



**Figure 8: Each target is pointed to by its satellite cursor, with a semi-transparent blue ear indicating the direction. Each target and its satellite are visually connected by a semi-transparent pink line.**

In comparison to *oriented cursor* and *ear*, *line* is more explicit but also brings in more visual clutter.

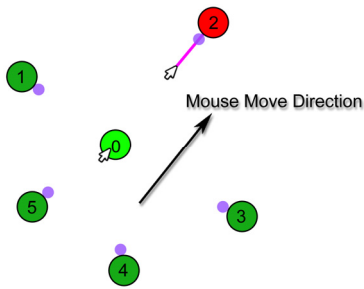
### The Hidden Satellite Cursor

The idea is to make visible a satellite cursor only when necessary and make invisible those the user does not intend to point with. Thus, most satellite cursors and their associated visual decorations are hidden and thus visual clutter and distraction can be largely reduced. The potential distraction of cursor change caused by UI updates, e.g. when a dialog pops up, is also eliminated since cursors are made visible only when mouse is moving. Further, as MAGIC indicates, the user first looks at the target then points. Thus user attention is focused on the intended target and its satellite cursor, which is not easily distracted by other cursors. Hence the problem is to determine the intended targets.

Several researchers [1,9,12] have explored predicting the intended target based on mouse movement parameters: movement direction, speed and peak velocity. However, due to the kinematic noise and randomness of user behavior, none achieves a high enough prediction accuracy to work alone in practice. However, for the satellite cursor, there is an important difference regarding target prediction. Previous pointing techniques work based on a single cursor and the aim of target prediction is to determine the most likely intended target. For the satellite cursor, target prediction is not for target pointing per se, but rather for rendering a reasonable subset of satellite cursors thus making even imperfect prediction quite useful.

*Target Prediction Algorithm.* Our target prediction algorithm only makes use of the cursor movement direction. Any target lying within an angular threshold of its satellite cursor’s movement direction is identified as an intended target. Since the directional bias tends to increase as a satellite cursor approaches its target, the angular threshold value is tuned according to the distance between the satellite cursor and its target. The movement direction is calculated upon most recent trajectory longer than 6 pixels to filter motor noises. The displacement/distance ratio of that trajectory should be above 0.9, otherwise it is determined as undirectional movement. All algorithm parameters are determined to make visible the satellite cursors of all intended targets as responsively as possible.

*Visualization.* The main cursor is always kept visible to indicate the current cursor position. When the mouse is moving, the target prediction algorithm calculates all possible intended targets and makes their satellite cursors and connection lines visible. The length of the connection lines is limited because displaying the connection line and the satellite cursor when it is far from its target is not necessary but contaminates the screen. To indicate the direction of the satellite cursor to its target, the ear decoration is always visible. When a mouse pause is detected, all satellite cursors turn invisible to mitigate visual clutter and distractions. Figure 9 illustrates.



**Figure 9:** As cursor moves from target 0 to target 1, only target 1's satellite cursor and connection line are visible. All other targets' only have their ear visible.

**Pointing method.** The hidden satellite cursor technique presents a different pointing method from that of the basic satellite cursor technique because the satellite cursor is not visible until the user actuates the mouse to point to its target. Interestingly, such a pointing method is just like the conservative MAGIC [19] pointing technique, which delays warping the cursor until the input device is actuated to prevent MAGIC pointing behaving over-zealously. Although this pointing method requires more hand-eye coordination effort, we believe users will get accustomed to it with practice and it is potentially the most promising solution for applying satellite cursor to real applications due to its gentler visual clutter and fewer distractions.

## EXPERIMENT 1

Previous work on improve pointing have demonstrated that the pointing performance can be modeled by Fitts' index of difficulty in motor space [4,8]. While the *amplitude* ( $A$ ) is defined as the distance between the cursor and the target in visual space, the *effective amplitude* ( $EA$ ) can be defined as the corresponding distance between the cursor and the target in motor space. The satellite cursor reduces pointing time by re-laying out the targets in motor space to achieve smaller effective amplitude. One hypothesis is that the performance of the satellite cursor can also be modeled by Fitts' Law using  $EA$ . The first goal of this experiment is to verify this hypothesis.

On the other hand, there is a significant difference between the satellite cursor and those proven to be modeled by Fitts' Law in terms of the pointing method. According to research from psychology and kinesiology, a pointing action consists of two processes: the *plan phase* and the *adjustment phase* [17]. In the *plan phase*, an early movement is planned according to the distance to a target [1]. However, the satellite cursor technique delays calculating the distance until the satellite cursor is located by the user in the vicinity of the intended target. Such a problem also exists in Zhai's MAGIC pointing [19] and the ninja cursor [11], which we refer as *cursor uncertainty*. Further, the *cursor uncertainty* can be divided into *directional uncertainty* and *distance uncertainty*. For the satellite cursor, the tiled layout strategy brings both *directional uncertainty* and *distance uncertainty* while the radial layout strategy brings in only *distance uncertainty*. For the hidden satellite cursor, the satellite cursor is not visible until the user actuates the mouse,

which may further increase the time for the user to locate it. So the second goal of this experiment is to investigate how *cursor uncertainty* and the hidden design affect the overall performance of the satellite cursor. The satellite cursor techniques used in this experiment are: the tiled satellite cursor (Tiled Satellite) with tiled layout & oriented cursor, the radial satellite cursor (Radial Satellite) with radial layout & oriented cursor, and the hidden satellite cursor (Hidden Satellite) with radial layout & oriented cursor & ear & line.

## Apparatus and Participants

The experiment was conducted on a 3.2GHz PC running Windows XP, and a 24" LCD display at 1920x1200 resolution. A standard optical mouse was used for pointing with the speed and acceleration rate set to the Windows XP default values (middle speed, no acceleration). Participants sat 50cm in front of the LCD display.

Ten volunteers participated in the experiment. Participants ranged in ages from 22 to 27, were all right-handed, familiar with traditional WIMP interfaces.

## Procedure

The experimental task required participants to select two equally sized targets back and forth in succession. The targets were rendered as solid circles, equidistant from the center of the display along the horizontal axis. The goal target was colored yellow, and the other green. Participants had to successfully click on the goal target to move on with the task.

For the satellite cursor,  $EA$  is determined by the layout of targets in visual space and the re-layout strategy in the aggregating step. So if there were two targets in the scene, both the tiled and the radial strategy will get an  $EA$  equal to the target width. To control  $EA$  in the experiment, we slightly modified the proposed two re-layout strategies by deliberately positioning the goal target  $EA$  distance from the main cursor in motor space when aggregating. For the radial strategy, the direction from the main cursor to the goal target was consistent with that from the non-goal target to the goal target in visual space. For the tiled strategy, that direction was randomly generated to reflect directional uncertainty. The only difference between the modified strategy and the original one for this task was the presence of distracter targets. Since the goal of this experiment is to investigate how *cursor uncertainty* affects the overall performance of the MAGIC pointing method of the satellite cursor technique, we feel the modified strategy would help simplify and clarify the problem. To our knowledge, this is the first quantitative study on this aspect.

## Design

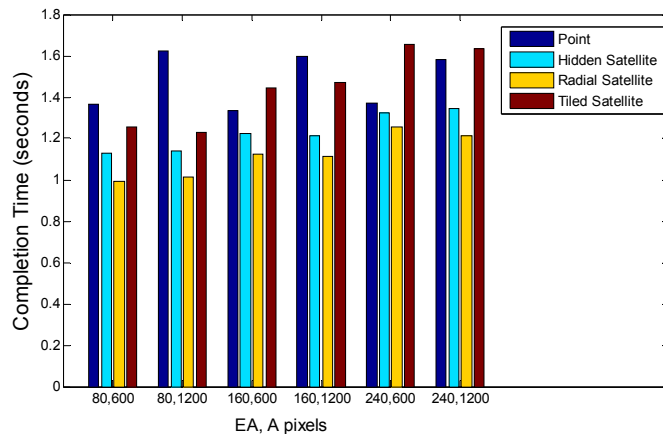
A repeated measures within-subject design was used. The independent variables were cursor type  $CT$  (Point, Tiled Satellite, Radial Satellite and Hidden Satellite), amplitude  $A$  (600 and 1200 pixels), effective amplitude  $EA$  (80, 160 and 240 pixels), and width  $W$  (16, 32 and 48 pixels). A fully crossed design resulted in 72 combinations of  $CT$ ,  $A$ ,  $EA$  and  $W$ .

Each participant performed the experiment in one session lasting about 50 minutes. The session was broken up by cursor type, with 3 blocks of trials completed for each cursor. In each block, participants completed trial sets for each of the 18 combinations of  $A$ ,  $EA$  and  $W$ , presented in random order. A trial set consisted of 4 clicks.

Before using each cursor, participants were given a single warm-up block to familiarize themselves with the cursor and task. The cursor order was counterbalanced across participants.

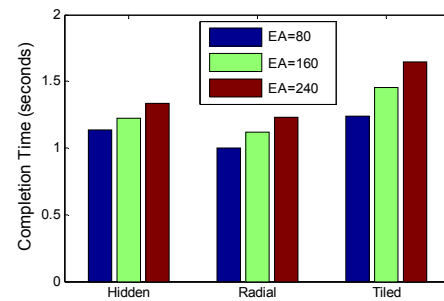
## Results

Completion time is the main dependent measure, and is defined as the time span from the goal target turning yellow to the participant successfully selecting it. For the satellite cursors, it included the time the participant spent on locating the goal target and its satellite cursor, which revealed the *cursor uncertainty* effect. Repeated measures analysis of variance showed significant main effects for  $CT$  ( $F_{3,853}=196.28, p<.00001$ ),  $A$  ( $F_{1,853}=21, p<.0001$ ),  $EA$  ( $F_{2,853}=72.21, p<.00001$ ) and  $W$  ( $F_{2,853}=293.05, p<.00001$ ) on completion time. The overall mean completion times were 1.498 s for Point, 1.229 s for Hidden Satellite, 1.119 s for Radial Satellite and 1.448 s for Tiled Satellite. There were also interaction effects  $CT \times A$  ( $F_{3,814}=29.22, p<.0001$ ) and  $CT \times EA$  ( $F_{6,814}=26.13, p<.0001$ ) for completion time. As Figure 10 illustrates, performance for the satellite cursors is dependent on  $EA$  whereas performance of the point cursor depends on  $A$ .



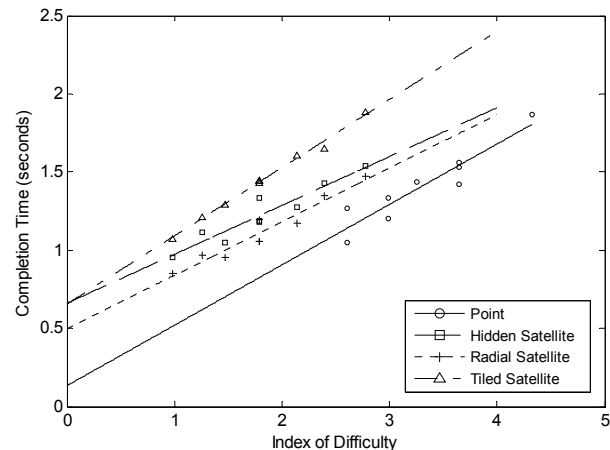
**Figure 10: Completion Time by  $EA$ ,  $A$  values for all cursors, averaged over all  $W$  values**

A post hoc analysis showed significant differences in performance between the Radial Satellite, the Hidden Satellite and the Tiled Satellite for all combinations of  $A$ ,  $EA$  and  $W$ , which meant the directional uncertainty and the hidden cursor both increased the time the user spent on locating the satellite cursor and directional uncertainty was more serious. Interestingly, for the three satellite cursors, there was an interaction effect  $CT \times EA$  ( $F_{4,606}=10.55, p<.0001$ ) on completion time, which indicated increased  $EA$  cost more time for locating the satellite cursor with directional uncertainty (Figure 11).



**Figure 11: Completion Time for Hidden, Tiled, and Radial Satellite cursors, illustrating  $CT \times EA$  interaction.**

Figure 12 plots the completion time as a function of the index of difficulty ( $ID$ ). For Point, we define  $ID$  as  $\log_2(A/W + 1)$ , while for the satellite cursor,  $\log_2(EA/W + 1)$ . Linear regression analysis showed that all four conditions fit the Fitts' Law with  $r^2$  value 0.9373 for Point, 0.9473 for Hidden Satellite, 0.9654 for Radial Satellite and 0.9960 for Tiled Satellite. So the performance of the satellite cursor can be well modeled by Fitts' Law using  $EA$ . This result is consistent with the fact that after locating the satellite cursor, pointing is done as if with a standard cursor and the  $a$  value (time when  $ID=0$ ) reflects the *uncertain cursor* effects on locating the satellite cursor.



**Figure 12: Completion time by  $ID$  for all cursors.**

It is interesting to note that while both the fit lines of Hidden Satellite and Tiled Satellite stays relatively parallel with that of Point, the slope of the fit line for Hidden Cursor is flatter. It approximates Tiled Satellite when  $ID$  is low and approximates Radial Satellite when  $ID$  is high. A possible explanation for this is that when a hidden satellite cursor is near to its target, the participant had to move the mouse more carefully to make the satellite cursor visible. And when the hidden satellite cursor is far from its target, steering the mouse toward the target without much amplitude constraint is easier.

## Feedback and Observation

Of all the 10 participants, 9 ranked Radial Satellite as their favorite cursor and 1 liked Hidden Satellite most. The reason is obvious: Radial Satellite is much easier to learn than Hidden Satellite, which required more hand-eye coordina-

tion effort. But after several trials with Hidden Satellite, all participants could figure out how it works and get used to it. Most participants complained that the unpredictable cursor direction of Tiled Satellite broke up their pointing process. Consistent with Zhai [19], users tend to feel much faster with the satellite cursors than Point even if it did not save much time. They also appreciated the satellite cursor’s advantage of saving motor effort. Some suggested that for Hidden Satellite, the satellite cursor should not be hidden when it overshoot its target by a small distance otherwise they had to locate it again by carefully moving the mouse. We incorporated this into the algorithm.

## EXPERIMENT 2

In experiment 1, we used a simple reciprocal pointing task to show that the satellite cursor can reduce pointing time with the controlled  $EA$  and its performance is accurately modeled by Fitts’ Law using  $EA$ . However, in real 2D GUI interfaces,  $EA$  is determined by the target layouts and the nearby targets may act as distracters for the user to locate the satellite cursor. So in the second experiment, we explore the satellite cursor’s performance in a more realistic environment with multiple targets of varying layout densities.

Given that directional uncertainty largely increased the time the user spent on locating a satellite cursor, we have two possible settings for the satellite cursor: Radial Satellite with lines and Hidden Satellite. While the former takes advantage of the explicit indication of the correspondence between a target and its satellite cursor, it also causes more visual clutter which may irritate users. For this reason, we choose Hidden Satellite in this experiment. In addition to comparing Hidden Satellite to a standard cursor, we include a third technique – the MAGIC pointing [19]. Instead of employing a gaze tracking equipment, we managed to emulate it by automatically warping the cursor to the vicinity of the new goal target (30 pixels to target center at random direction) after the participant successfully selects a target with no latency. So the participant looks at the new goal target, locate the vicinal cursor and then point with it as MAGIC pointing indicates. The simulated MAGIC is thus arguably faster than a real gaze-based MAGIC, such as Zhai’s or Rake Cursor. It hence serves as an ideal performance reference in this experiment.

### Apparatus and Participants

The apparatus used was the same as in experiment 1.

Another ten volunteers participated in the experiment. Participants ranged in ages from 21 to 25, were all right-handed, familiar with the traditional WIMP interfaces.

### Procedure

The experimental task required participants to select randomly generated goal targets in sequence from target layouts with randomly distributed targets (Figure 13). The targets were rendered as solid circles. The goal target was colored yellow, and the other light blue. Participants had to successfully select the goal target before a new goal appeared.

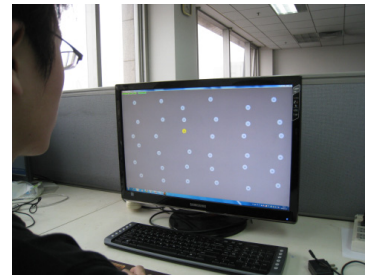


Figure 13: The experiment scenario with the layout density of 6x6 and target width 48 pixels

### Design

A repeated measures within-subject design was used. The independent variables were cursor type  $CT$  (Point, MAGIC and Hidden Satellite), layout density  $N$  (2x2, 4x4 and 6x6), width  $W$  (16,32,48). A fully crossed design resulted in 27 combinations of  $CT$ ,  $N$  and  $W$ . Each participant performed the experiment in one session lasting approximately 30 minutes. The session was broken up by cursor type, with 3 blocks of trials completed for each cursor. In each block participants would complete trial sets for each of the 9 combinations of  $N$  and  $W$  presented in random order. For each trial, the target layout was randomly and evenly generated with no overlap between each target according to  $N$  and participants were required to select 9 targets in sequence. Before using each cursor, participants were given a single warm-up block to familiar themselves with the cursor and task. The cursor order was counterbalanced across participants.

### Result

Completion time is the main dependent measure. Repeated measures analysis of variance showed significant main effects for  $CT$  ( $F_{2,48}=1534.19$ ,  $p<.0001$ ),  $N$  ( $F_{2,48}=9.82$ ,  $p<.001$ ),  $W$  ( $F_{2,48}=418.93$ ,  $p<.0001$ ) on completion time. The overall mean completion times were 1.888 s for Point, 1.528 s for Hidden Satellite, 1.099 s for MAGIC. There were also interaction effects  $CT \times N$  ( $F_{4,48}=120.94$ ,  $p<.0001$ ) and  $CT \times W$  ( $F_{4,48}=3.57$ ,  $p<0.013$ ) for completion time, which indicates  $N$  and  $W$  affect the three technique in different way. As Figure 14 illustrates, the performance of the MAGIC pointing is mainly dependent on  $W$  because almost all the cursor movement is saved by the warped cursor.

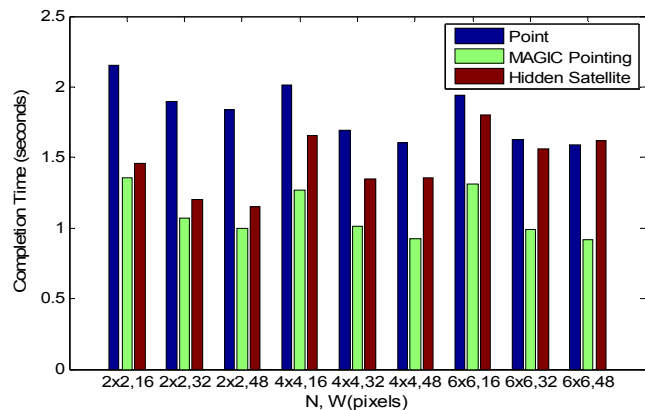


Figure 14: Completion Time by  $N$ ,  $W$  values for all cursors



For Hidden Satellite, it is interesting to observe that there was an interaction effect of  $N \times W$  ( $F_{4,8}=65.17$ ,  $p<.0001$ ) indicating that its pointing performance is affected by  $W$  in different ways under different  $N$ . A possible explanation relates to the Hidden Satellite's design that requires the user to move the mouse to make the satellite cursor visible. Given  $EA$  is determined by  $N$  and  $W$ , for very sparse layouts ( $N=2 \times 2$ ), decreasing  $W$  makes  $EA$  small so that users have to carefully move the mouse to make visible and locate the satellite cursor, while for relatively denser ( $N=6 \times 6$ ) layouts, increasing  $W$  makes  $EA$  very large which also causes difficulty for locating the satellite cursor, especially when it is larger than the length threshold of the connection line, which is 250 pixels in this experiment. As shown in Figure 14, although the performance of Hidden Satellite approximates MAGIC for sparse layouts, it also degenerates as the layout density increases and approximates Point for relatively denser layouts ( $N=6 \times 6$ ,  $W=48$ ). This is because for dense layouts, the  $A/EA$  is not large enough to compensate for the time spent on locating the satellite cursor for Hidden Satellite.

Another dependent measure is mouse movement amount. There was main effects for  $CT$  ( $F_{2,8}=5912.44$ ,  $p<.0001$ ),  $N$  ( $F_{2,48}=202.87$ ,  $p<.0001$ ) and  $W$  ( $F_{2,48}=5.45$ ,  $p<.04$ ) on mouse movement amount and an interaction effect of  $CT \times N$ , which indicated Hidden Satellite and MAGIC's different method of saving pointing distance (Figure 15). For Hidden Satellite, we also fit the mouse movement amount to  $S_{\text{targets}}$  (the sum of areas of all targets) for each combination of  $N$  and  $W$ . The resulted  $r^2$  value is 0.9987, which consists with the principle of the satellite cursor that pointing distance is mainly dependent on  $S_{\text{targets}}$ . Besides, from the obvious gap between the decreasing ratio of pointing distance and that of pointing time for Hidden Satellite, we know that the mouse speed is much slower for Hidden Satellite than Point. Since in this experiment participants were required to complete the task as fast as possible, we may expect Hidden Satellite further outperforms a standard cursor under more relaxed conditions.

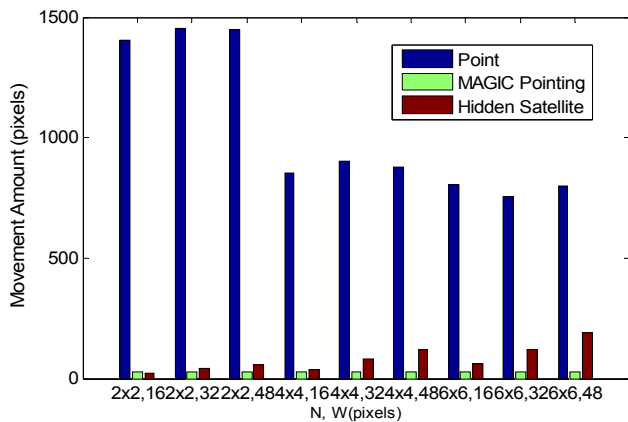


Figure 15: Mouse Movement Amount by  $N$ ,  $W$  values for all cursors

## Feedback and Observation

All participants preferred Hidden Satellite over Point for the advantage of saving motor effort and subjectively felt it was much faster as found in Experiment 1. Without informing the participants in advance that for the radial layout strategy, visually distant targets were also distant in the motor domain, three participants discovered it and thought it was helpful for planning mouse movement. Consistent with the user performance data, most participants found that the Hidden Satellite was easier to use for  $EA$  of medium length (50-200 pixels). Too small and too large  $EA$  both caused difficulties.

## Extended Design

### Design implications for real applications

Like other pointing techniques based on increasing  $W$  or decreasing  $A$ , the satellite cursor's performance depends on the amount of empty space. However, in many real applications, there is a large view region occupying most of the screen with little empty space left. One solution may be selecting with the satellite cursor by middle button and selecting with the system cursor by left button of the mouse, and only applying satellite cursor technique to the targets outside the view region.

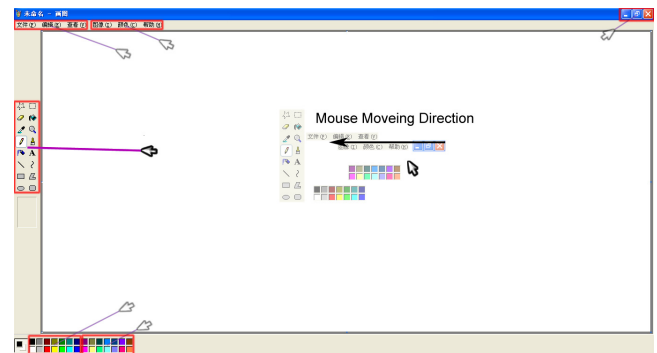
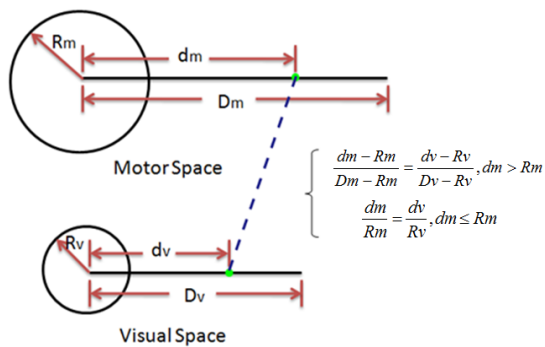


Figure 16: Illustrates the scenario when applying Satellite Cursor for MS Paint application to support target selection outside the painting area: selecting targets with satellite cursors and painting with the system cursor; clusters of targets are grouped into one (framed by red rectangles) and are associated with a single satellite cursor; only the satellite cursor which moves toward its target is visible while others are hidden, shown as semi-transparent cursors in the figure.

### Non-uniform mapping between visual and motor spaces

In the basic satellite cursor technique, the size of a target in visual space is same with that in motor space and so is the pointing vector. However, the satellite cursor enables non-uniform mapping in that both the target size and length of pointing vector can be different in the two spaces [4] if only the corresponding cursors enter into the target simultaneously in the two spaces. The mapping is illustrated in Figure 16. This feature can be used to control target size in motor space or limit the distance between a target and its satellite cursor in visual space.



**Figure 17: Non-uniform mapping between cursors' position in visual space and motor space. The green solid circles represent cursors' position in the two spaces. D (Dm or Dv) is the initial distance between the cursor and target and d (dm or dv) is the real time distance between the cursor and target.**

### CONCLUSIONS AND FUTURE WORK

The satellite cursor takes advantage of multiple cursors to reduce pointing amplitude in motor space. By assigning each target with a separate cursor in its vicinity, movement plan of the satellite cursor is not affected by the existence of distracter targets. Re-laying out targets in motor space using multiple cursors achieves both maximum usage of empty space and great flexibility in terms of both positions and sizes of the targets. An interesting aspect of the satellite cursor is that it realizes MAGIC pointing without gaze tracking and its performance can be well modeled by Fitts' Law using *EA*. Using this technique, the user first looks at the target, then locates the vicinal cursor and finally points with the cursor. An important factor that influences its pointing performance is the cursor uncertainty of direction and distance from the target, which determines the time users spend on locating the satellite cursor. In this paper, we proposed the radial layout strategy to address the directional uncertainty, three visual elements (line, ear and oriented cursor) to indicate the correspondence between a target and its satellite cursor and the hidden satellite cursor to mitigate visual clutter. The validity of those designs is demonstrated in the two controlled user studies, which also shows that the satellite cursor can significantly reduce mouse movement and pointing time, especially for sparse layouts of target.

Future work includes: investigating new re-layout strategies and target prediction algorithms to resolve cursor uncertainty; designing new visualization techniques to mitigate visual clutter; exploring re-layout strategies for complicate UI elements, such as menu, palette and etc; and applying this technique to real WIMP interfaces and explore its performance in practice.

### ACKNOWLEDEMENTS

This research is supported by National High-Tech Research and Development Plan of China under Grant No. 2009AA01Z336, China Postdoctoral Science Foundation funded project (Grant No. 20100470331) and MSRA.

### REFERENCES

- Asano, T., Sharlin, E., Kitamura, Y., Takashima, K., Kishino, F. (2005). Predictive interaction using the delphian desktop. *UIST*. p. 133-141.
- Baudisch, P., Cutrell, E., Robbins, D., Czerwinski, M., Tandler, P., Bederson, B., and Zierlinger, A. (2003). Drag-and-pop and drag-and-pick: Techniques for accessing remote screen content on touch- and pen-operated systems. *Proceedings of Interact*. p. 57-64.
- Baudisch, P., Zotov, A., Cutrell, E., Hinckley, K. (2008). Starburst: a target expansion algorithm for non-uniform target distributions. *AVI*. p. 129-137.
- Blanch, R., Guiard, Y., and Beaudouin-Lafon, M. (2004). Semantic pointing: improving target acquisition with control-display ratio adaptation. *CHI*. p. 519-525.
- Blanch, R., Ortega, M. (2009). Rake Cursor: Improving Pointing Performance with concurrent input channels. *CHI*. p. 1415-1418.
- Cockburn, A., Firth, A. (2003). Improving the acquisition of small targets. *HCI*. p. 181-196.
- Fitts, P.M. (1954). The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, 47. p. 381-391.
- Grossman, T., Balakrishnan, R. (2005). The bubble cursor: enhancing target acquisition by dynamic resizing of the cursor's activation area. *CHI*. p. 281- 290.
- Guiard, Y., Blanch, R., and Beaudouin-Lafon, M. (2004). Object pointing: a complement to bitmap pointing in GUIs. *Graphics Interface*. p. 9-16.
- Kabbash, P. and Buxton, W. (1995). The "Prince" technique: Fitts' law and selection using area cursors. *CHI*. p. 273-279.
- Kobayashi, M., Igarashi, T. (2008). Ninja cursors: using multiple cursors to assist target acquisition on large screens. *CHI*. p.949-958.
- Lank, E., Cheng, Y.-C.N., Ruiz, J. (2007). Endpoint prediction using motion kinematics. In Proc. *CHI*. p. 637-646.
- McGuffin, M., Balakrishnan, R. (2002) Acquisition of expanding targets. *CHI*. p. 57-64.
- McGuffin, M., Balakrishnan, R. (2005). Fitts' law and expanding targets: Experimental studies and designs for user interfaces. *TOCHI*. p. 12(4):388-422.
- Raiha, K.J., Spakov, O. (2009). Disambiguating ninja cursors with eye gaze. *CHI*. p. 1411-1414.
- Wobbrock J.O., Fogarty, J., Liu, S., Kimuro, S., and Harada, S. (2009). The angle mouse: target-agnostic dynamic gain adjustment based on angular deviation. *CHI*. p.1401-1410.
- Woodworth, R. (1899). The accuracy of voluntary movement. *Psychological Review Monograph Supplement*, 3.
- Worden, A., Walker, N., Bharat, K., Hudson, S. (1997). Making computers easier for older adults to use: area cursors and sticky icons. *CHI*. p. 266-271.
- Zhai, S., Morimoto, C., Ihde, S. (1999). Manual and Gaze Input Cascaded (MAGIC) Pointing. *CHI*. p. 246-253.