Peephole Pointing: Modeling Acquisition of Dynamically Revealed Targets

Xiang Cao, Jacky Jie Li, Ravin Balakrishnan

Department of Computer Science

University of Toronto

{caox, ravin}@dgp.toronto.edu, jacky.li@utoronto.ca

ABSTRACT

Peephole interaction occurs when a spatially aware display is moved and acts as a viewport to reveal different parts of the virtual space that cannot all fit within the display at once. We investigate pointing within this peephole metaphor, where the targets may not be initially visible on the display, but are dynamically revealed by moving the display. We develop and experimentally validate a quantitative model for peephole pointing. Our results indicate that the model accurately accounts for peephole pointing for a variety of display sizes, both with and without users' having prior knowledge of the target location.

Author Keywords

Pointing, Peephole, Fitts' law, modeling

ACM Classification Keywords

H5.2. Information interfaces and presentation (e.g., HCI): User Interfaces, Theory and methods.

INTRODUCTION

Although handheld computing devices are becoming ubiquitous and have significant computational power, they are inherently limited by their small screen size. As such, interaction techniques are required to view and interact with information that cannot be simultaneously displayed on the small display. Apart from virtually scrolling/panning the workspace on the display, researchers have investigated physically moving a spatially aware display that acts as a window to reveal different parts of the virtual space that is stationary relative to the physical world [6, 9, 15, 22, 28] (Figure 1). Such a spatially aware display can be either a physical screen on the device [9, 28], or a projection image created by a miniaturized projector embedded in the handheld [6, 22]. Various terms, including "Peephole" [28], "Flashlight" [6] and "Spotlight" [22] have been used to describe this interaction style; in this paper, we will use "peephole" to refer to this concept in general.

When the offscreen information's location is known – perhaps from previous visits or via techniques for visualizing off-screen targets such as Halo [5] – peephole navigation can be quick albeit by definition more demanding than when the information is already visible. Even more demanding is the case when the information's location is unknown and an exhaustive search of the virtual space must be undertaken. Either process is quite different from operating traditional interfaces where a larger proportion of the desired content is visible on-screen. As peephole interaction becomes deployable, it is important to study and model user performance when using it, in order to provide insights for future designs. In particular, existing models such as Fitts' law [8] have primarily been investigated in situations where the targets are always visible, and it is unclear if or how these models might apply to the peephole situation where targets have to first be revealed by spatial navigation before they can be acquired.

In this paper we study user performance in pointing tasks – a fundamental building block for higher-level interactions – where the display needs to be moved to reveal the target. Based on theoretical analysis we propose a quantitative model for such "peephole pointing", and experimentally validate it. Our work considers how user performance is affected by the size of the display and the presence or absence of prior knowledge of the target's location.
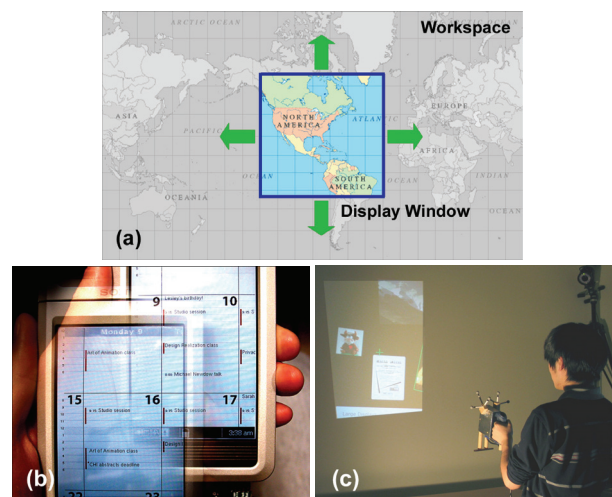

Figure 1. (a) Peephole display reveals a portion of the space. (b) Using a spatially aware PDA [28] (c) Using a spatially aware handheld projector [6]. (images reproduced with permission).

BACKGROUND

Several researchers have explored using a spatially aware display as a dynamic window for browsing and interacting with a larger workspace. Early work includes the Chameleon [9] – a Wizard-of-Oz implementation using a handheld TV, and the activeLENS [15] – an armature-mounted screen tracked by the joint angles of the arm. Similarly, the Boom Chameleon [25] uses an armature-mounted screen for viewing 3D scenes, augmented with a pen for pointing and annotation. For desktop style applications, Yee presents Peephole Displays [28], a spatially tracked PDA operated by a stylus. The virtual workspace is anchored to the user's physical reference frame. Simultaneous navigation and manipulation are supported by moving the display and the pen at the same time using two hands. Applications include drawing on a large canvas, viewing maps and browsing calendars. Instead of using a device with an embedded display screen, Cao and Balakrishnan [6] and Rapp et al. [22] explored using a handheld projector to create a projection display to reveal virtual workspaces residing on physical surfaces such as a wall or a table. Multiple workspaces can coexist to cover the entire physical environment, and information can be overlaid onto physical objects to augment them. A cursor that stays in the center of the projection image is used to interact with the workspace.

A few researchers have empirically studied navigation performance under the peephole metaphor. Mehra et al. [20] compared peephole navigation and traditional panning simulated on a desktop monitor, and reported that users perform better with peephole navigation. Rohs and Essl [23] compared several existing information navigation techniques on a camera-tracked handheld display. These studies differ from ours in that they focus on navigation tasks rather than pointing tasks. In addition, they only compared performance between interaction metaphors or techniques, but did not attempt to build a quantitative model.

Researchers have investigated "target acquisition" performance in scrolling tasks, which is somewhat similar to the task we study, in the sense that the target is not visible initially. Hinckley et al. [14] report that scrolling time follows Fitts' law [8] when the target's location is known, and Andersen [4] reports the scrolling time is linearly dependent on the distance to the target when its location is not known ahead of time. Both studies used a single window or display size, while we manipulate display size in our experiments and include it in our model.

Fitts' law [8] is broadly accepted to model aimed pointing tasks in general. The movement time T needed to point to a target of width W and at distance (or amplitude) A can be expressed as:

$$T = a + b \log_2 \left(\frac{A}{W} + c \right) \qquad (1)$$

where a and b are empirically determined constants, and c is 0, 0.5 or 1 (details discussed in MacKenzie [17]). The logarithmic term $\log_2(A/W + c)$ is referred to as the Index of Difficulty (*ID*) (measured in "bits") of the task. Kabbash and Buxton [16] show that Fitts' law also applies to the case that the cursor is an area while the target is a point, with W set as the width of the area cursor. Researchers have also expanded Fitts' law to model two- and three-dimensional targets [3, 11, 12, 18].

Although Fitts' law has been proved highly successful in modeling conventional pointing tasks, it is unclear whether it applies when the target may be outside the initial display region. Furthermore, it is likely that the size of the display also influences the task performance. We aim at developing a model for target acquisition performance in this peephole situation, which incorporates the display size S in addition to A and W.

PROBLEM SETUP

Our present work is one step towards providing a theoretical foundation and empirical data that can guide the design of interfaces utilizing the peephole metaphor. We sought to obtain an understanding of the factors that affect peephole pointing performance, and develop a quantitative model that characterizes such behavior.

For the sake of simplicity, in this research we choose to study a one-dimensional pointing task, with the constraint imposed by the target only in the direction collinear to the pointing motion. This is an idealized representation of real world pointing tasks, and is also the behavior that Fitts' law originally models. In practice, the targets are displayed as ribbons with infinite height (Figure 2).
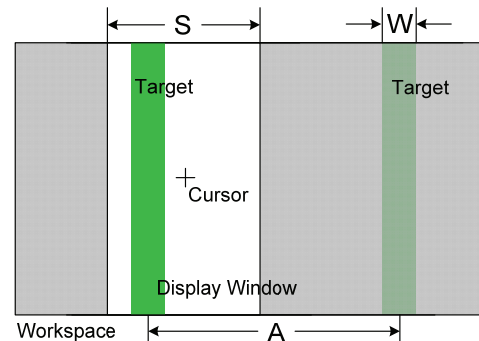

Figure 2. Problem Setup.

The following terms are used throughout our paper:

Workspace: the entire space in which the targets can reside. Only a portion of the workspace is visible and accessible at any given time.

Display Window: a window that can be moved within the workspace. The workspace region within the display window is visible and can be accessed by the user.

In practice, the shape of the display window may vary depending on the device and the purpose, and could be a

rectangle [9, 28], circle [22] or arbitrary quadrangle [6]. However, in our study, because the targets are 1D, we choose to make the display window 1D as well. The display window has a finite size along the direction collinear to the pointing motion, but extends infinitely along the direction perpendicular to the pointing motion, just as the targets do. By doing so, we maintain the 1D nature of the task. Using a 2D display window may impose unintended 2D constraints on the movement, since only the region within the display window is accessible.

A typical pointing task in this scenario consists of first moving the display window to reveal the target (if it is not inside the initial display window already), and then move the cursor to actually reach it. Three parameters characterize such a task:

A – distance between the initial cursor position and the center of the target.

W – width of the target.

S – size of the display window.

where A and W are variables included in the Fitts' law equation, while S is unique to peephole pointing, and thus needs to be incorporated into our model. In the case of a 1D display window, S is defined as the width of the display window, collinear to the pointing motion.

We also consider a few other factors that may affect the user's performance:

Prior knowledge of target location: when the target is not initially visible, whether the user has knowledge about the target location could affect the amount of effort required to search for the target, and in turn influence the task performance. The prior knowledge may come from previous visits to the target, or from techniques supporting visualization of off-screen targets such as Halo [5].

Cursor control mechanism: researchers have explored two mechanisms to control cursor position in these situations:

1. C*oupled cursor*: the cursor position is fixed with respect to the display window (typically in the center). Moving the display window also moves the cursor in the workspace. Thus the display also serves as a pointing device, and requires only one input to operate. This mechanism is used by most handheld projector systems [6, 22], where a dedicated pointing device is not available or preferable.

2. *Decoupled cursor*: the cursor position is controlled by a dedicated pointing device that is different from that controlling the display window's position. One possible scenario is where the user moves the display window using the non-dominant hand, while controls the cursor using the dominant hand. The cursor pointing device can move independently of the display window, but is sensed only when within it. This mechanism has been utilized on PDAs and handheld screens, using a pen for pointing [25, 28].

TENTATIVE MODEL

Building upon existing research, we perform a theoretical analysis of peephole pointing, and develop a tentative model for movement time that incorporates A, W and S.

In general, a peephole pointing task consists of two stages: first *M*oving the *D*isplay window to reveal the target (we call this *MD*), and then *M*oving the *C*ursor to hit the target (we call this *MC*). Note that this distinction is made more in terms of the cognitive process rather than necessarily the actual dynamics of the input device movements. In the coupled cursor case, there is no physical distinction between moving the display window and moving the cursor. Even in the decoupled cursor case (e.g. using a stylus on a PDA), the user may likely choose to rest the stylus (and the hand holding it) on the screen when the screen is moving. In this case the cursor is roughly synchronized with the display window until the user sees the target and stops the display window. The user will then adjust the cursor within the display window. Cognitively, the stylus moves in the reference frame defined by the display window, rather than by the world. On the other hand, *MD* and *MC* may in practice also overlap in time, and thus may not have a clear boundary. Nevertheless, we feel this conceptual distinction is useful in framing our analysis.

Depending on whether the user has prior knowledge of the target location, the performance of *MD* and *MC* can differ:

Case 1: Without Prior Knowledge
When the user does not know the target location, *MD* requires searching for the hidden target with the display window (Figure 3a). Unfortunately, to the best of our knowledge, there is no existing model for this type of searching. (Prior research on visual search [27] mainly focused on identifying a target among several distracters, with both target and distracters visible all the time, which is fundamentally different from our task). However, intuition and observations during our pilot studies suggest that the time required for *MD* (T_{MD1}) should increase as A increases, since a longer distance requires more time to search through. T_{MD1} should also decrease as S increases because: (1) a larger display window reaches the target earlier, resulting in a shorter search distance; (2) a larger display window makes it easier to visually identify the target; (3) the user tends to move the display window faster when it is larger, with less concern of missing the target (observed in both pilot study and formal experiment). With these considerations, we posit that T_{MD1} plausibly follows a Fitts' law relationship with A and S:

$$T_{MD1} = a_{D1} + b_{D1} \log_2\left(\frac{A}{S} + c\right) \qquad (2)$$

where a_{D1} and b_{D1} are constants that depend on the device and task setting, and c is 0, 0.5 or 1 (according to [17]). We neglect W in this equation as W does not impose a significant movement constraint in the *MD* stage, especially when W is considerably smaller than S.
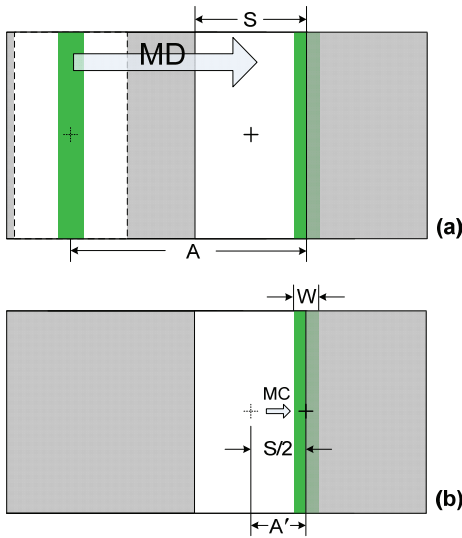
Figure 3. Peephole pointing without prior knowledge of the target location. (a) Moving the display window (*MD*) (b) Moving the cursor (MC)

Once the target is revealed, the user moves the cursor to reach the target (*MC*) (Figure 3b). *MC* is a standard pointing task that can be modeled by Fitts' law, where the target width is *W*, and the effective target distance A' is the distance between the target and the current cursor position. In the ideal situation, *MC* starts once the target is revealed in the display window, and the cursor is residing at the center of the display window at the time (which is always true in the coupled cursor case), then A' is between $(S - W) / 2$ (i.e., target fully revealed) and $(S + W) / 2$ (i.e., target barely revealed), with an average of $S/2$.

However, in practice, due to limits of reaction time, the user will often overshoot the display window before *MC* starts. Since in the *MD* stage the user tends to move faster with a larger display window, this overshoot is likely to be larger with a larger S. Further, in the decoupled cursor case the cursor may have a positive or negative offset (with a mean of zero) from the center of the display window, depending on the previous action, the way the pointing device is held etc. Considering the somewhat unpredictable nature of these factors, we assume the mean value of the effective target distance as $\overline{A'} = k_1 S$, where k_1 is some constant coefficient (because of overshooting, k_1 is likely smaller than 1/2). Therefore the average time needed for *MC* is

$$T_{MC1} = a_{C1} + b_{C1} \log_2\left(\frac{k_1 S}{W} + c\right) \qquad (3)$$

where a_{C1} and b_{C1} are the Fitts' law parameters for this setting, and *c* is 0, 0.5 or 1.

For the ease of mathematical derivation, we choose *c* to be 0 for both Eq. (2) and (3) (which is the equivalence of the original form of Fitts' law [8]), then total movement time is:

$$T_1 = T_{MD1} + T_{MC1}$$

$$= a_{D1} + b_{D1} \log_2\left(\frac{A}{S}\right) + a_{C1} + b_{C1} \log_2\left(\frac{k_1 S}{W}\right)$$

$$= a_{D1} + \log_2\left(\frac{A}{S}\right)^{b_{D1}} + a_{C1} + \log_2\left(\frac{S}{W}\right)^{b_{C1}} + b_{C1} \log_2 k_1$$

$$= (a_{D1} + a_{C1} + b_{C1} \log_2 k_1) + \log_2\left(A^{b_{D1}} S^{-(b_{D1} - b_{C1})} W^{-b_{C1}}\right) \qquad (4)$$

Because T_{MD1} describes searching without knowing the target location, which is more difficult than pointing to a visible target (described by T_{MC1}), we expect that $b_{D1} > b_{C1}$, resulting in a negative relationship between *S* and T_1.

Case 2: With Prior Knowledge
When the user already knows about the target location (such as from spatial memory of previous visits), *MD* is not truly a searching action, but a planned motion towards the target to cover it with the display window (Figure 4a). This is essentially the behavior of an area cursor, also modeled by Fitts' law as reported by Kabbash and Buxton [16], where *W* is set to be the width of the area cursor. Although they used a point target in their experiment, the relationship should still hold when the target width is considerably smaller than the area cursor. The only difference is that here the user does not see the target until it is revealed by the display window, and prior knowledge is used to guide the initial movement. Thus we have

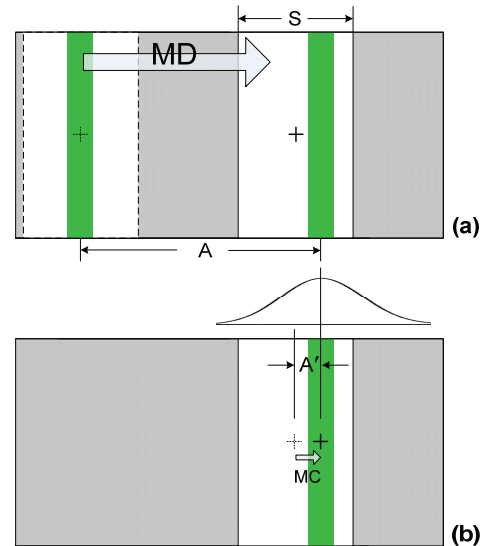$$T_{MD2} = a_{D2} + b_{D2} \log_2\left(\frac{A}{S} + c\right) \qquad (5)$$

Figure 4. Peephole pointing with prior knowledge of the target location. (a) Moving the display window (*MD*) (b) Moving the cursor (MC)

Similar to case 1, *MC* is a typical pointing action with target width *W* and effective target distance A' being the distance between the target and the current cursor position (Figure 4b). However, here A' is caused by the spread (distribution) of display window positions resulting from *MD*. *MD* is a

rapid approximate pointing task towards the target without initial visual cues, therefore has an open-loop (or "ballistic") characteristic to some extent. Several researchers [12, 21, 26] have investigated the spread of endpoints from such an action. They showed that (in the 1D case) the position of the endpoints X can be approximated by a normal distribution $N(X_0, \sigma)$, where the mean value X_0 is the center of the target, and the standard deviation $\sigma = fA$, f being an empirically determined constant. Therefore $A' = |X - X_0|$, and from the property of normal distribution we know the mean value $\overline{A}' = \sigma\sqrt{2/\pi} = fA\sqrt{2/\pi} = k_2 A$, where $k_2 = f\sqrt{2/\pi}$ is a constant coefficient. (Again, in the decoupled cursor case, the offset between the cursor and the center of the display window is averaged out in the mean). Thus,

$$T_{MC2} = a_{C2} + b_{C2}\log_2\left(\frac{k_2 A}{W} + c\right) \qquad (6)$$

Similarly, we choose $c = 0$ for both Eq. (5) and (6), and the total movement time

$$T_2 = T_{MD2} + T_{MC2}$$

$$= a_{D2} + b_{D2}\log_2\left(\frac{A}{S}\right) + a_{C2} + b_{C2}\log_2\left(\frac{k_2 A}{W}\right)$$

$$= a_{D2} + \log_2\left(\frac{A}{S}\right)^{b_{D2}} + a_{C2} + \log_2\left(\frac{A}{W}\right)^{b_{C2}} + b_{C2}\log_2 k_2$$

$$= (a_{D2} + a_{C2} + b_{C2}\log_2 k_2) + \log_2\left(A^{b_{D2}+b_{C2}} S^{-b_{D2}} W^{-b_{C2}}\right) \qquad (7)$$

Summarizing case 1 and 2 (Eq. (4) and (7)), we have

$$T = a + \log_2\left(A^{b_A} S^{-b_S} W^{-b_W}\right) \qquad (8)$$

where a, b_A, b_S, b_W are constants that depend on device and task property, and $b_A = b_S + b_W$ ($b_A, b_S, b_W > 0$). An alternate expression of Eq. (8) is

$$T = a + \log_2\left(A S^{-b_S/b_A} W^{-b_W/b_A}\right)^{b_A}$$

$$= a + \log_2\left(\left(\frac{A}{S}\right)^{b_S/b_A}\left(\frac{A}{W}\right)^{b_W/b_A}\right)^{b_A}$$

$$= a + \log_2\left(\left(\frac{A}{S}\right)^{n}\left(\frac{A}{W}\right)^{1-n}\right)^{b} \quad (let \quad n = b_S/b_A, \quad b = b_A)$$

$$= a + b\left(n\log_2\left(\frac{A}{S}\right) + (1-n)\log_2\left(\frac{A}{W}\right)\right) \qquad (9)$$

where a, b and n ($0 < n < 1$) are empirically determined constants that vary depending on the device and task property. Specifically, n describes the relative importance of S in terms of impact on movement time, compared to W.

Given the similarity between Eq. (9) and Fitts' law, we can define the Index of Difficulty for peephole pointing with a display window of size S as:

$$ID_S = n\log_2\left(\frac{A}{S}\right) + (1-n)\log_2\left(\frac{A}{W}\right) \qquad (10)$$

Inspired by the widely preferred variation of Fitts' law: $ID = \log_2(A/W + 1)$, we can propose a variation of Eq. (10):

$$ID_S = n\log_2\left(\frac{A}{S} + 1\right) + (1-n)\log_2\left(\frac{A}{W} + 1\right) \qquad (11)$$

so as to guarantee ID_S is positive, especially when $S > A$, which is valid in practice. Eq. (11) also suggests that

$$\lim_{S\to\infty} ID_S = (1-n)\log_2\left(\frac{A}{W} + 1\right) \qquad (12)$$

since the task becomes a standard pointing task when the display window is infinitely large, modeled by Fitts' law. Eq. (10) does not have this desirable property as it approaches $-\infty$ when $S\to\infty$.

Note that Eq. (8) may also be equivalently expressed as

$$T = a + b\left(n\log_2\left(\frac{A}{S}\right) + (1-n)\log_2\left(\frac{S}{W}\right)\right)$$

$$or \quad T = a + b\left(n\log_2\left(\frac{A}{W}\right) + (1-n)\log_2\left(\frac{W}{S}\right)\right) \qquad (13)$$

with a slightly different definition of b and n ($0 < n < 1$), while keeping the signs of $b_A, b_S,$ and b_W. Similar to Eq. (11), we may generate two other candidate variations:

$$ID_S = n\log_2\left(\frac{A}{S} + 1\right) + (1-n)\log_2\left(\frac{S}{W} + 1\right) \qquad (14)$$

$$ID_S = n\log_2\left(\frac{A}{W} + 1\right) + (1-n)\log_2\left(\frac{W}{S} + 1\right) \qquad (15)$$

Eq. (14) is reasonable given its similarity to our derivation of Eq. (4) in case 1. However, it does not regress to Fitts' law when $S\to\infty$. Eq. (15) satisfies this regression property, but does not have an intuitive interpretation. In addition, Eq. (11) yields $ID_S = 0$ when $A = 0$, which is intuitive since the required movement is completely eliminated. Neither Eq. (14) nor Eq. (15) satisfies this requirement.

Nevertheless, we will use our experiment data to verify each variation of ID_S (Eq. (10), (11), (14), (15)) and determine which model best fits the data.

EXPERIMENT SETUP

Apparatus

We choose to simulate a peephole display on a desktop monitor (19-inch, 1600 x 1200 pixels) (Figure 5). The entire screen represents the workspace, with only the region corresponding to the display window revealed, and the rest (virtually) masked in black. A Wacom™ Intuos2 tablet (12 x 18 inch) is used for input. The decision to simulate the peephole display on a desktop computer rather than using a spatially aware handheld device was to ensure the most

reliable and highest quality possible input and output devices, and to enable the abstract 1D display window in our experiment. This prevents our results from being confounded by tracking errors and other performance limitations of current spatially aware handheld displays.
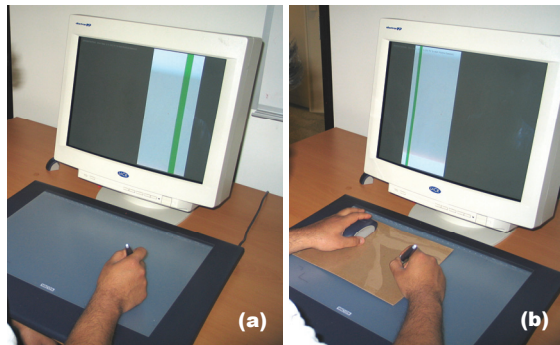

Figure 5. Experiment Setup. (a) Coupled Cursor. (b) Decoupled Cursor.

Techniques

Two techniques are investigated:

Coupled cursor (Figure 5a): a stylus is used to control the position of both the display window and the cursor when held over the tablet. The cursor, displayed as a crosshair, always stays at the center of the display window.

Decoupled cursor (Figure 5b): a Wacom™ puck held in the non-dominant hand can be moved on the tablet to control the position of the display window. The dominant hand holds the stylus used to control the cursor position, independent of the display window. Although the user is free to move the cursor outside the display window and still able to see it, the system only responds to the stylus tap (for selecting the target) when the cursor is inside. To simulate the real world scenario of using a stylus on a moving PDA (as in [28]), a thin board is glued to the bottom of the puck and moves with it. The user can tap the stylus on the board, and rest the stylus hand on it while moving the puck.

Task

A reciprocal 1D pointing task is used, in which both the display window and the targets are one-dimensional, vertically extending to the border of the screen. In each sequence of trials, the participant selects two equal-width targets back and forth in succession. The distance between the targets is set according to the experiment condition, while the exact positions of them are randomized for each sequence, so as to prevent the participant from guessing the target position. Within the same sequence the target positions remain unchanged across the trials. At the beginning (first trial) of each sequence, the position of the first target is displayed as a red X, in order to guide the participant to move the display window toward it and reveals the first target. However, before the display window actually moves over the red X, the display window is rendered only as a wireframe silhouette without revealing the content inside. This is to prevent the participant from

seeing the second target if the display window happens to pass it. Once the display window reaches the red X, it reverts to its normal appearance, and the first target is revealed. The participants can then use the cursor to select it (by tapping the stylus), and immediately starts to search for (if not already inside the display window) the second target. The position of the first target provides a cue as to which direction to search. For example, if the first target lies in the left half of the screen, the second target will always be to its right (but not necessarily in the right half of the screen), and vice versa.

After selecting the second target, the participant moves back to select the first one, and so forth. Note that unlike a conventional reciprocal pointing task, where all trials in a sequence are essentially equivalent, in our task the first trial is fundamentally different from the others in the sequence. The first trial (i.e. the first pointing action from the first target towards the second target) represents pointing *without prior knowledge* (apart from the directional cue discussed above) and requires extensive searching, but in all following trials the participant has already gained knowledge about the target location from previous selections. This provides us a way to cover both conditions of *prior knowledge of target location* in a single sequence.

The participant must successfully select the target before s/he can proceed to the next trial. Once the target is selected, its color turns from green to gray, and a short beep is played to indicate success.

Since we are only interested in the movement along the horizontal axis, we constrained the cursor to be always vertically centered on the screen. The input area of the tablet is cropped to provide a 1:1 control-display gain. All buttons on the puck and the stylus are disabled.

We measured the completion time (including possible error correction time), and the number of errors made in each trial.

Design

A fully crossed within-participant factorial design was used. Independent variables were target distance A (128, 256, 512, 1024 pixels), target width W (8, 16, 32, 64 pixels), display window size S (32, 64, 128, 256, 512 pixels), cursor control mechanism (coupled, decoupled), and *prior knowledge of target location* (yes, no). The range of A, W and S values covered both typical and extreme (e.g. $S < W$) scenarios.

Each participant used both cursor control mechanisms, within which, three consecutive blocks were performed. Within every block, each combination of A, W and S were tested for one sequence of trials. The presentation order of these combinations was randomized within each block. We chose to have only 2 trials in each sequence, in which the first trial (i.e., pointing from the first target towards the second target) represents pointing *without prior knowledge*, and the second trial (i.e., returning from the second target to the first target) represents pointing *with prior knowledge*, so

that we have equal numbers of data points for both conditions. We evenly randomized the movement direction (left or right) in the first trial (the second trial has the reverse direction), in order to prevent the movement direction from compounding with the *prior knowledge* factor. An input error was counted when the participant taps outside the target. Participants were asked to keep the error rate under 4% during each block. 20 practice trials were performed before each cursor control condition started.

As a baseline, each participant also performed a standard pointing task, with the entire workspace always visible (i.e. the display window is infinitely large). The participant used the stylus to control the cursor. A similar experiment structure was used for the baseline task, keeping all factors but the display window size *S*. Each participant performed the peephole pointing task using two different cursor control mechanisms plus the baseline task, with the order of the three counterbalanced across participants.

Participants
12 right-handed volunteers, aged 18–33, participated. The experiment lasted about 1.5 hours for each participant. Participants were encouraged to take breaks between blocks.

EXPERIMENT RESULTS

Movement Time
For the peephole pointing task, the overall average movement time *T* is 2118 ms. *T* increases monotonically as *A* increases, and decreases monotonically as *W* or *S* increases, as suggested by our proposed model. An analysis of variance showed that all these effects are statistically significant ($p<.001$). Pairwise means comparisons also showed that every condition of *A*, *W*, or *S* is significantly different from the others ($p<.001$).

There was a significant main effect for *cursor control mechanism* ($F_{1,11} = 55.4$, $p<.001$), with mean times of 1884 ms for *coupled cursor* and 2351 ms for *decoupled cursor*. Different *cursor control mechanisms* result in different behaviors, and in turn different performance. However, the trend that *coupled cursor* performs faster than *decoupled cursor* should not be regarded as a general conclusion, since it may depend on the specific device used. For example, we might expect *coupled cursor* to become slower if we used the non-dominant hand puck to control the cursor rather than the dominant hand stylus.

There was also a significant main effect for *prior knowledge of target position* ($F_{1,11} = 491$, $p<.001$), with the first trial (*without prior knowledge*) of each sequence averaging 2562 ms, while the second trial (*with prior knowledge*) averaging 1674 ms. As a comparison, for the baseline task, there is no significant difference ($F_{1,11} = 2.14$, $p = .171$) between the two trials. Therefore we conclude that the performance difference between the two trials in the peephole pointing task is indeed caused by the knowledge of target position, rather than a practice effect.

There were significant interactions for *cursor control mechanism* x *prior knowledge* ($F_{1,11} = 4.83$, $p = .050$), *S* x *cursor control mechanism* ($F_{4,44} = 32.8$, $p < .001$), and *S* x *prior knowledge* ($F_{4,44} = 89.5$, $p<.001$). In particular, *S* has a larger effect on *T* when there is no prior knowledge of the target location (Figure 6). This can explained that without prior knowledge, *MD* is a pure searching task that relies heavily on the visual feedback in the display window, while with prior knowledge *MD* has a more ballistic nature, for which the display window plays a less important role. However, the effect of *prior knowledge* decreases as *S* increases. When *S* is small, only a small amount of information is revealed in the display window, so any prior knowledge of the target location would add considerable information and significantly benefit the task performance. As *S* becomes larger, the additional information gained from prior knowledge becomes smaller, resulting in less performance gain. Eventually when *S* approaches infinity, the task becomes a standard pointing action, and the effect of prior knowledge is eliminated, as shown by the lack of significant difference between trials in the baseline task.
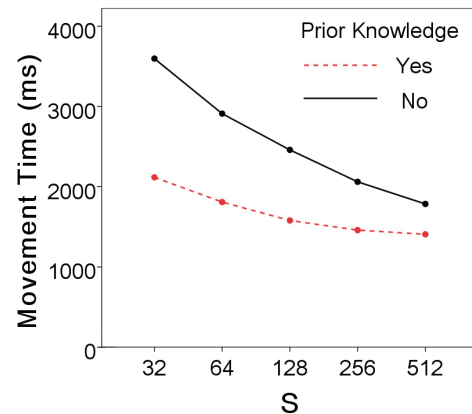

Figure 6. Movement time by display window size and prior knowledge.

There was a significant main effect for block number on *T* ($F_{2,22}=18.7$, $p<.001$), indicating learning between blocks.

Errors
An error occurred when the user tapped outside the intended target. The overall average error rate was 3.5%. The factors that significantly affected the error rate include: *cursor control mechanism* ($F_{1,11} = 7.50$, $p=.019$) with 2.9% errors for *coupled cursor* and 4.1% errors for *decoupled cursor*, possibly caused by the coordination skills required for bimanual input; *W* ($F_{3,33} = 27.3$, $p<.001$), where the error rate increases as *W* decreases, because of the increased task difficulty; and *S* ($F_{4,44} = 2.65$, $p=.046$). The error rate is lowest (2.4%) when *S* = 32, largest (4.4%) when *S* = 512, and roughly 3.5% for all other conditions between. This is an interesting and somewhat counterintuitive phenomenon, which may be because users became over-careful with a tiny display window, and over-relaxed with a huge display window. No significant interaction exists between any factors for error rate.

ID_S	Cursor Control Mechanism	Prior Knowledge	a (ms)		b (ms/bit)		n		R^2
			Estimate	Std. Err.	Estimate	Std. Err.	Estimate	Std. Err.	
$n\log_2\left(\frac{A}{S}\right)+(1-n)\log_2\left(\frac{A}{W}\right)$	Coupled	Yes	444	46.9	316	12.1	0.304	0.0302	0.900
		No	561	115	652	29.5	0.533	0.0363	0.868
	Decoupled	Yes	615	65.8	437	16.9	0.441	0.0306	0.897
(Eq. 10)		No	782	139	796	35.6	0.575	0.0363	0.873
$n\log_2\left(\frac{A}{S}+1\right)+(1-n)\log_2\left(\frac{A}{W}+1\right)$	Coupled	Yes	171	46.2	382	13.2	0.376	0.0281	0.923
		No	-168	108	843	31.0	0.606	0.0287	0.906
	Decoupled	Yes	175	63.4	548	18.1	0.515	0.0260	0.924
(Eq. 11)		No	-143	128	1040	36.7	0.646	0.0275	0.913
$n\log_2\left(\frac{A}{S}+1\right)+(1-n)\log_2\left(\frac{S}{W}+1\right)$	Coupled	Yes	-244	67.3	707	26.3	0.625	0.00974	0.913
		No	-697	160	1276	62.6	0.720	0.0150	0.891
	Decoupled	Yes	-259	96.9	899	37.8	0.679	0.0120	0.905
(Eq. 14)		No	-737	187.3	1524	73.1	0.738	0.0152	0.902
$n\log_2\left(\frac{A}{W}+1\right)+(1-n)\log_2\left(\frac{W}{S}+1\right)$	Coupled	Yes	-84	65.8	743	54.2	0.454	0.0268	0.897
		No	-942	205	1948	169	0.341	0.0241	0.780
	Decoupled	Yes	-276	112	1189	92.1	0.380	0.0238	0.845
(Eq. 15)		No	-1176	255	2514	210	0.321	0.0221	0.775

Table 1. Summary of model fitting results.

Model Fitting

We fit the movement time T to the four candidate formulations (Eq. (10), (11), (14), (15)) of ID_S using least-square regression. Since both *cursor control mechanism* and *prior knowledge of target location* affect T, we break down the data into four categories according to these two factors, and fit them separately. Table 1 shows the parameter estimates and standard errors for those estimates. The last column provides the R^2 values for the regression.

Eq. (11) has the best fit with experiment data in all 4 categories, with R^2 always greater than 0.9 (Figure 7). In addition, Eq. (11) has several desirable properties as discussed earlier. Therefore our model for peephole pointing is best represented as:

$$T = a + b\left(n\log_2\left(\frac{A}{S}+1\right)+(1-n)\log_2\left(\frac{A}{W}+1\right)\right) \quad (16)$$

where a, b and n ($0 < n < 1$) are empirically determined constants that may vary depending on the device and task.

As a baseline, we also fit the data to the three variations of standard Fitts' law: $ID = \log_2(A/W+1)$, $ID = \log_2(A/S+1)$, and $ID = \log_2(S/W+1)$, which all yield average R^2 under 0.66. This further confirmed the advantage of our proposed model which yielded significantly better fits.

In particular, n describes the relative importance of display window size S in terms of impact on the movement time. Compared to a and b, which reflect mainly the performance of the device or user group, n captures more of the essence of the task itself. As an example, we compare the n value between the four categories we fit to:

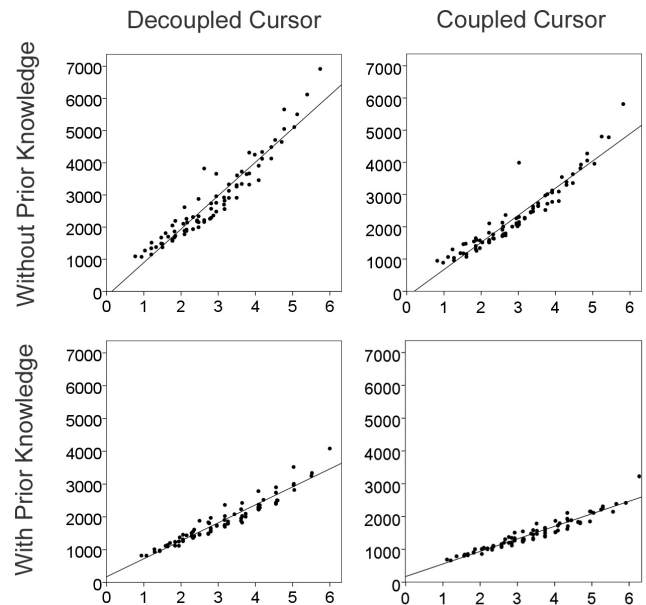| n	Coupled Cursor	Decoupled Cursor
With Prior Knowledge	0.376	0.515
Without Prior Knowledge	0.606	0.646

Figure 7. Model Fitting of Eq. (11). (horizontal axis: ID_S ; vertical axis: movement time (ms))

For both *cursor control mechanisms*, n is smaller when there is prior knowledge about the target location, indicating a lesser impact of S. This is validated by the data trend we analyzed in the previous section. However, n is larger in the *decoupled cursor* condition than in the *coupled cursor* condition. The fact that the stylus needs to be inside the display window to be responsive imposes an additional motor constraint related to S, while this constraint does not exist in the coupled cursor case. This may explain the larger impact of S in the *decoupled cursor* condition. Further, the difference of n caused by *cursor control mechanism* is smaller *without prior knowledge*, since the searching behavior that accounts for the majority of the task effort is mainly a cognitive process, relatively independent of the

control mechanism. When prior knowledge becomes available, the cognitive aspect of the task is reduced and the motor control aspect dominates, which is more vulnerable to influences of the control mechanism.

In addition, Eq. (16) shows that when S becomes sufficiently large (e.g. when $S > 2A$, the target is inside the initial display window), the benefit of further enlarging it becomes marginal. For example, enlarging S from $2A$ to $4A$ only results in ID_S decreasing by $0.26n$, versus $0.74n$ when S changes from $A/4$ to $A/2$. This can also be observed from the trend in Figure 6: the relationship curve becomes flatter as S increases, and will eventually converge as $S\rightarrow\infty$, at which point the action becomes a standard pointing task. Our model suggests that the movement time follows Fitts' law as $S\rightarrow\infty$, validated by the unsurprising good fit of the baseline task performance to Fitts' law ($R^2 = 0.976$).

DISCUSSION

One interesting observation about our model is its mathematical similarity to the very first model for 2D pointing tasks, suggested by Crossman [7]:

$$T = a + b\log_2\left(\frac{A}{W}+1\right) + c\log_2\left(\frac{A}{H}+1\right)$$ (W and H are target

width and height respectively), which can be reformulated to resemble Eq. (16). This model suggests that W and H affect T independently. However, better models for 2D pointing tasks have been developed later on [3, 12, 18], which more properly address the strong interaction found between W and H. In our task, the somewhat separable stages of MD and MC make it more appropriate to tackle S and W separately in the model, as supported by its good match with experimental data.

As we discussed, the parameter n captures in large the nature of a given peephole pointing task, and is less influenced by the device used. Thus to analyze behavior under a particular setting, we may first acquire the n value (by theoretical analysis or empirical study), and make design decisions according to it. In general, a larger n means we need to focus on increasing the display size (or virtual size, such as using a Fisheye [10] visualization); while a smaller n suggests we should endeavor to enlarge the interface components (or enlarge their effective width, such as by dynamically expanding the targets as the cursor approaches them [19]).

In our experiment, we dealt with cases that the user has either no prior knowledge of the target location, or near-perfect knowledge of it (all participants expressed they could remember very well the target location during the second trials). However in practice there are many cases that lie between these two extremes. For example, the user may obtain partial (or "soft") knowledge through visualizations of off-screen targets. We expect the resulting performance also lies somewhere between the two cases we studied. How exactly the performance varies with the amount (and format) of the information available is worth further investigation.

During the experiment we observed different strategies employed by participants to search for the targets. Some people moved the display window at a more or less constant speed within each trial, until they saw the target and stopped the display window. Some others first moved the display window in a quick movement, hoping to get a glance of the target, and then backtrack to it. This strategy might provide some benefit with large display windows, but was not always successful when the display window was small. In the latter case, the participant might have to revert to the first strategy after an initial failure to find the target, resulting in performance loss. Analysis of the movement logs indicates 32.1% of the trials involved some amount of backtracking of the display window, including cases of either the intentional quick movement strategy, or an unintentional overshooting. The number of backtrackings decreases as S becomes larger. Regardless of the variety of the strategies, our model is general enough to account for the data. However, it is worthwhile to consider how one might accommodate these strategies when designing peephole interfaces, for example making the interface components more "glanceable".

It is worthwhile to note that the *decoupled cursor* mechanism is a typical example of asymmetric bimanual action [13], where the non-dominant hand performs coarse movements and sets the reference frame, within which the dominant hand performs fine movements and operations. Analysis of the movement logs revealed that the cursor resides inside the display window for 88.0% of the movement time on average. The instant movement velocity of the cursor (calculated every 0.1 sec, relative to the display window) has a mean of 65.7 pixels/sec versus 223.9 pixels/sec for the display window (absolute). These both conform to the asymmetric work division pattern between the non-dominant and dominant hands.

CONCLUSIONS & FUTURE WORK

We have presented an experimental study of pointing task where the targets are initially invisible and revealed by a moving display window (i.e., peephole pointing). Based on theoretical analysis, we proposed a model (with several variations) for the movement time under such circumstances, taking into account the size of the display window (S):

$$T = a + b\left(n\log_2\left(\frac{A}{S}+1\right) + (1-n)\log_2\left(\frac{A}{W}+1\right)\right)$$

Our model was validated by the experimental data under various settings including different cursor control mechanisms, and both with and without prior knowledge of the target location. We also analyzed the effects of these settings on the movement time, and discussed implications of our research. To the best of our knowledge, our work is the first attempt in modeling user performance in peephole pointing. We hope it inspires more research in this direction.

Our current research focuses on the movement time, and only touches the surface of error analysis. It would certainly

be worthwhile to investigate the speed-accuracy tradeoff in such a task, as researchers [24, 29] have explored with standard pointing tasks. Using our current research framework, we are also interested in studying other generic tasks such as crossing [2], steering [1] or pointing at 2D targets with a 2D display window. Finally, our evaluation used simulated apparatus with high quality tracking and display in order to ensure reliable data. It is worth testing our model using a spatially aware handheld device or projector, which would give a sense of how the model generalizes to different physical movement styles and tracking noise associated with such devices.

ACKNOWLEDGEMENTS

REFERENCES

1. Accot, J. and Zhai, S. (1997). Beyond Fitts' Law: Models for trajectory-based HCI tasks. *ACM CHI Conference*. p. 295-302.
2. Accot, J. and Zhai, S. (2002). More than dotting the i's - foundations for crossing-based interfaces. *ACM CHI Conference*. p. 73-80.
3. Accot, J. and Zhai, S. (2003). Refining Fitts' law models for bivariate pointing. *ACM CHI Conference*. p. 193-200.
4. Andersen, T.H. (2005). A simple movement time model for scrolling. *ACM CHI Conference, extended abstracts*. p. 1180 - 1183.
5. Baudisch, P. and Rosenholtz, R. (2003). Halo: a technique for visualizing off-screen objects. *ACM CHI Conference*. p. 481-488.
6. Cao, X. and Balakrishnan, R. (2006). Interacting with dynamically defined information spaces using a handheld projector and a pen. *ACM UIST Symposium*. p. 225-234.
7. Crossman, E. (1956). The measurement of perceptual load in manual operations. University of Birmingham.
8. 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.
9. Fitzmaurice, G.W. (1993). Situated information spaces and spatially aware palmtop computers. *Communications of the ACM*, 36(7). p. 38-49.
10. Furnas, G. (1986). Generalized fisheye views. *ACM CHI Conference*. p. 16-23.
11. Grossman, T. and Balakrishnan, R. (2004). Pointing at trivariate targets in 3D environments. *ACM CHI Conference*. p. 447-454.
12. Grossman, T. and Balakrishnan, R. (2005). A probabilistic approach to modeling two-dimensional pointing. *ACM Transactions on Computer Human Interaction*, 12(3). p. 435-459.
13. Guiard, Y. (1987). Asymmetric division of labor in human skilled bimanual action: The kinematic chain as a model. *Journal of Motor Behavior*, 19(4). p. 486-517.
14. Hinckley, K., Cutrell, E., Bathiche, S., and Muss, T. (2002). Quantitative analysis of scrolling techniques. *ACM CHI Conference*. p. 65 - 72.
15. Ishii, H. and Ullmer, B. (1997). Tangible bits: towards seamless interfaces between people, bits and atoms. *ACM CHI Conference*. p. 234-241.
16. Kabbash, P. and Buxton, W. (1995). The "Prince" technique: Fitts' law and selection using area cursors. *ACM CHI Conference*. p. 273-279.
17. MacKenzie, S. (1992). Fitts' law as a research and design tool in human-computer interaction. *Human-Computer Interaction*, 7. p. 91-139.
18. MacKenzie, S. and Buxton, W. (1992). Extending Fitts' law to two-dimensional tasks. *ACM CHI Conference*. p. 219-226.
19. McGuffin, M. and Balakrishnan, R. (2002). Acquisition of expanding targets. *ACM CHI Conference*. p. 57-64.
20. Mehra, S., Werkhoven, P., and Worring, M. (2006). Navigating on handheld displays: Dynamic versus static peephole navigation. *ACM Transactions on Computer Human Interaction*, 13(4). p. 448-457.
21. Meyer, D.E., Smith, J.E., Kornblum, S., Abrams, R.A., and Wright, C.E. (1988). Optimality in human motor performance: Ideal control of rapid aimed movements. *Psychological Review*, 95. p. 340-370.
22. Rapp, S., Michelitsch, G., Osen, M., Williams, J., Barbisch, M., Bohan, R., Valsan, Z., and Emele, M. (2004). Spotlight Navigation: Interaction with a handheld projection device. *Pervasive Conference, Video paper*.
23. Rohs, M. and Essl, G. (2006). Which one is better?: information navigation techniques for spatially aware handheld displays. *International Conference on Multimodal Interfaces*. p. 100 - 107.
24. Schmidt, R.A., Zalaznik, H.N., Hawkins, B., Frank, J.S., and Quinn, J.T. (1979). Motor output variability: A theory for the accuracy of rapid motor acts. *Psychological Review*, 86. p. 415-451.
25. Tsang, M., Fitzmaurice, G.W., Kurtenbach, G., Khan, A., and Buxton, B. (2002). Boom chameleon: simultaneous capture of 3D viewpoint, voice and gesture annotations on a spatially-aware display. *ACM UIST Symposium*. p. 111-120.
26. Welford, A. (1968). The fundamentals of skill. London: Methuen.
27. Wolfe, J.M. (1998). Visual Search. In *Attention*, H. Pashler, Editor. University College London Press: London, UK.
28. Yee, K.-P. (2003). Peephole displays: pen interaction on spatially aware handheld computers. *ACM CHI Conference*. p. 1-8.
29. Zhai, S., Kong, J., and Ren, X. (2004). Speed-accuracy trade-off in Fitts' law tasks - on the equivalency of actual and nominal pointing precision. *International Journal of Human-Computer Studies*, (Special Issue of Fitts (1954) 50th Anniversary).