Understanding the Uncertainty in 1D Unidirectional Moving Target Selection

Jin Huang[1, 2, 3]
huangjin@iscas.ac.cn

Feng Tian[1, 2, 3*]
tianfeng@iscas.ac.cn

Xiangmin Fan[1, 2]
xiangmin@iscas.ac.cn

Xiaolong (Luke) Zhang[4]
lzhang@ist.psu.edu

Shumin Zhai[5]
zhai@acm.org

[1] State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing, China
[2] Beijing Key Lab of Human-Computer Interaction, Institute of Software, Chinese Academy of Sciences, Beijing, China
*Coresponding author: tianfeng@iscas.ac.cn

[3] School of Computer and Control Engineering, University of Chinese Academy of Sciences, Beijing, China
[4] Pennsylvania State University, University Park, USA
[5] Google Inc., Mountain View, USA

ABSTRACT

In contrast to the extensive studies on static target pointing, much less formal understanding of moving target acquisition can be found in the HCI literature. We designed a set of experiments to identify regularities in 1D unidirectional moving target selection, and found a *Ternary-Gaussian* model to be descriptive of the endpoint distribution in such tasks. The shape of the distribution as characterized by μ and σ in the Gaussian model were primarily determined by the speed and size of the moving target. The model fits the empirical data well with 0.95 and 0.94 R^2 values for μ and σ, respectively. We also demonstrated two extensions of the model, including 1) predicting error rates in moving target selection; and 2) a novel interaction technique to implicitly aid moving target selection. By applying them in a game interface design, we observed good performances in both predicting error rates (e.g., 2.7% mean absolute error) and assisting moving target selection (e.g., 33% or a greater increase in pointing accuracy).

Author Keywords

Moving Target Selection, Endpoint Distribution, Error Rate Prediction, Pointing Accuracy

ACM Classification Keywords

H.5.2 [**Information interfaces and presentation**]: User interfaces – *theory and methods*; H.1.2 [**Models and principles**]: User/machine systems – *human factors*.

INTRODUCTION

Interactive systems with dynamic content, such as video games, traffic control displays and video surveillance systems are becoming ubiquitous nowadays. The development of the Augmented Reality (AR) and Virtual Reality (VR) technologies enables even richer interactions with the rapidly changing environment. The selection of moving targets in these dynamic contexts is a common yet challenging task. Users need to continually track the moving target and simultaneously plan the timing for selection. Such actions demand high sensory-motor coordination [28]. Compared with static target selection, acquiring moving targets can leads to longer task completion time and higher error rate.

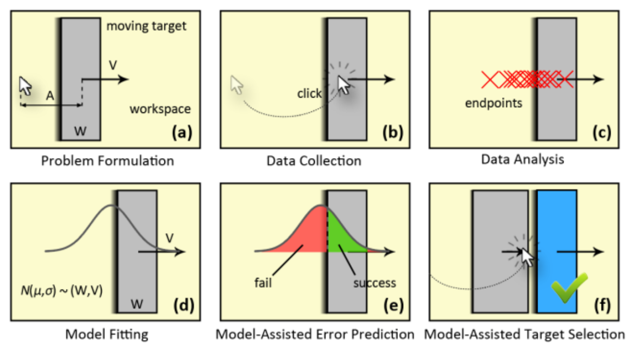

Figure 1. A snapshot of our work. We formally define the problem of moving target selection (a). We propose a *Ternary-Gaussian* model to interpret endpoint distribution in moving target selection (b, c, d). We demonstrate two extensions of the model: 1) predicting pointing errors (e), and 2) assisting target selection (f).

A variety of novel and effective techniques have been proposed to aid the selection of moving targets. These techniques improve the acquisition efficiency by either *enlarging the activation area* or *reducing the movement speed* of the target. For example, the *Comet* technique [11, 13] attaches a selectable tail with each target and the size of the tail is based on the width and speed of the target. With the *Click-to-Pause* technique [12], a user pauses the whole scene (i.e. reduces the moving speed to *zero*) when pressing the mouse button. Then the user moves the cursor over the target and releases the button to select it. Similarly, to avoid the potential loss of information, the *Target Ghost* [13] creates a static proxy for each target. The user then selects the easy-to-reach proxy without pausing the whole scene.

CHI 2018, April 21–26, 2018, Montreal, QC, Canada.

DOI: https://doi.org/10.1145/3173574.3173811

Despite the effectiveness of these techniques, many of the design decisions and parameters are largely ad hoc. We attribute this to the insufficient understanding of human performance in these scenarios. As a result, it is still uncertain whether the chosen parameters can release the full potential of these techniques.

Even though extensive HCI studies have investigated human performance in target pointing, e.g., understanding endpoint distribution [34, 5, 6] or predicting movement time (Fitts' law [8]), the vast majority of them focus on acquiring static targets. Although the findings of these studies have guided the design of applications numerous times in the past, they cannot fully explain how people perform moving target selection. In this paper, we present a first step in understanding human performance in moving target selection. Specifically, we investigate the selection uncertainty manifested in endpoint distribution in 1D unidirectional moving target selection tasks. We constructed a set of experiments to identify the regularities in moving target selection, and proposed a *Ternary-Gaussian* model to be descriptive of the endpoint distribution in such tasks. The model fitted the empirical data well with 0.95 and 0.94 R^2 values for μ and σ, respectively. We also demonstrated two extensions of the model, including error rate prediction and assisting target selection. By applying these extensions in a game interface design, we observed good performances in both predicting the error rates (e.g., 2.7% mean absolute error) and improving the pointing accuracy (e.g., 33% or a greater increase in pointing accuracy). Figure 1 shows the snapshot of our work.

Our research makes the following contributions:

- we found that the endpoint distribution in moving target selection tasks is affected by the speed and size of the target, not by the initial distance between the cursor and the target;
- we found a *Ternary-Gaussian* model to be descriptive of the movement endpoint distribution;
- we applied and extended the model in two task scenarios: including 1) predicting pointing error rates during the interface design process, and 2) providing implicit aid to the selection of moving targets without modifying the visual appearance of the interface;
- we evaluated the applications in a game interface design and observed good performances in both predicting the error rates and improving the pointing accuracy.

RELATED WORK

Our work was inspired by prior work on both designs and theoretical models for both static and dynamic target selection. Table 1 shows the existing works and the relationship between these works with the current literature.

Static Target Selection

In HCI community, Fitts' law [8, 19] is the most famous user performance model which describes the rule of speed-accuracy tradeoff and predicts the average movement time (*MT*) to select a target with a given distance to the cursor and width. Fitts' law was extended by many studies in higher dimensions (2D [3, 20] and 3D [9, 23]) and complex interfaces, such as trajectory-based [1, 2] and cross-based [4] tasks, revealed targets [7] and gaze targets [22] selection. However, Fitts' law and its extensions cannot describe users' performance on accuracy because they do not take selection uncertainty into account.

Selection uncertainty is mostly investigated as endpoint distribution in the HCI literature, since it directly impacts the interaction results and can be easily described with statistics methods. One of the most common benefits from studying endpoint distribution is to replace the nominal target width with the *effective width* [33, 37] in Fitts' law, which is defined by the mean and standard deviation of the distribution function. It was found that effective width can partially, while not completely, compensate speed variance from the prediction based on nominal width [37]. Recently, Bi et al. [5] considered the absolute precision of pointing devices and proposed a dual Gaussian model, as an extension of Fitts' law in touch-based selection tasks.

<table>
<tr><th colspan="2">category</th><th>static</th><th>dynamic</th></tr>
<tr><td rowspan="3">theoretical</td><td>movement time</td><td>Fitts' law [8]</td><td>Jagacinski's model [16]</td></tr>
<tr><td>endpoint distribution</td><td>Effective Width [33]
FFitts' law [5, 6]</td><td>Ternary-Gaussian model</td></tr>
<tr><td>error rate prediction</td><td>Wobbrock's model [34]</td><td>Temporal Pointing [18]
Error Model</td></tr>
<tr><td>practical</td><td>selection technique</td><td colspan="2">Expending Target [27]
Bubble Cursor [10]
Comet and Ghost [11, 13]
BayesPointer</td></tr>
</table>

Table 1. Related work divided into 7 categories.

Dynamic Target Selection

Compared with static targets, prior studies on dynamic targets are more relevant to our research. Jagacinski et al. [16] showed that selection of moving target was highly correlated to the velocity of the target and developed an analytical model to estimate *MT*. Following this work, Hoffmann [14] further presented a model for *MT* prediction in moving target selection by introducing the steady-state position error which reduced the effective target width. Although the above models predict the time duration rather than the endpoint distribution of moving target selection, they inspired us that the difficulty in moving target selection could increase when the moving speed of the target increased.

In addition to moving targets, other dynamic targets such as scaling targets and temporal targets have also been studied. McGuffin and Balakrishnan [21], and Zhai et al. [36] explored the predictive performance of Fitts' law on expanding targets. They suggested that the performance was dominated by the target's final size when clicked, not the initial one.

We find few work in the HCI literature that directly studies the endpoint distribution in dynamic target selection. One work that gave insights of the usage of endpoint distribution was the Temporal Pointing [18]. The study presented a model to predict error rates in temporal pointing tasks with a temporal distance from beginning and a limited time window for selection. However, they did not model the distribution per se, letting the underlying reason of the error prone unclear.

Using close-loop system in human motor control theory to understand the process of dynamic target selection is conceptually appealing. Numbers of optimal feedback control (OFC) system were presented which provided good simulation to the reaching movement involves static and moving targets [30, 31, 24]. In spite of simulating the movement with high similarity, it is hard to directly use these models in HCI designs, due to the difficulties in tuning the parameters to achieve high-fidelity simulations.

Based on human motor control theory, studies on pursuit models [25, 26, 38] are more practical and relevant to this paper. Results in these works indicated that increasing target speed and decreasing target size can make the tracking process harder. Although pursuit models provided a good formulation for the tracing process of moving target selection, there is no selection operation (i.e. click) in such models and the typical performance measures (e.g., RMS errors and time on target) are different from our study.

Error Rate Prediction

Error rate is one of the most important factors in HCI, predicting error rate has been applied in a wide range of scenarios such as text entry and computer games [34]. Given the distance, size of the key buttons or the enemies, and the time need to be complete the typing or the shooting, knowing the errors that may happen in these specific situations can offer valuable hints on revisions to the designers, such as increasing the button size or decreasing the enemy moving speed. In which, the moving speed can only be considered through a moving model presented in this paper.

Although error rate is investigated by most studies in HCI, little work can be found on theoretically modeling the error per se. Wobbrock et al. [34] formulated a 1D predictive model for error rates based on Fitts' law parameters, their results indicated that the effect of target size on error rate is much greater than that of target distance. In latter work [35], they further extended the model into 2D condition. However, they found inconsistent results in the two studies, we argue that instead of modeling the selection error rate directly, giving a more fundamental understanding of the selection results as we did in this paper, may be a better choice.

In dynamic target selection, as mentioned, error rate prediction in temporal pointing [18] is more similar with the study we did. We share the same idea of error rate prediction that leveraging the endpoint distribution to explain the failed attempts of user. Their model shows a shorter time window lead to higher error rates, which is also very closed to our observation that faster speed and smaller size lead to more errors. However, the former is in temporal domain while the latter is in spatial, and we believe it is still very meaningful to make a deeper understanding of the endpoint distribution behind the error rate itself involves spatial moving targets.

Target Selection Techniques

In contrast to physical pointing, virtual pointing can be remarkably improved by manipulating the control-display parameters [10]. Extensive techniques have been introduced to improve the pointing performance, some of them, such as Area Cursor [17], Bubble Cursor [10], were first presented in static target scenario and then be adapted in dynamic scenario. Some were designed facing the dynamic target directly, such as Comet and Ghost [11, 13]. Our survey focuses on the techniques that are adaptable in moving target selection.

These techniques can be summarized into two major categories: pointer enhancement and target enhancement [15]. For pointer enhancement, increasing the pointer size is one of the most straightforward strategies. It can be done by extending the time window of intercepting the moving target [32], or reducing the effective target distance [15]. Another strategy is to increase the pointer speed [16]. However, this strategy also increases the pointing error rate. The cursor enhancement techniques, such as the Area Cursor [17], Bubble Cursor [10] and the Implicit Fan Cursor [29], required the computer's knowledge of the most probable target in relation to the UI layout. Besides, these techniques do not consider the effect of target velocity, yet cannot compensate for errors caused by human's motor delay when tracking the moving target.

For target enhancement, according to [15], there are two possible enhancement strategies: increasing target size and reducing target speed (usually down to zero). The first strategy was used in Expending target (or fisheye) [27, 20, 36] and Comet [11, 13] based on interface parameters such as cursor position and speed of the target. Target Lock [11] made the selection easier by eliminating the need for accurate click: as soon as the pointer moves over a target, the target is locked with the pointer and the user can click anywhere to complete selection. This technique literately makes the target size infinite. For the second strategy, pause and click is a widely used method in video systems. Hold [12] temporarily pauses the moving content to provide a static target. Proxy-based technique was used in Target Ghost [13] to creates stationary proxies of all the objects in the scene based on their position at the time the users invokes them, without disrupting object movement. Despite the effectiveness of these techniques, many of the design decisions and parameters are largely ad hoc due to the insufficient understanding of human performance in these scenarios.

PROBLEM FORMULATION

This research investigates the endpoint distribution in 1D-speed-fixed moving target acquisition tasks with a mouse cursor (Figure 2), as the first step toward understanding human performance in such scenarios. We hope this work could inspire further exploration in a wider range of scenarios, such as changing speed and 2D/3D moving target acquisition, pointing by finger tapping, etc.

In this scenario, a user controls a mouse to point a vertical ribbon target with a certain width in a full display. Before the user starts pointing, the mouse cursor keeps still at the start position. The target is moving away or moving towards the cursor at a fixed speed. The user clicks the mouse button to finish the selection process. By repeating this process, we can get a series of actual endpoints.

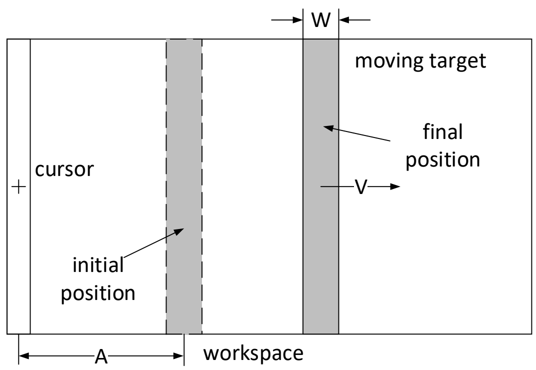

Figure 2. Abstracted task of 1D moving target selection.

The factors involved are defined as follows:

- *A*: the initial distance between the cursor and the center of the target, right before the pointing process starts
- *W*: the width of the target
- *V*: the moving velocity of the target

MODELING ENDPOINT DISTRIBUTION IN MOVING TARGET SELECTION

Hypotheses

Inspired by the findings of previous studies and our pilot tests, we had the following hypotheses on moving target acquisition.

H1: *The endpoint distribution in moving target selection is Gaussian.*

This hypothesis is based on the results of previous studies [37, 5, 6], which have shown that the endpoint distribution of static target pointing is Gaussian. In our pilot tests of moving target selection, we observed that this conclusion might also hold in moving target selection

H2: *The initial distance A does not affect the endpoint distribution.*

Previous research [37, 5, 6] has shown that the initial distance between the pointing device and the target does not influence the endpoint distribution. Instead, it is mainly affected by the process near the end of the pointing operation. This finding has also been confirmed by previous experiments on target expanding [21, 36], in which it was found that users were able to adjust their behaviors to take the full advantage of the final target size, even though the target expanded only when the cursor was approaching.

H3: *The target width (W) and the moving velocity (V) affect the endpoint distribution.*

Target width (W). Previous studies [37, 5, 6] on static target pointing demonstrated that the Gaussian parameter σ of the end point distribution is linearly proportional to the target width. They attributed this results to user's tendency of making the maximum use of the tolerance (i.e. target width) thus to save her time and effort in target selection. We believe such finding could also hold when pointing moving targets.

Moving velocity (V). Motion control theory indicates that there is time delay in the sensory-motor control system [31], which might cause the tendency of the endpoints falling behind the target. Such tendency becomes stronger when the target moves faster. This theory has also been used in Comet [11, 13] to, assist moving target selection by augmenting the moving target with a tail, the size of which is based on the speed and width of the target. Therefore, we hypothesize that the target velocity could affect endpoint distribution.

Ternary-Gaussian Model

Based on the hypotheses discussed earlier, we further propose a *Ternary-Gaussian* model to interpret the endpoint distribution in moving target selection.

Specifically, the location of the endpoints is a random variable *X* following a Gaussian distribution:

$$X \sim N(\mu, \sigma^2) \tag{1}$$

Here we take the center of the target's final location when clicked as the origin of x-axis, whose direction is the same with target movement direction.

Moreover, we also hypothesize that X is the sum of three normally distributed components:

$$X = X_a + X_m + X_s \sim N(\mu, \sigma) \tag{2}$$

Where $X_a \sim (\mu_a, \sigma_a^2)$, $X_m \sim N(\mu_m, \sigma_m^2)$ and $X_s \sim N(\mu_s, \sigma_s^2)$ correspond to the precision of the pointing device, target velocity and target width, respectively. We explain these components as follows.

The first component, X_a, reflects the absolute precision uncertainty of a motor system that includes the input device, which has also been included in the *dual-distribution hypothesis* of FFitts Law [5]. It is independent of users' desire to follow the specified task precision (e.g., target width and velocity) and cannot be controlled by a speed-accuracy tradeoff. Therefore, the distribution parameters μ_a and σ_a are two constants.

The second component, X_m, depends on the uncertainty caused by the motion of the target. We assume that the Gaussian parameters μ_m and σ_m are both proportional to the moving velocity (V).

The third component, X_s, depends on the desired precision of hitting the target and the corresponding action speed. Such variability is controlled by the speed-accuracy tradeoff of human motor systems revealed in Fitts' law [8]. We assume that the distribution parameters μ_s and σ_s are both proportional to the target width (W) as in previous works [37, 5, 6].

In our pilot tests of moving target selection, we observed the interaction effect between W and V on σ. Specifically, the positive correlation coefficient between V and σ reduced when W increased, which implied that X_m and X_s are not independent of each other. We model the relationship by setting their covariance to a term V/W.

Then, by getting the sum of the three Gaussian distributions, we have a total Gaussian distribution with parameter μ:

$$\begin{aligned} \mu &= \mu_a + \mu_m + \mu_s \\ &= a + bV + cW \end{aligned} \tag{3}$$

and parameter σ:

$$\begin{aligned} \sigma &= \sqrt{\sigma_a^2 + \sigma_m^2 + \sigma_s^2 + \mathrm{cov}(X_m, X_s)} \\ &= \sqrt{d + eV^2 + fW^2 + g\frac{V}{W}} \end{aligned} \tag{4}$$

where a, b, c, d, e, f and g are constants which can be measured via experiments.

We conducted two experiments to test the hypotheses and evaluate the *Ternary-Gaussian* model.

EXPT 1: INVESTIGATING THE EFFECT OF THE INITIAL DISTANCE

In this experiment, we investigated the effect of the initial distance (A) on endpoint distribution in moving target selection, by controlling the width (W) and velocity (V) of the target.

Participants and Apparatus

We recruited 12 subjects (6 females and 6 males, with an average age of 27) in this study. All of them are right-handed and are daily users of computer and mouse.

The experiment was conducted on a Dell OptiPlex 9020 laptop computer, with an Intel Core i7 4 Quad core CPU at 3.6 GHz and a 23-inch (533.2×312mm) LED display at 1,920×1,080 resolution. The pointing device was a Dell MS111 mouse (1000 dpi). We used Window 10's default cursor settings and a transfer function with constant CD gain of 10.44 in this study.

Design and Procedure

We leveraged a within-subjects design with four conditions corresponding to the four levels of A (192, 384, 768, 1152 pixels). Each condition included 30 trials. The orders of trials were randomized for each participant and they could have a rest between trials. It took about 10 minutes to complete the study.

In each trial, a participant clicked a "start" button to start. After a short interval (i.e. randomized from 700 - 2,000ms), the laptop played a beep sound and displayed the moving target (i.e. a vertical blue ribbon with a fixed width of 96 pixels and a fixed speed of 192 pixels/second). Participants were asked to acquire the target as quickly and accurately as they could. They could only click the mouse once per trial, regardless of whether they hit the target or not. We recorded the coordinates of all endpoints.

Results

In total, we got 48 (4 conditions x 12 participants) sets of endpoints. All of them passed the Kolmogorov-Smirnov test (alpha=0.05) for normality of the distribution, which supported **H1**. We removed the outliers (0.76% of the data) which deviated from the mean for more than three standard deviations, and estimated the actual μ and σ via maximum likelihood estimation (MLE) for each of the 48 Gaussian distributions.

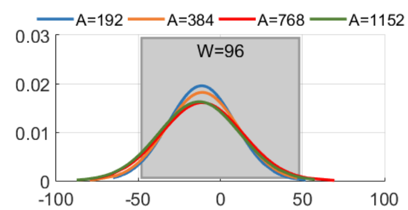

Figure 3. Endpoint distributions estimated from data for the 4 levels of A, gray area marked the target.

Repeated-measure ANOVA showed that A did neither significantly affect μ ($F_{3,9}=0.945$, $p=.911$) nor σ ($F_{3,9}=0.639$, $p=.237$). Figure 3 shows the endpoint distributions corresponding to the four levels of A (192, 384, 768, and 1152), in which the values of μ were (-11.03, -11.82, -11.08, and -12.66), and the values of σ were (19.28, 21.25, 23.99, 21.94). We observed that these distributions overlapped each other greatly. Please note that all the values of μ were negative, indicating the tendency of the endpoints falling behind the target.

In summary, we demonstrate through empirical evidence that the initial distance A did not exhibit significant effect on the endpoint distribution, which supported **H2**.

EXPT 2. EFFECTS OF TARGET SIZE AND MOVING SPEED

In this experiment, we investigated the effect of the width (W) and velocity (V) of the target on endpoint distribution in moving target selection. The participants, apparatus, and the pointing task design were the same with that of **Expt 1**. This experiment had two separate sessions, exploring the following two movement directions in 1D pointing tasks:

- Moving-away: user pursues the target
- Moving-towards: user intercepts the target

We separated the two sessions to avoid placing a high time burden on each participant and to reduce the potential impact of fatigue. The interval between the two sessions was one week and we believe it is enough to avoid the order effect.

For each session, we leveraged a within-subjects design of 16 conditions, with 4 levels of W (24, 48, 96, and 144 pixels) crossed with 4 levels of V (96, 192, 288, and 384 pixels/second). Each condition included 30 trials. The orders of trials were randomized for each participant and they could have a rest between trials. It took about 40 minutes to complete the one session of experiment.

Since we validated that the initial distance (A) did not affect the endpoint distribution, we randomly chose a value for A (between 384 and 864 pixels to the left/right boundary for moving-away/moving-towards condition of the display) in each trial during the study.

Results

All the 384 (16 conditions × 12 participants × 2 directions) sets of endpoints passed the Kolmogorov-Smirnov test (alpha=0.05) for normality of the distribution, which supported **H1**. We removed the outliers (0.88% of the data) in the same way of **Expt 1**, and estimated the actual μ and σ via MLE for each of the 384 Gaussian distributions.

Previous research indicated that the direction of target movement can result in differences in targeting strategies and motor control movements [11, 32]. As a result, we run the analysis on the datasets of the two sessions separately.

Moving-away Dataset

Figure 4 shows the endpoint distributions of the 16 conditions in moving-away dataset.

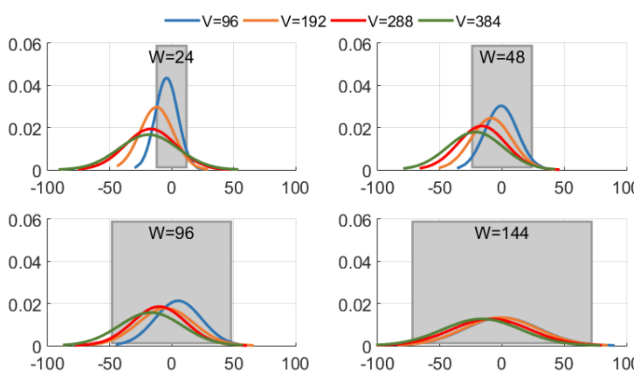

Figure 4. Endpoint distributions estimated from data in moving-away dataset.

Both V ($F_{3,9}$=*30.538, p<.001*) and W ($F_{3,9}$ =*6.502*, *p=.012)* exhibited significant effect on μ through two-way repeated-measure ANOVA. The values of μ were (0.16, -6.49, -13.10, and -17.96 pixels) corresponding to the four levels of V (Figure 5.a). Pair-wise comparisons showed significant differences across all pairs (*p<.05*). The values of μ were (-12.63, -11.83, -6.16, and -6.76 pixels) corresponding to the four levels of W (Figure 5.b). Pair-wise comparisons showed significant differences across all pairs (*p<.05*) except the following two (-12.63 vs. -11.83, *p=0.494*; -6.16 vs. -6.76, *p=.652*).

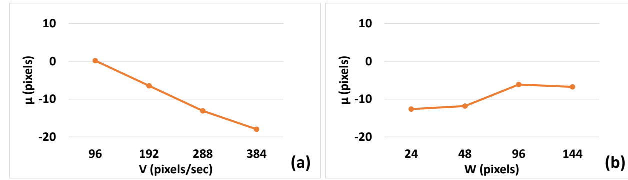

Figure 5. Average μ of subjects' endpoint distributions for 4 levels of V (left) and W (right).

Both V ($F_{3,9}$=*9.730*, *p=.003*) and W ($F_{3,9}$=*30.214, p<.001*) exhibited significant effect on σ. The values of σ were (17.32, 19.41, 21.60, and 24.88 pixels) corresponding to the four levels of V (Figure 6.a). Pair-wise comparisons showed significant differences across all pairs (*p<.05*). The values of σ were (15.64, 16.84, 21.59, and 29.15 pixels) corresponding to the four levels of W (Figure 6.b). Pair-wise comparisons showed significant differences across all pairs (*p<.05*) except the following case (15.64 vs. 16.84, *p=.209*).

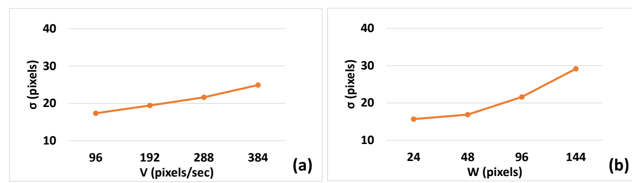

Figure 6. Average σ of subjects' endpoint distributions for 4 levels of V (left) and W (right).

The statistical analysis revealed significant interaction effects of V×W on both μ ($F_{9,3}$=*34.512*, *p=.007*) and σ ($F_{9,3}$=*4.806*, *p<.001*), which suggested that their effects on endpoint distribution were not independent. Figure 7 shows such effects.

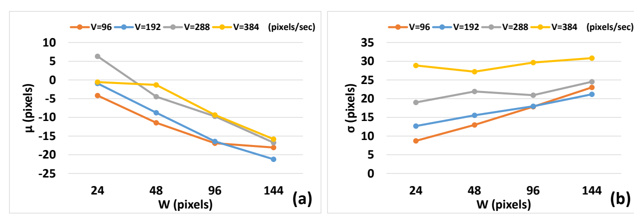

Figure 7. Interaction effect of V × W on μ (left) and σ (right).

Moving-towards Dataset

The same statistical analysis was run on the moving-towards dataset, and we obtained results that were highly consistent with moving-away condition. Therefore, we briefly summarized them as follow:

Both V ($F_{3,9}$=*27.65*, *p<.001*) and W ($F_{3,9}$ =*60.27*, *p<.001)* exhibited significant effect on μ. Both V ($F_{3,9}$=*16.701*, *p<.001*) and W ($F_{3,9}$=*23.705*, *p<.001*) exhibited significant effect on σ. The statistical analysis revealed similar significant interaction effects of V×W on both μ ($F_{9,3}$=*2.001*, *p=.047*) and σ ($F_{9,3}$=*4.43*, *p<.001*) as they did in moving-away condition.

In summary, this experiment demonstrated that 1) the endpoint distributions were Gaussian, which supported **H1**; 2) both of the target width W and the moving speed V exhibited significant effects on the endpoint distribution, which supported **H3**; 3) the interaction $V \times W$ also exhibited a significant effect on the endpoint distribution; and our hypotheses are held across the two moving directions.

FITTING THE TERNARY-GAUSSIAN MODEL

After we validated the three hypotheses, in this section, we estimated the constants of the *Ternary-Gaussian* model (a to g in Equation 3 and 4) on both of the moving-away and moving-towards situations.

Note that a Gaussian distribution is determined by two parameters (μ and σ), therefore the fitting of *Ternary-Gaussian* model includes fitting the function μ=f(V, W) and the function σ=g(V, W) to the empirical data. We have two movement directions, so we need to use the two functions to fit the two data sets generating 4 model parameters in total (i.e. *μ-away*, *σ-away*, *μ-towards* and *σ-towards*).

We used the least square regression method to estimate the constants. Overall, the models fit the empirical data well with 0.926 and 0.97 R^2 values for *μ-away* and *σ-away*, and 0.978 and 0.923 R^2 values for *μ-towards* and *σ-towards*, respectively. Table 2 shows the estimated constants in Equation 3 and 4, and the corresponding R^2 values for the regressions.

parameters	constants	R^2	mean R^2
μ-away	$a1$ = 1.3921 $b1$ = - 0.064 $c1$ = 0.0579	0.926	0.952
μ-towards	$a2$ = - 6.0301 $b2$ = - 0.0505 $c2$ = 0.1785	0.978	
σ-away	$d1$ = 1.4597 $e1$ = 0.0015 $f1$ = 0.0386 $g1$ = 19.6039	0.97	0.946
σ-towards	$d2$ = 47.3084 $e2$ = 0.0022 $f2$ = 0.0295 $g2$ = 10.908	0.923	

Table 2. The estimated constants of the model and the R^2 values for the regression.

Figure 8 shows the 3D plot of the function μ=f(V, W) for both moving away and towards conditions comparing with actual values. The points in the plot represent the actual values—blue means that they were overestimated by the model while red means underestimated. Figure 9 shows the 3D plot of the function σ=g(V, W) for both moving away and towards conditions comparing with actual values.

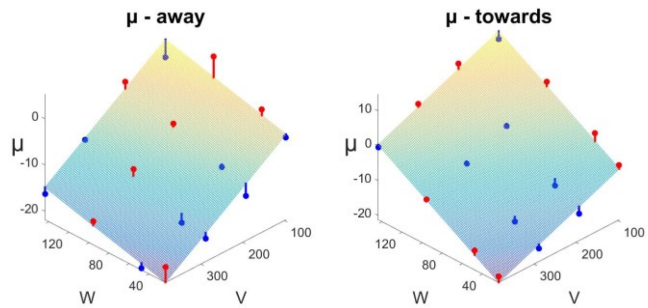

Figure 8. 3D plot of the model function μ=f(V, W) compare to the actual μ from data.

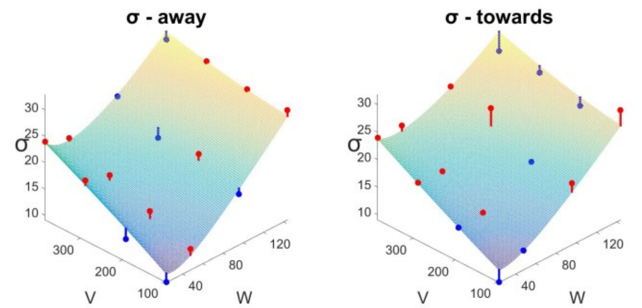

Figure 9. 3D plot of the model function σ=g(V, W) compare to the actual σ from data.

In summary, we built a *Ternary-Gaussian* model for each movement direction and found they were descriptive of the endpoint distributions in moving target selection. The models fitted the empirical data well with 0.95 and 0.94 R^2 values for μ and σ, respectively.

In the following two sections, we discuss two extensions of the model. First, we derived the mathematical function for estimating the error rates in moving target selection. Second, we present an effective interaction technique to aid the selection of moving target, where the parameters are determined based on the *Ternary-Gaussian* model.

MODEL-ASSISTED ERROR RATE PREDICTION

In target pointing tasks, error rate is defined as the percentage of failures among all the trials. For basic pointing technique, a failure means that the endpoint falls outside the target. With the *Ternary-Gaussian* model, given the width, speed, and direction of the moving target, we first calculate the corresponding μ and σ which characterize the endpoint distribution. Then we can predict the error rate via cumulative distribution functions (CDF).

The CDF of the Gaussian distribution specified by μ and σ is:

$$P(x) = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{x} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \qquad (5)$$

The following equation gives us the probability that X falls into the range of (-∞, x):

$$P(x) = \frac{1}{2}\left[1 + erf\left(\frac{x-\mu}{\sigma\sqrt{2}}\right)\right] \qquad (6)$$

where erf(x) is the error function encountered in integrating the normal distribution:

$$erf(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt \qquad (7)$$

According to the definition, the error rate is the probability that X falls out of the range (x_0, x_1), where x_0 and x_1 represent the left and right boundaries of the target:

$$\begin{aligned} ER(\mu,\sigma) &= 1-[P(x_1)-P(x_0)] \\ &= 1-\frac{1}{2}[erf(\frac{x_1-\mu}{\sigma\sqrt{2}})-erf(\frac{x_0-\mu}{\sigma\sqrt{2}})] \end{aligned} \qquad (8)$$

where μ and σ are defined by our model. Figure 10 illustrates the implication of this formulation.

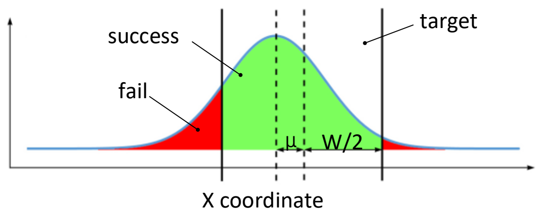

Figure 10. Error rate is computed by integrating the distribution within the range outside the target boundary.

We denote Equation 8 as the *Error Model* founded on the *Ternary-Gaussian* model. We then used the data in previous section (including both moving directions) to evaluate the goodness of fit and the generalizability of the *Error Model*.

We found that the *Error Model* fitted the data well with 0.974 and 0.966 R^2 values for moving-away condition and moving-towards condition, respectively. The fitting results of all 16 conditions in each moving direction are displayed in Figure 11.

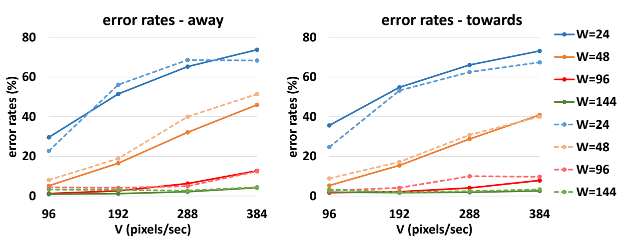

Figure 11. Actual (dashed lines) and predicted (plain lines) error rates for 32 conditions.

We further performed a repeated two-fold cross-validation to test the generalizability of the *Error Model*. The model parameters were obtained over the data of 6 randomly chosen subjects and tested on the rest 6. We use mean absolute error (MAE) between model prediction and the actual values from other 6 subjects as the fitness score. Over 100 iterations, we obtain average MAE of 4.7% (SD=1.6) in moving away direction and 5.8% (SD=2.3) in moving towards direction.

MODEL-ASSISTED TARGET SELECTION

In this section, we present *BayesPointer*, an interaction technique to aid moving target selection in an *implicit* manner. Here the word *implicit* means that this approach does not modify the appearance of existing interface. *BayesPointer* integrates the *Ternary-Gaussian* model into Bayes' rule to determine the intended target, rather than relying on the physical boundaries. The decision-making strategy of *BayesPointer* is formulated as follows:

Let $T=\{t_1,t_2,\ldots,t_n\}$ be the n moving targets. Given a endpoint s, the conditional probability that t ($t \in T$) is the intended target is $P(t|s)$. Determining the intended target is equivalent to finding $t*$ that maximizes $P(t|s)$. By using Bayes' rule, we calculate $P(t|s)$ as:

$$P(t \mid s) = \frac{P(s \mid t)P(t)}{P(s)} \qquad (9)$$

where $P(t)$ denotes the prior probability of selecting t without the observation of s, which is set $1/N$ (N is the total number of targets), assuming that each target has an equal prior selection chance; $P(s|t)$ is the likelihood function which expresses how probable the endpoint s is intended to select the target t. The meaning of $(s|t)$ is consist with the probability density function of endpoint distribution; $P(s)$ is the normalization constant that holds the same across each target.

As a result, the intended target $t*$ can be chosen as follow:

$$t* = \arg\max_t (P(s \mid t)) \qquad (10)$$

Noted that each target t has a specific V and W. Thus, we can calculate the corresponding μ and σ of its distribution. Then we have:

$$\begin{aligned} t* &= \arg\max_t (p(x \mid \mu_t, \sigma_t)) \\ &= \arg\max_t (\frac{1}{\sqrt{2\sigma_t^2\pi}} e^{-\frac{(x-\mu_t)^2}{2\sigma_t^2}}) \end{aligned} \qquad (11)$$

which is the decision-making strategy of *BayesPointer* to determine the intended target.

To avoid the situation that *BayesPointer* always return a intended target even when user intentionally click on a blank space, a target can only be returned when the click falls into the [-3σ, 3σ] range to the center of the corresponding distribution.

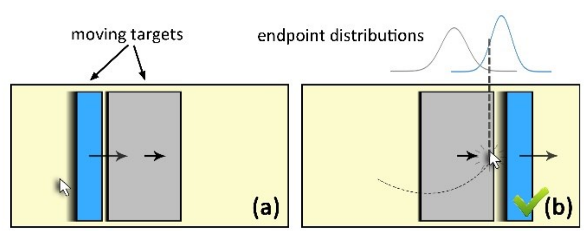

Figure 12. The working process of *BayesPointer*. (a) The two moving targets with different speed and size; (b) the blue one is determined as the intended one with our *BayesPointer*.

Figure 12 illustrates how *BayesPointer* works in practice. Two targets with different moving speeds and sizes appear in the workspace. Suppose that the user is trying to select the blue target. Due to motor delay, his/her click falls behind the target and hits the gray one. While the basic selection technique treats the gray one as the intended target in this case, *BayesPointer* identifies the blue one as the true target instead, given its higher probability calculated in the likelihood function (i.e. endpoint distributions). We evaluate this technique in the following section.

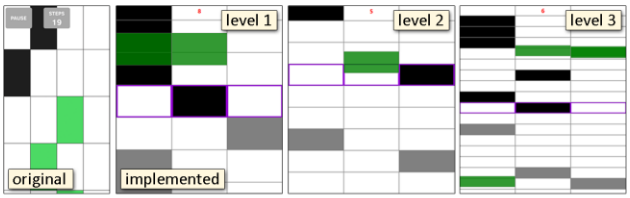

Figure 13. The original design (left) of *Don't Touch The White Tile* and the three variants in our study (right three).

EXPT 3. USING THE MODEL IN A GAME INTERFACE DESIGN

We conducted this study to explore the feasibility of using the *Ternary-Gaussian* model to predict error rates and to assist moving target selection in real-world applications. We used a popular game named *Don't Touch The White Tile*[1] (Figure 13) as the testbed. In the game, a player's goal is to hit the black tiles as many times as possible while not hitting the white tiles. The game involves selecting targets moving in one direction, which matches our model quite well. At the same time, additional interferences exist in this scenario, including but not limited to the highly required task switch and visual search ability of players, where they need to make continuous selections back-to-back in order to prevent the tiles from hitting the ground. In the following of the paper, we show that our model and techniques can also perform well in this relatively complex situation.

Implementation

In the original game, white and black tiles were randomly grouped in a mosaic style dropping continuously from the top to the bottom. Players had to tap the black tile in the lowest row while not hitting white ones. The game ended when the player missed the target or hit a wrong tile. During the gameplay, the moving speed of the tiles constantly increased until the player finally made a mistake. The final score was the number of black tiles that the player hit.

With the same rule, we implemented a PC version of the game with some minor modifications (Figure 13). Our implementation gave each player 5 lives, which allowed them to make some mistakes. Such modification allowed us to collect more endpoints in a row. We also added hints (i.e. thick purple border) to the target row so that players could find it more easily. To make the game more challenging, interference targets in green color moved from the bottom to the top irregularly. In addition, to test our model in conditions of varying target sizes, three game levels with decreasing target heights were designed. Theoretically, smaller target height could make the game more challenging.

We created two versions integrated with different selection techniques: *Basic* and *BayesPointer*. The former was the basic selection technique in the Windows operating system and served as the baseline, and the latter used Bayes' rule to determine the intended target.

Note that the game involved two-dimensional targets, and the targets move in vertical direction. Thus we have to adapt our technique by: 1) in horizontal axis, to select a target, the click point must fall inside the target's horizontal boundaries; 2) in vertical axis we use *BayesPointer* to select the target. In addition, the moving-towards parameters were used because the tiles in this game were always moving towards the cursor.

Methods

Subjects

Twelve subjects (6 females and 6 males, average age 24.5) were recruited to participate in the experiment. All of them were right-handed and were daily users of computers and mouse. None of them were in **Expt 1** or **2**. We used the same apparatus with previous experiments.

Design

We leveraged a within-subjects design and made comparisons between the two selection techniques: *Basic* and *BayesPointer*.

All tiles in the game were 230 pixels width, with 135 pixels, 90 pixels, and 45 pixels heights in level 1 to 3, respectively. The moving speed of the tiles increased with fixed increment in each level. The highest speeds were 1312, 875, 437 pixels/second, respectively. We chose lower speed setups in level 2 and 3 to prevent the game becoming too difficult.

Each of the 12 subjects had to complete 3 gameplay sessions with each technique. In total, we had 3×2×12=72 trials. Participants were allowed to rest between trials and between game levels. Each trial took about 2 minutes, and it took about 12 minutes in total for each participant to finish the test. Participants practiced with each technique before starting the formal study. The order of techniques was counterbalanced across participants.

Procedure

For each trial, the participant clicked a button to start the game. Once the game started, participants should keep playing until they lost all the 5 lives. All endpoints, no matter succeeded or failed, were recorded. The target row turned into gray if the participant correctly clicked the black target, and if the participant clicked a wrong target, it blinked in red for 0.5 second as a hint.

Measures

In this experiment, we collected error rates for all the velocity (V) × height (H) combinations and the total score of each

[1] https://itunes.apple.com/cn/app/id866148386?m

level. We asked participants to fill out a post-survey about the perceived pros and cons of each technique.

Results and Discussion

Predicting Error Rate

The click data of *Basic* technique was used to test the performance of error rate prediction. Because of the five-mistake-and-die setup of the game, very few (n<=4) players could survive after speed 4, leading the data in this study followed a relatively long tail distribution, with 85.4% fell into speed 1-4 and 14.6% fell into higher speed levels. Therefore, we only used the data in speed 1-4, since the reset of them were not able to represent the general performance of humans.

The MAE of error rate prediction in the 12 $V \times H$ conditions is 2.7%, which mean the model over/under estimated an average 2.7% errors for each condition. Figure 13 shows the actual and predicted error rates for all conditions.

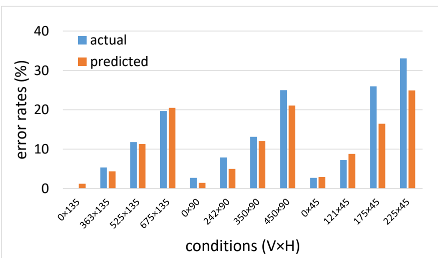

Figure 14. Actual and predicted error rates for 12 conditions.

As shown in Figure 14, the error rate increases when the speed increases and when the target size decreases. This trend has been well predicted by the model, but the error rates with faster speed, especially in the last two conditions in level 3, were underestimated. We attribute this to the demand on the ability of task switch and visual search, which introduces additional challenges in the game, especially in extreme conditions that moving speed was high and target size was small (e.g., last 2 conditions in Figure 14). As a result, the actual error rates were higher than predicted in such conditions.

Assisting the Selection of Moving Target

Statistical analysis showed a significant different between the two techniques on the final scores ($F_{1,11}=22.215$, $p=.0006$). With *BayesPointer*, participants had an average scores of 34.94, 31.63 and 25.69 in level 1 to level 3, respectively. The increases were 33.6%, 43.0% and 39.9% compared with the *Basic* technique (Figure 15 (a)). Results showed a significant different between the two techniques on error rate ($F_{1,11}=36.819$, $p<.001$). *BayesPointer* exhibited lower error rates, the level 1 had the lowest error rate (0.153), followed by level 2 (0.159) and level 3 (0.181), which were 25.7%, 24.8% and 23.4% lower than the *Basic* technique (Figure 15 (b)).

Subjective feedback

A 7-points Likert scale for rating the techniques according to preference and accuracy showed, that participants like using *BayesPointer* (M=5.16, SD=0.93) more than using *Basic* (M=4.08, SD=0.66), and they thought that using *BayesPointer* (M=5.33, SD=1.17) got higher accuracy than using *Basic* (M=4.16, SD=0.89) as well.

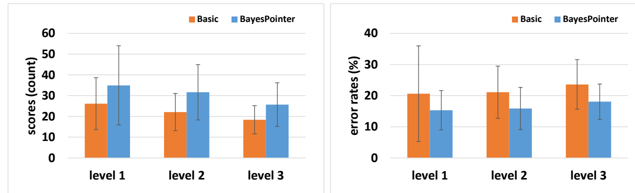

Figure 15. Average scores and error rates of the two techniques in 3 game levels.

CONCLUSIONS AND FUTURE WORK

In this paper, we reported our work in understanding and modeling the endpoint distribution in 1D unidirectional moving targeting selection. We proposed a *Ternary-Gaussian* model to interpret the distribution of the endpoints for targets moving in two horizontal directions. Results show that our model fit the data very well. We also demonstrated how our model can be used to predict error rates and assist selection of moving targets. We observed good performance when applying these two means in a game interface design.

As one of the first attempts to model human behavior uncertainty in moving target selection, our work provides the HCI community with a theoretical foundation and empirical evidence for future research and design in such scenarios. For instance, when designing the difficulty curve of a game which the main challenge is to select moving targets, our model can be used to predict error and guide game designers to modify the game parameters such as healthy points, size and speed of the enemies. Our model is not limited to gaming, in other animation systems (e.g., simulation systems, video monitoring system), our model can also be used to aid target selection on these systems in an implicit manner.

In the future, we are interested in extending our research in several directions. First, we will examine whether our model can be transferred into user interfaces with interaction styles other than mouse (e.g., touch-based, pen-based, gesture-based). Second, we will also pursue modeling uncertainty in selecting moving targets with changing speed and in 2D/3D space. Furthermore, we will explore how our model can help to improve interaction efficiency in general user interfaces and compare it with other state-of-the-art pointing techniques (e.g., Comet [11, 13] or Bubble Cursor [10]).

ACKNOWLEDGMENTS

We thank Dongqi Han and Nianlong Li for help with software implementation, the members of IEL ISCAS, and the reviewers for their constructive feedback. This work was funded by National Key R&D Program of China (Grant No. 2016YFB1001405), and National Natural Science Foundation of China (Grant No. 61232013 and 61422212), and Key Research Program of Frontier Sciences, CAS (Grant No. QYZDY-SSW-JSC041), and partially supported by CAS Pioneer Hundred Talents Program.

REFERENCES

1. Johnny Accot and Shumin Zhai. 1997. Beyond Fitts' law: models for trajectory-based HCI tasks. In *Proceedings of the ACM SIGCHI Conference on Human factors in computing systems* (CHI '97), 295-302. http://dx.doi.org/10.1145/258549.258760
2. Johnny Accot and Shumin Zhai. 1999. Performance evaluation of input devices in trajectory-based tasks: an application of the steering law. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems* (CHI '99), 466-472. http://dx.doi.org/10.1145/302979.303133
3. Johnny Accot and Shumin Zhai. 2003. Refining Fitts' law models for bivariate pointing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '03), 193-200. http://dx.doi.org/10.1145/642611.642646
4. Johnny Accot and Shumin Zhai. 2002. More than dotting the i's ---foundations for crossing-based interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '02), 73–80. http://dx.doi.org/10.1145/503376.503390
5. Xiaojun Bi, Yang Li and Shumin Zhai. 2013. FFitts law: modeling finger touch with fitts' law. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '13), 1363-1372. http://dx.doi.org/10.1145/2470654.2466180
6. Xiaojun Bi and Shumin Zhai. 2013. Bayesian touch: a statistical criterion of target selection with finger touch. In *Proceedings of the 26th annual ACM symposium on User interface software and technology* (UIST '13), 51-60. http://dx.doi.org/10.1145/2501988.2502058
7. Xiang Cao, Jacky Jie Li and Balakrishnan Ravin. 2008. Peephole pointing: modeling acquisition of dynamically revealed targets. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '08), 1699-1708. http://dx.doi.org/10.1145/1357054.1357320
8. Paul M Fitts. 1954. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of experimental psychology*, 47, 6 (Jun 1954), 381-391. http://dx.doi.org/10.1037/h0055392
9. Tovi Grossman and Ravin Balakrishnan. 2004. Pointing at trivariate targets in 3D environments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '04), 447-454. http://dx.doi.org/10.1145/985692.985749
10. Tovi Grossman and Ravin Balakrishnan. 2005. The bubble cursor: enhancing target acquisition by dynamic resizing of the cursor's activation area. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '05), 281-290. http://dx.doi.org/10.1145/1054972.1055012
11. Tyler J. Gunn, Pourang Irani and John Anderson. 2009. An evaluation of techniques for selecting moving targets. In *CHI '09 Extended Abstracts on Human Factors in Computing Systems* (CHI EA '09), 3329-3334. http://dx.doi.org/10.1145/1520340.1520481
12. Abir Al Hajri, Sidney Fels, Gregor Miller and Michael Ilich. 2011. Moving target selection in 2D graphical user interfaces. In *Proceedings of the International Conference on Human Computer Interaction* (INTERACT '11), 141-161. http://dx.doi.org/10.1007/978-3-642-23771-3_12
13. Khalad Hasan, Tovi Grossman and Pourang Irani. 2011. Comet and target ghost: techniques for selecting moving targets. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '11), 839-848. http://dx.doi.org/10.1145/1978942.1979065
14. Errol R. Hoffmann. 1991. Capture of moving targets: a modification of Fitts' Law. *Ergonomics*, 34, 2, 211-220. http://dx.doi.org/10.1080/00140139108967307
15. Michael Victor Ilich. 2009. *Moving target selection in interactive video*. Master's thesis, (December 2009) University of British Columbia, Vancouver, BC.
16. Richard J. Jagacinski, Daniel W. Repperger, Sharon L. Ward and Martin S. Moran. 1980. A test of Fitts' law with moving targets. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 22, 2 (April 1980), 225-233. http://dx.doi.org/10.1177/001872088002200211
17. Paul Kabbash and William Buxton. 1995. The "prince" technique: Fitts' law and selection using area cursors. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '95), 273-279. http://dx.doi.org/10.1145/223904.223939
18. Byungjoo Lee and Antti Oulasvirta. 2016. Modelling Error Rates in Temporal Pointing. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (CHI '16), 1857-1868. http://dx.doi.org/10.1145/2858036.2858143
19. I. Scott MacKenzie. 1992. Fitts' law as a research and design tool in human-computer interaction. *Human-Computer Interaction*, 7, 1 (March 1992), 91-139. http://dx.doi.org/10.1207/s15327051hci0701_3
20. Scott MacKenzie and William Buxton. 1992. Extending Fitts' law to two-dimensional tasks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '92), 219-226. http://dx.doi.org/10.1145/142750.142794
21. Michael McGuffin and Ravin Balakrishnan. 2002. Acquisition of expanding targets. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '02), 57-64. http://dx.doi.org/10.1145/503376.503388

22. Darius Miniotas. 2000. Application of Fitts' law to eye gaze interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '02), 339-340. http://dx.doi.org/10.1145/633292.633496
23. Atsuo Murata and Hirokazu Iwase. 2001. Extending Fitts' law to a three-dimensional pointing task. *Human Movement Science*, 20, 6 (December 2001), 791-805. http://dx.doi.org/10.1016/S0167-9457(01)00058-6
24. Qian Ning, Yu Jiang, Zhongping Jiang and Pietro Mazzoni. 2013. Movement duration, fitts's law, and an infinite-horizon optimal feedback control model for biological motor systems. *Neural Computation*. 25, 3 (March 2013), 697-724. http://dx.doi.org/10.1162/NECO_a_00410
25. Andriy Pavlovych, Wolfgang Stuerzlinger. 2011. Target following performance in the presence of latency, jitter, and signal dropouts. In *Proceedings of Graphics Interface* 2011 (GI '11), 33-40.
26. E. C. Poulton. 1974. *Tracking skill and manual control*. New York: Academic Press.
27. Manojit Sarkar, and Marc H. Brown. Graphical fisheye views of graphs. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '92), 83-91. http://dx.doi.org/10.1145/142750.142763
28. Reza Shadmehr, Maurice A. Smith and John W. Krakauer. 2010. Error correction, sensory prediction, and adaptation in motor control. *Annual Reviews of Neuroscience* 33, 1 (March 2010), 89–108. http://dx.doi.org/10.1146/annurev-neuro-060909-153135
29. Xiaojun Su, Oscar Kin-Chung Au and Rynson W.H. Lau. 2014. The implicit fan cursor: a velocity dependent area cursor. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '14), 753-762. http://dx.doi.org/10.1145/2556288.2557095
30. Emanuel Todorov and Michael I. Jordan. 2002. Optimal feedback control as a theory of motor coordination. *Nature Neuroscience.* 5, 11 (November 2002), 1226-1235. http://dx.doi.org/10.1038/nn963
31. Emanuel Todorov. 2005. Stochastic optimal control and estimation methods adapted to the noise characteristics of the sensorimotor system. *Neural Computation.* 17, 5, (May 2005), 1084-1108. http://dx.doi.org/10.1162/0899766053491887
32. James R. Tresilian. 2005. Hitting a moving target: Perception and action in the timing of rapid interceptions. *Perception & Psychophysics*, 67, 1 (January 2005), 129-149. http://dx.doi.org/10.3758/BF03195017
33. A. T. Welford. 1968. *Fundamentals of Skill*. Methuen.
34. Jacob O. Wobbrock, Edward Cutrell, Susumu Harada and I. Scott MacKenzie. 2008. An error model for pointing based on Fitts' law. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '08), 1613-1622. http://dx.doi.org/10.1145/1357054.1357306
35. Jacob O Wobbrock, Alex Jansen, and Kristen Shinohara. 2011. Modeling and predicting pointing errors in two dimensions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '11), 1653-1656. http://dx.doi.org/10.1145/1978942.1979183
36. Shumin Zhai, Stéphane Conversy, Michel Beaudouin-Lafon and Yves Guiard. 2003. Human on-line response to target expansion. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '03), 177-184. http://dx.doi.org/10.1145/642611.642644
37. Shumin Zhai, Jing Kong and Xiangshi Ren. 2004. Speed–accuracy tradeoff in Fitts' law tasks—on the equivalency of actual and nominal pointing precision. *International Journal of Human-Computer Studies*, 61, 6 (December 2004), 823-856. http://dx.doi.org/10.1016/j.ijhcs.2004.09.007
38. Shumin Zhai, Paul Milgram, Anu Rastogi. 1997. Anisotropic human performance in six degree-of-freedom tracking: an evaluation of three-dimensional display and control interfaces. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans* 27(4): 518-528. http://dx.doi.org/10.1109/3468.594917