Modeling the Uncertainty in 2D Moving Target Selection

Jin Huang huangjin@iscas.ac.cn
Feng Tian* tianfeng@iscas.ac.cn
Nianlong Li linianlong16@mails.ucas.ac.cn
Xiangmin Fan xiangmin@iscas.ac.cn
State Key Laboratory of Computer Science, ISCAS, Beijing, China
Beijing Key Lab of Human-Computer Interaction, ISCAS, Beijing, China
University of Chinese Academy of Sciences, Beijing, China

ABSTRACT
Understanding the selection uncertainty of moving targets is a fundamental research problem in HCI. However, the only few works in this domain mainly focus on selecting 1D moving targets with certain input devices, where the model generalizability has not been extensively investigated. In this paper, we propose a *2D Ternary-Gaussian* model to describe the selection uncertainty manifested in endpoint distribution for moving target selection. We explore and compare two candidate methods to generalize the problem space from 1D to 2D tasks, and evaluate their performances with three input modalities including mouse, stylus, and finger touch. By applying the proposed model in assisting target selection, we achieved up to 4% improvement in pointing speed and 41% in pointing accuracy compared with two state-of-the-art selection technologies. In addition, when we tested our model to predict pointing errors in a realistic user interface, we observed high fit of 0.94 R^2.

Author Keywords
Moving Target Selection; Endpoint Distribution; Device Factors; Error Rate; Pointing Accuracy

ACM Classification Keywords
• Human-centered computing~User models • Human-centered computing~HCI theory, concepts and models

INTRODUCTION
Moving target acquisition is one of the most fundamental interaction tasks in modern user interfaces. Understanding the selection uncertainty in such tasks may be helpful to the design of interactive systems with dynamic contents, such as video games, traffic control displays and video surveillance systems. However, this aspect is just getting started to attract attention of the HCI community [22, 28]. As the first step towards understanding human performance in such tasks, these works were conducted only in interaction scenarios where targets were moving unidirectionally, with certain input devices (e.g., mouse). However, today's interfaces usually involve pointing targets that are moving in 2D space [5]. We cannot transfer and generalize the findings and results derived from 1D experiments to 2D tasks for granted since it introduced more complex factors. In addition, these models only have been validated on certain input modalities such as a mouse. There are few generalizable findings that can provide empirical support for the application of these models to other input modalities, which prevents the adoption of such models in a wider range of interfaces.

This study aims to address the existing challenges by proposing a 2D model to be descriptive of the endpoint distribution in 2D spaces with three different input modalities. Nevertheless, modeling the endpoint distribution of moving target selection in 2D space is not a trivial task. There are at least three challenges involved. First, extending from 1D to 2D tasks brings higher degrees of freedom in target shape, position, and moving direction; thus, the complexity of the problem space is greatly increased. Second, there are multiple options in determining the coordinate system to contextualize the endpoints of moving targets, but their pros and cons have not been well studied. Third, the ideal model should be robust across heterogeneous input modalities and provide good generalizability.

We propose a *2D Ternary-Gaussian* model to be descriptive of the endpoint distribution in moving target selection. Our model relies on a *velocity coordinate system* to express the endpoint distribution based on evidences of human perception research [1, 36, 7, 2, 24, 8, 33]. This coordinate system largely reduces the complexity of the task by taking the *tangent direction* of and the *normal direction* target velocity as x and y axes, and target center as origin. To fully understand the mechanism of selection uncertainty for the task, we explore two candidate models under the 2D *Ternary-Gaussian* framework by evaluating their performances in three input modalities including mouse, stylus and finger touch. The model that considers the influence of temporal constraints in the *normal direction* shows superior performance than the one does not. By applying the winner model into two interaction scenarios, we found that our approaches significantly enhance the selection efficiency compared with two state-of-the-art selection technologies (e.g. up to 4% selection speed and 41% selection accuracy) and precisely model the selection error rates in a realistic user interface (e.g. 0.94 R^2).

This study contributes a set of findings that are inspirational for future research in 2D moving target selection:

1) In the *tangent direction*, the relationship between target size, target velocity and selection uncertainty are consistent with previous studies. [10, 11, 22].

UIST '19, October 20–23, 2019, New Orleans, LA, USA.

ACM ISBN 978-1-4503-6816-2/19/10…$15.00.
https://doi.org/10.1145/3332165.3347880

2) In the *normal direction*, the mean of endpoints is close to target center, and the standard deviation of endpoints is related to target speed although there is no velocity component in this direction.

3) The effects of target speed on selection uncertainty may related to two independent components: the inaccuracy of hand's rapid motion and the delay of human sensory-motor system.

4) In large-scale display, mouse with a constant control-display (CD) gain has higher selection uncertainty than the finger touch and stylus.

RELATED WORK

In this section, we reviewed the theories and models related to user behaviors in moving target selection, which informed the construction of the model. We summarized these existing works into three aspects: user performance in target selection, human perception of moving targets, and device factors in user performance model.

User Performance in Target Selection

One of the most famous rules governing user performance in pointing tasks is the rule of *speed-accuracy tradeoff*. In general, the more accurate the task to be accomplished, the longer it takes and vice versa. It was first reveal by Fitts' law [14], and then quantitatively evaluated in different instruction conditions [31], layers of pointing precision [45], tasks [43, 22, 23] and devices [10, 37].

In static target selection, it is widely accepted that the speed-accuracy tradeoff in pointing is imposed by the task parameters through Fitts' index of difficulty (*ID*) [45]. Based on this, researchers developed numerous user performance models including movement time (*MT)* prediction in 2D [5, 40] and 3D [18, 30] pointing tasks, trajectory-based tasks [3, 4] and crossing-based tasks [6]. The rule of speed-accuracy tradeoff also motivated researchers on building models for predicting error rate, such as the error models proposed by Wobbrock et al. based on Fitts' law parameters [39, 40]. Their models can precisely predict the error rates of pointing 1D and 2D targets. In addition, the models for describing the selection uncertainty are also governed by the rule of speed-accuracy tradeoff. These models were used as tools for studying the phenomenon of speed-accuracy tradeoff [45], adjusting Fitts' law performance [10] and assisting selection in small screens [11].

In moving target selection, the effect of initial distance between the cursor and the target becomes much smaller and usually negligible for total movement time [26] and pointing accuracy [22] in position control system such as mouse and stylus. However, the rule of speed-accuracy tradeoff holds since the user should move faster to hit the target moving with faster speed which increases the selection time and decreases the pointing accuracy. Jagacinski et al. [26] found that *MT* of moving target selection is highly correlated to target speed and developed an analytical model to estimate *MT* for such tasks. Following this work, Hoffmann [20] further presented a model for *MT* prediction in moving target selection by introducing the steady-state position error. By introducing temporal pointing [28], researchers presented a model to predict error rates for selecting moving targets in temporal domain. By integrating the perception process of visual cues into this model, its performance was improved in visual moving targets [29]. However, the temporal pointing model is limited in considering the timing of hitting an approaching target solely, and the motion uncertainty of moving the cursor to intercept the target was omitted.

Recently, Huang et al. [22] proposed a *Ternary-Gaussian* model by combining the movement uncertainty caused by motion and size of the target. They provided empirical evidences that the *Ternary-Gaussian* model precisely predicted the endpoint distribution in moving target selection, and demonstrated how the model can be used to predict error rate of moving targets and assist moving target selection. Nonetheless, the *Ternary-Gaussian* model is limited for 1D pointing tasks and no study has been conducted to validate its performance across different pointing devices. In comparison, we validated our 2D model with different input modalities to provide a wider coverage in terms of application scenarios in HCI.

Human Perception of Moving Target

There are many studies investigating human perception of moving targets in the psychology community [27, 17, 36, 32, 15], we found the following ones were highly in line with our research. Smeets et al. showed that when human tried to point on moving targets, they perceived target speed and moving direction independently [1]. They confirmed the independency in an experimental setting which is very similar to ours. In that study, participants were required to click the moving target towards specific direction at a constant speed on the display with an organic glass rod. Results in this experiment showed that the perceived direction of motion is treated differently than the perceived speed, which suggested that the motion of a moving target cannot be broken down into speed components in different directions. This conclusion provided us important foundations for building the coordinate system to express the endpoint distribution in 2D moving target selection.

Evidences showed that participants tended to move their hands more quickly towards faster targets than slower ones [24, 25]. Together with the observed positive correlation of endpoint error to speed of hand following the rule of speed-accuracy tradeoff [45, 14], we assume that the selection uncertainty is proportional to the target speed. Evidences showed the delay of human sensory-motor system affects various motion control and coordination [7, 28, 16, 42, 13]. In addition, visual perception studies showed that participants tend to point the moving target on a location on the expected moving trajectory ahead of the target at the estimated time [7, 2, 24].

As a result, we consider that there is a shift effects of endpoint along the target's moving path, which is caused by the delay of human sensory-motor system when performing the acquiring actions.

Device Factors in User Performance Model

As one of the quantitative foundations for human-computer interaction research, Fitts' law showed outstanding robustness across devices so that it has been used as a tool for computer input device evaluation [38, 37], where the index of performance (IP) can measure human performance in certain context. Such robustness can be partially attributed to applying the terms of intercept and slope in the Fitts' equation to reflect the device factors, which makes the term of index of difficulty maintain permanently to reveal the central regularity between human capacity and task parameters.

Besides pointing tasks, trajectory-based tasks such as drawing, writing, and navigation are also common. The Steering law [3], developed by Accot and Zhai, has been proved to be another robust human behavior law predicting the *MT* in such tasks. Accot and Zhai themselves firstly introduced the steering paradigm to evaluate mouse, pen tablet, trackpoint, touchpad and trackball [4], and then optimized design parameters in mouse and touch screen [34]. Both works found that steering law fitted well in different devices and task parameters, while the latter further presented that steering difficulty can be used as a criterion to choose which device should be adopted and what parameters are optimized with a specified application.

Therefore, beyond accuracy, the capability of being robust between different input devices is another important factor that we consider in establishing the model. If the performance of our model cannot hold between input modalities, the significance of this piece of work can be reduced. This disobeys the original vision of developing the model, which is finding the general underlying regularities in moving target selection to improve user experience and guide future interface design.

PROBLEM FORMULATION

As shown in **Figure 1**, in the task of 2D moving target selection, a user controls a pointer to acquire a circular target. Before starting, the pointer keeps still at the start position. The target is initialized on a certain distance from the start position, and then moved at a fixed speed. Users controls the pointer to point on the target to finish the selection process. The involved factors included:

- Initial distance (*A*): the initial distance between the pointer and the center of the target.
- Target size (*W*): the diameter of the circular target.
- Target speed (*V*): the speed or the velocity magnitude of the target.

In practice, the task could involve more complex factors such as shape, moving direction and initial azimuth from the pointer. If these factors change dynamically during the task, it yields innumerable moving trajectories. To simplify the problem space, we formulated the task using a circular target with fixed sizes, speeds. The target is initialized at random position on a circle with a certain radius, and then moves toward a random direction with a specified speed. Thus, the problem that we tried to solve in this study is to build a model to be descriptive of the endpoint distribution for 2D moving target selection tasks with specified task parameters *A*, *W* and *V*. To establish such a model, we must first find an appropriate coordinate system which is convenient and rational, and then identify each of the possible models under this coordinate system. Finally, the model should be robust across input modalities and provide good generalizability in application scenarios.

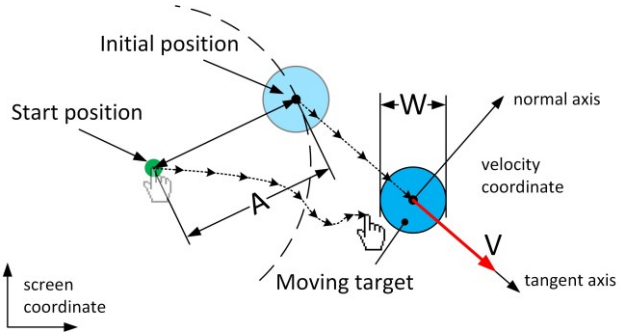

Figure 1. The task of 2D moving target selection.

MODELING THE ENDPOINT DISTRIBUTION IN 2D SAPCE

A recent work known as *Ternary-Gaussian* model [22] showed high performance on describing the endpoint distribution in 1D moving target selection.

In comparison, we presented a 2D *Ternary-Gaussian* model which adopted the central idea and some of the design decisions of the 1D model. We accomplished this by introducing a *velocity coordinate system* and, exploring two candidate models under the 2D *Ternary-Gaussian* framework. The first one only considered spatial constraints in the axis normal to velocity direction, and the second one considered both spatial and temporal constraints. We first introduced the *velocity coordinate system* in the following section.

Velocity Coordinate System

The *velocity coordinate system* is a target's local coordinate with x axis tangent to the target's moving direction and y axis normal to the target's moving direction. The origin of the coordinate system is set on the center of the target's final location when selection process complete, see **Figure 1**. This coordinate system provides us two major benefits: 1) it facilitates the description of common user performances, such as error rate, which can only be calculated from relative endpoint location; 2) it reduces the complexity of the problem and conforms to the nature of pointing movement as explained below.

For setting the x and y axes of the coordinate system, one choice is making the two axes parallel to screen's boundaries. Based on this setting, we can further decompose the target

velocity (including speed and direction) into two axes. However, there is evidence that the human perception of motion of an object cannot be broken down into speed components in different directions, but speed and moving direction are perceived and used separately [1]. Moreover, the decomposition will greatly increase the complexity of the model as we need to use an additional task parameter θ to calculate the two velocity components.

Therefore, we chose to set the x axis parallel to the target's moving direction while y axis normal to the target's moving direction, and named the x direction as *tangent axis*, y direction as *normal axis*.

2D Ternary-Gaussian Model

The prior *Ternary-Gaussian* model [22] showed that the endpoints on moving target selection follow a Gaussian distribution, and provides evidence that the initial distance (A) does not affect the endpoint distribution, while the target width (W) and the moving velocity (V) affect the endpoint distribution.

Inspired by these evidences, we formulated the location of the endpoints in the *velocity coordinate system* as a two-dimensional random variable X following a 2D Gaussian distribution:

$$X \sim N(\mu, \Sigma). \tag{1}$$

We assumed that the random variable X in the *tangent* and *normal* axes are independent (jointly normally distributed) similar with FFitts Law [10] with μ and $\sum$ as follow:

$$\mu=\begin{pmatrix} \mu_t \\ \mu_n \end{pmatrix} \text{ and, } \quad \Sigma=\begin{pmatrix} \sigma_t^2 & 0 \\ 0 & \sigma_n^2 \end{pmatrix}, \tag{2}$$

where, μ_t, σ_t, μ_n and σ_n represent the means and standard deviation of endpoint distributions in *tangent axis* and *normal axis*, respectively.

X can be viewed as the sum of three normally distributed components:

$$X = X_a + X_m + X_s \sim N(\mu, \Sigma), \tag{3}$$

where $X_a \sim N(\mu_a, \Sigma_a)$, $X_m \sim N(\mu_m, \Sigma_m)$ and $X_s \sim N(\mu_s, \sum_s)$ correspond to the **a**bsolute precision of the pointing device, the **m**otion and the **s**ize of the target, respectively.

According to the *velocity coordinate system*, the velocity component in *tangent axis* is equal to the target speed ($V_t = V$), while the velocity component in *normal axis* is zero ($V_n = 0$). The target's precision tolerance in both directions are equal ($W_t = W_n = W$) as we model a circular target. We can treat the endpoint distribution in *tangent axis* as moving target selection, and *normal axis* as static target selection, both in 1D space. We then propose two hypotheses based on this and result in the following two candidate models:

Hypothesis-1: uncertainty in *normal axis* can be treated as static target selection with only spatial constraints.

Based on this hypothesis, we present **Model-1**:

X_a is an absolute precision uncertainty which is independent of users' intention 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 of this Gaussian component are constants:

$$\mu_a=\begin{pmatrix} a_t \\ a_n \end{pmatrix} \text{ and, } \Sigma_a=\begin{pmatrix} d_t & 0 \\ 0 & d_n \end{pmatrix} \tag{4}$$

X_m depends on the uncertainty caused by the motion of the target. In *tangent axis*, it has been proven that the mean and standard deviation are both proportional to the moving velocity (V) [22]. In *normal axis*, based on **Hypothesis-1**, V does not affect the uncertainty in the *normal axis*, thus we can define distribution parameters of this Gaussian component as:

$$\mu_m=\begin{pmatrix} b_t V \\ 0 \end{pmatrix} \text{ and, } \Sigma_m=\begin{pmatrix} e_t V^2 & 0 \\ 0 & 0 \end{pmatrix} \tag{5}$$

X_s depends on the precision tolerance of the target. Previous works [10, 11, 45, 22] showed that for both moving targets and static targets, the distribution parameters of this Gaussian component are proportional to the target size:

$$\mu_s=\begin{pmatrix} c_t W \\ c_n W \end{pmatrix} \text{ and, } \Sigma_s=\begin{pmatrix} f_t W^2 & 0 \\ 0 & f_n W^2 \end{pmatrix} \tag{6}$$

Following the previous *Ternary-Gaussian* model [22], we model the speed-accuracy tradeoff relationship between X_m and X_s by setting their covariance to a term:

$$Cov(X_m, X_s)=\begin{pmatrix} g_t \frac{V}{W} & 0 \\ 0 & 0 \end{pmatrix} \tag{7}$$

X_m has no velocity component in *normal axis*, thus the covariance matrix of X_m and X_s does not contain value in *normal axis* either.

Evidences indicated that, users tried to acquire the target on the path where the target moves [7, 2, 24], thus, we considered the mean of endpoint distribution on *normal axis* as a negligible value and set $\mu_n \approx 0$ empirically.

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

$$\mu=\begin{pmatrix} a_t + b_t V + c_t W \\ 0 \end{pmatrix}, \tag{8}$$

and parameter $\sum$:

$$\Sigma=\begin{pmatrix} d_t + e_t V^2 + f_t W^2 + g_t \frac{V}{W} & 0 \\ 0 & d_n + f_n W^2 \end{pmatrix}. \tag{9}$$

As we can see, the standard deviation in *normal axis* is:

$$\sigma_n = \sqrt{d_n + f_n W^2}, \tag{10}$$

reflecting that, the uncertainty in *normal axis* is only related to spatial constraints (W). However, it's unsafe to simply determine that σ_n is not affected by other factors. Evidences showed factors such as temporal constraint [28, 41] or instruction conditions [45] affected uncertainty in static target selection. Therefore, we proposed another hypothesis and associated candidate model:

Hypothesis-2: uncertainty in *normal axis* can be treated as static target selection with both spatial and temporal constraints.

Based on this hypothesis, in addition to the spatial constraint, **Model-2** assumes the uncertainty in *normal axis* is also affected by the temporal urgency generated from target velocity. We explain this as follow.

For the whole catching movement, standard deviation of endpoints associated with V consists of two independent components. First, standard deviation of endpoints caused by the directionless rapid movement of hand. This component is proportional to the target speed (V) as user must move their hands with higher speed to catch faster targets no matter the target is moving in what direction [24, 25]. Second, standard deviation of endpoints generated from the difference of time delays of human sensory-motor system between individuals and trials [7, 28, 16]. This component is proportional to target velocity projected on each one of the two axes (i.e., V_t or V_n), since the endpoint displacements Δ_i equal to multiplying the delay by the target speed (Δ_i = delay × $V_i \mid i=\{t, n\}$). We call the first component as *rapid component*, and the second as *delay component*, these two components are independent because they are generated from different mechanisms.

Rapid component exists in the two axes since it is directionless, while *delay component* only exists in the *tangent axis* as $V_n = 0$. Further, only the *delay component* correlated to W because user can only use target's precision tolerance to compensate the displacement of endpoint, but they cannot use it to change the *rapid component*, which is directionless and determined by individual physical quality. Thus, it is in fact that the V_t yields the term V/W in *tangent axis* as $V_t = V$. Thus, we have the same formulations of σ_t with **Model-1**:

$$\sigma_t = \sqrt{d_t + e_{t-d}V_t^2 + e_{t-r}V^2 + f_tW^2 + g_t\frac{V_t}{W}}, \quad = \sqrt{d_t + e_tV^2 + f_tW^2 + g_t\frac{V}{W}} \tag{11}$$

but additional factor V is added to the formulation of σ_n presenting the effects of temporal constraints:

$$\sigma_n = \sqrt{d_n + e_nV^2 + f_nW^2}. \tag{12}$$

By setting the ignorable stochastic parameter μ_n to zero, we have the final Gaussian distribution with parameter μ:

$$\mu = \begin{pmatrix} a_t + b_tV + c_tW \\ 0 \end{pmatrix}, \tag{13}$$

and parameter $\sum$:

$$\sum = \begin{pmatrix} d_t + e_tV^2 + f_tW^2 + g_t\frac{V}{W} & 0 \\ 0 & d_n + e_nV^2 + f_nW^2 \end{pmatrix}. \tag{14}$$

The symbols a, b, c, d, e and f with subscripts t or n in the two models are constant coefficients, which can be measured via experiments. We conducted an experiment to evaluate the two candidate models.

STUDY 1: MODEL VALIDATION

In this section, we evaluated the proposed two candidate models with 2D pointing tasks defined in **Figure 1**. Furthermore, we validated the robustness of the proposed models with three input modalities including mouse, stylus and finger touch.

Participants and Apparatus

We recruited 13 subjects (6 females and 7 males, with an average age of 25) in this study. All subjects are right-handed and are familiar with computer, as well as the three pointing devices.

For sake of being consistent between different input modalities, the experiment were conducted on a HiteVision X7 interactive system with a build-in computer and a 70 inches LED display at 1,920×1,080 resolution, which supports the three input modalities. The mouse was a dell MS111 mouse (1000 dpi) with a constant CD gain of 10. We chose to use a constant CD gain to eliminate software-level enhancement for mouse (e.g., pointer acceleration in Windows' default setting). Stylus and finger input had a 0.8mm pointing accuracy with the system. We used a common carbon pencil as the stylus, which is supported by the touch screen. By considering the screen size and control area of users' hand, we implemented the experiment in a window with in a physical size of 1023×574mm.

Design and Procedure

The experiment contained 48 conditions, with 3 input modalities crossed 4 levels of W and 4 levels of V:

- *Modalities:* Mouse, Stylus and Finger Touch
- *W:* 16, 32, 64 and 96 pixels
- *V:* 96, 192, 288 and 384 pixels/sec

The initial distance A was randomized between 128 to 192 pixels as it has been proved to have no effects on the endpoint distribution. Initial azimuth and moving direction were both randomized between 0 to 360° as defined in the task. Each condition included 10 trials resulting a total of 13 participants × 48 conditions × 10 trials = 6240 clicks. The order of input modalities was counterbalanced across participants, and they can take a break between trials. It took about 30 minutes to complete the study.

In each trial, a participant clicked the "start" button on the center of the window to start. After a short interval (i.e. randomized from 700 - 2,000ms), the device played a beep sound and displayed the moving target. The target is a blue circular target with a specified diameter initialized at a certain radius distance and a certain azimuth from the "start" button. The target moved with a fixed speed and direction right after it appeared on screen. Participants were asked to acquire the target as quickly and accurately as they could. They could only point on the target once per trial, regardless of whether they hit the target or not. We recorded the coordinates of all endpoints.

		Mouse				Stylus				Finger Touch			
		tangent axis		normal axis		tangent axis		normal axis		tangent axis		normal axis	
		μ_t	σ_t	μ_n	σ_n	μ_t	σ_t	μ_n	σ_n	μ_t	σ_t	μ_n	σ_n
Dual-Gaussian	R^2	-	0.088	-	0.629	-	0.095	-	0.670	-	0.146	-	0.755
	MAE	10.646	4.259	0.857	1.855	14.262	1.923	0.676	0.834	12.500	1.926	0.488	0.629
	MWD	149.268				279.555				227.817			
Model-1	R^2	0.956	0.951	-	0.629	0.989	0.897	-	0.670	0.971	0.896	-	0.755
	MAE	1.280	0.912	0.857	1.855	0.716	0.681	0.676	0.834	1.216	0.654	0.488	0.629
	MWD	3.461				1.814				2.603			
Model-2	R^2	0.956	0.951	-	0.958	0.989	0.897	-	0.939	0.971	0.896	-	0.944
	MAE	1.280	0.912	0.857	0.562	0.716	0.681	0.676	0.358	1.216	0.654	0.488	0.323
	MWD	**3.340**				**1.769**				**2.579**			

Table 1. The fitting results of the two candidate models in the three modalities.

Measures

We used three measurements including r-squared (R^2), mean absolute error (MAE), and mean Wasserstein distance (MWD) as the fitting scores for the candidate models. Since a 2D Gaussian distribution contains 4 parameters (μ_t, σ_t, μ_n and σ_n), in addition of using R^2 and MAE to measure them separately, we use MWD to give an overall fitting score for a model.

Wasserstein distance is a statistical distance defined between two probability distributions. Intuitively, the metric is the minimum "cost" of turning one pile (distribution) into the other. The mean of WD (i.e., MWD) is the average Wasserstein distance from actual distributions to the predicted distributions across all conditions, it ranged from 0 to positive infinity, and a smaller MWD indicates a higher fitting score.

Thus, we had 9 metrics to evaluate the goodness of fit for each model. As a baseline, we also evaluated the *Dual-Gaussian* model in FFitts Law [10] into the comparison.

Results

In total, we got 48 sets of endpoints correspond to 48 conditions. The majority of them (44 sets) passed normality test using 2D Kolmogorov-Smirnov with a confidence level of 95%. We estimated the actual μ and σ via maximum likelihood estimation (MLE) for each of the 48 Gaussian distributions. We used the `nlinfit` function provided in `MATLAB` to estimate each coefficient of the candidate models, and evaluated the models' performance with the metrics calculated by comparing the predicted Gaussian distributions with the actual 48 Gaussian distributions.

Table 1 shows the fitting results of the *Dual-Gaussian* and the two candidate models with the three input modalities. The best MWD values are marked bold. Because μ_t and μ_n in *Dual-Gaussian* model and μ_n in the two candidate models are set to zero, we do not report R^2 of these parameters.

Overall, the two models showed good fits to the endpoint distributions on all input modalities. They outperformed the *Dual-Gaussian*, with MWD values ranged from 1.76 to 3.46 compared to the values ranged from 149.26 to 279.55. We can see from Table 1 that the improvement came from the better fits of μ and σ in *tangent axis*. This is because the *Dual-Gaussian* model does not take the uncertainty generated from V into account.

That the MAE of μ_t of the two candidate models ranged from 0.71 to 1.28 indicated they successfully describe the shifting effects of the means of endpoints. In contrast, that *Dual-Gaussian* model simply treated the μ_t as zero lead to a much larger MAE of μ_t ranged from 10.64 to 14.26. That the MAE of σ_t of the two candidate models ranged from 0.65 to 0.91 indicated that they predicted the variabilities of endpoints closely to the actual data, which was much better than the MAE ranged from 1.92 to 4.25 in *Dual-Gaussian* model. This indicated that V indeed increases the standard deviation of endpoint in target selection. R^2 values of these two parameters reflected consistent results.

Although the two candidate models empirically set μ_n to zero, this choice was appropriate as they got low enough MAE of μ_n ranged from 0.48 to 0.85, which were mean errors less than 1 pixel.

Dual-Gaussian and **Model-1** had the same fitting performances for σ_n with R^2 ranged from 0.62 to 0.75, and with MAE ranged from 0.62 to 1.85. Their performances were significantly lower than **Model-2**, with R^2 ranged from 0.93 to 0.95, and with MAE ranged from 0.32 to 0.56. This is because *Dual-Gaussian* and **Model-1** assumed that the uncertainty in *normal axis* only affected by spatial constraints, while **Model-2** considered both spatial and temporal constraints for σ_n by adding V into the formulation.

Because of the better fitting for σ_n, **Model-2** outperformed **Model-1** with MWD values ranged from 1.76 to 3.46 compared the values ranged from 1.81 to 3.46 for all three input modalities. Thus, we choose **Model-2** as the final 2D

Ternary-Gaussian model. We provide detailed analysis of this model in the three input modalities in the following sections.

Model Fitting

Figure 2 shows the actual endpoints mapped on the targets with 16 width × velocity conditions on three different input modalities. It is obvious that the endpoints of mouse are more dispersed than the other two modalities. This may be because mouse is an indirect pointing device. Users need to convert the control coordinate to the screen coordinate, which bring more systematic uncertainty compared to the direct input modalities. In addition, certain settings such as CD gain, dpi, and screen size may also affect the selectin uncertainty of the mouse. The distributions of the stylus and finger touch are relatively close. This may be because the use of the 70 inches large display tends to overwhelm the differences between finger and the stylus. However, these differences still exist as we will see in the later analysis.

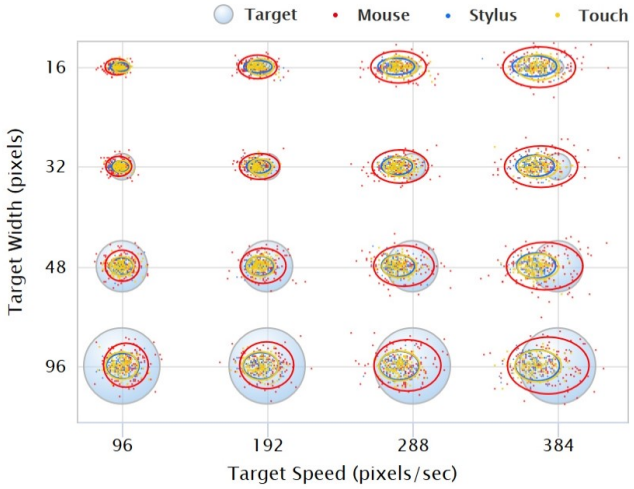

Figure 2. Actual endpoints and predicted distributions of 16 width × velocity conditions on three different input modalities. The gray circles are the targets, and the solid ellipses are 95% confidence ellipses of predicted distributions. Colors of red, yellow and blue were used to distinguish the three modalities.

Overall, despite differences exist across devices, the proposed 2D *Ternary-Gaussian* model fitted the endpoint distribution well for all the three input modalities. For mouse, the model achieved R^2 values of 0.95, 0.95 and 0.95 for μ_t, σ_t and σ_n, respectively; for stylus, the values are 0.98, 0.89 and 0.93, and 0.97, 0.89 and 0.94 for finger touch. Although the R^2 values for σ_t were lower in stylus and finger touch compared to mouse, the model got even better performances in MWD in these two modalities. This is because that the model fitted better for μ_t in the latter two input modalities resulted in a higher overall distribution similarity across conditions. As show in **Figure 2**, the confidence ellipses of predicted distributions nearly contain 95% of the actual endpoints in all conditions.

Model Coefficients

Table 2 shows the coefficients of the model in three modalities. The first two columns give the Gaussian parameters and their associated coefficients; the third column indicates the category that the coefficients belong to; the next three columns give the estimated coefficients for the three input modalities.

		Term	Mouse	Stylus	Touch
μ_t	a_t	-	-3.5997	3.1386	5.9644
	b_t	V	-0.0811	-0.1230	-0.1171
	c_t	W	0.1306	0.0474	0.0062
σ_t	d_t	-	17.6601	0.0000	0.0019
	e_t	V^2	0.0063	0.0011	0.0012
	f_t	W^2	0.0122	0.0079	0.0076
	g_t	V/W	0.8433	5.5052	4.0429
σ_n	d_n	-	3.2071	3.6062	9.1229
	e_n	V^2	0.0017	0.0004	0.0003
	f_n	W^2	0.0172	0.0049	0.0050

Table 2. The estimated coefficients of the model.

From coefficients of μ_t, we learned that all input modalities have a negative b_t, indicated that, it is a general phenomenon that target speed shifts the mean of endpoints in the opposite direction to the moving direction. It was interesting to see the coefficient a_t on the mouse is a negative value. This might be because that the lag of clicking the button on the mouse leads to the absolute and negative shift of the mean of endpoints. We found a larger value of c_t in mouse. This told us that the users may rely more on using W to compensate the shift effects of target speed with mouse.

From coefficients of σ_t, we learned that the mouse is much more uncertain than the other two direct input modalities, where larger d_t, indicated a larger absolute variability of the device, while larger e_t and f_t, told us that the endpoint variabilities can be more easily to be increased by target speed and target size on mouse. We found a much smaller g_t in mouse compare to stylus and touch. This can be explained from another perspective for the larger standard deviation in mouse, that users can hardly use target size to compensate the standard deviation caused by target speed.

From coefficients of σ_n, we learn that there are also larger variabilities generated from target speed and target size on mouse. And touch input shows a d_n three times larger than the other two. This may be because that this modality has larger absolute variability when treated as static target (in *normal axis*), which consists with the *Dual-Gaussian* model in FFitts Law [10].

Although the proposed model contained 10 coefficients, considering its prediction of multiple variables (i.e. μ_t, σ_t and σ_n), robustness across devices and interpretability for the user behavior in moving target acquisition, we believed that it is

still acceptable, and it is worthwhile to further study the performance of the model in practice.

STUDY 2: ASSISTING MOVING TARGET SELECTION

2D-BayesPointer

In this section, we proposed a *2D-BayesPointer*, a novel interaction technique to aid moving target selection in 2D space. The main idea of *2D-BayesPointer* is using the 2D *Ternary-Gaussian* model as the likelihood function in Bayes' rule to infer the intended target when user clicked. The *2D-BayesPointer* works as follows:

Assuming there are n targets $T=\{t_1,t_2,\ldots,t_n\}$ in a workspace. A user points on that workspace and yields an endpoint s. Then the conditional probability that t ($t \in T$) is the intend target is $P(t|s)$, which can be calculated with the Bayes formula:

$$P(t \mid s) = \frac{P(s \mid t)P(t)}{P(s)}, \qquad (15)$$

where $P(t)$ denotes the prior probability of selecting t, which is equal for each target; $P(s|t)$ is the likelihood function which consisted with the probability density function (PDF) of endpoint distribution; $P(s)$ is the normalization constant that holds the same across targets. Determining the intended target is equivalent to finding $t*$ ($t* \in T$) that has a maximum $P(s|t)$ among all the targets. This process is illustrated in **Figure 3**.

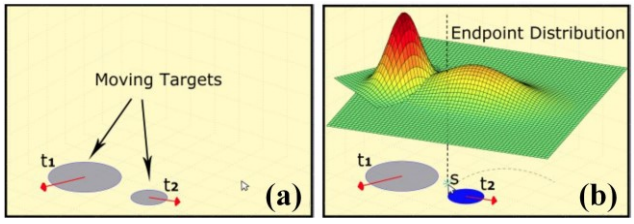

Figure 3. a) Two moving targets in the workspace; b) *2D-BayesPointer* determined t_2 as the intended target because $P(s|t_2) > P(s|t_1)$.

To avoid the situation that *2D-BayesPointer* always returns an intended target even when users intentionally click on a blank space, a click that falls outside the range of the 99% confidence ellipse is omitted.

Experiment

We conducted an experiment to evaluate the *2D-BayesPointer*. The experiment was a pointing test with multiple circle targets moving in a workspace. Participants used a computer mouse to select a specified target assigned by the system. We compared the time and accuracy performance of our technique with other two state-of-the-art moving target selection techniques including *Bubble Cursor* [19] and *Comet* [21] in the experiment. We also add the Windows basic selection technique (i.e. *Basic*) in the comparison as baseline. **Figure 4** showed the interfaces of the four techniques.

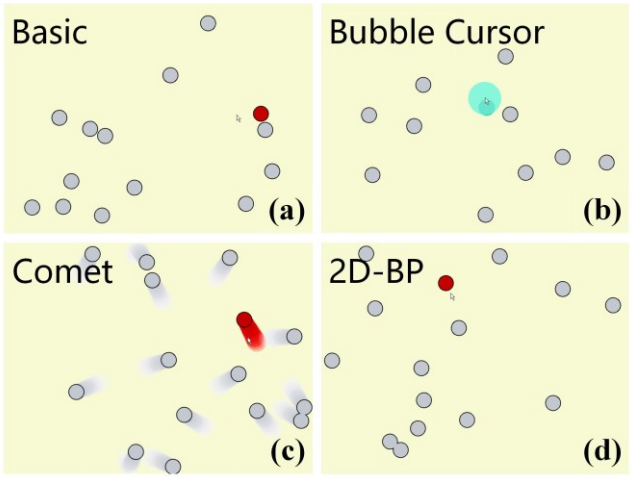

Figure 4. Interfaces of the four techniques.

Participants and Apparatus

Sixteen subjects (6 females and 10 males, average age 26) were recruited to participate in the experiment. All subjects were right-handed and were familiar with computer and mouse. We ran the experiment on a regular desktop computer with a 23 inches display at 1,920×1,080 resolution. A Dell MS111 mouse was used with a same setting as in **Study 1**.

Design and Procedure

We leveraged a within-subjects design to compare between the four techniques (i.e. *Basic*, *Bubble Cursor*, *Comet* and *2D-BayesPointer*,) under 16 conditions consisted of 4 levels of W (i.e. 24, 48, 96 and 144 pixels) crossed by 4 levels of V (i.e. 96, 192, 288 and 384 pixels/sec). Each subject was asked to play 10 trials in each condition, yielding 4 *Techniques* × 4 W × 4 V × 16 participants × 10 trials = 10240 trials in total. It took about 20 minutes for each participant to finish the test. The trials within and between W × V conditions were randomly assigned and the order of techniques was counterbalanced between participants.

In a trial, 15 circle targets appeared randomly in a workspace and moved toward random directions. All the targets had the same W and V. When the targets hit the edges of the workspace, they bounced back from the edges. One of the targets was colored red and participants were asked to select it as accurately as possible and as fast as possible. A trial finished until the participants successfully selected the red target.

For building the *2D-BayesPointer*, we had to estimate the coefficients of the 2D *Ternary-Gaussian* model in this experimental setting. Therefore, before the formal experiment, we conducted a calibration test for the estimation. The test has the same procedure as **Study 1**, but with the experiment apparatus and design in this study.

Measures

We collected task completion time and error rates for all trials. Task completion time was the time duration from trial start until the participant successfully selects the red target.

Error rate was calculated as the number of failed selections divide by the total number of selections.

Results

We used the repeated-measures ANOVA test for all our analyses in this study. Greenhouse-Geisser correction was used for the violation of sphericity.

Task Completion Time

Results showed a main effect of *Technique* ($F_{1.03,15.47}$=32.999, p<.001), *W* ($F_{1.40,21.07}$=5.617, p=.019) and *V* ($F_{3,45}$=3.348, p=.027) on completion time. Significant interaction effect was observed for the pair of *Technique* × *W* ($F_{9,135}$=4.237, p<.001). Other interaction effects were not significant. Pairwise comparisons using the Bonferroni adjustment yielded significant differences across all pairs of techniques (p<.05) except pairs of *2D-BayesPointer* vs. *Bubble Cursor* (p=.119), *Comet* vs. *Bubble Cursor* (p=.106). *2D-BayesPointer* had the lowest average completion time (1026ms), followed by *Bubble Cursor* (1074ms), *Comet* (1123ms), and *Basic* (2371ms). **Figure 5** shows the average completion time across *Technique* with varied *V* and *W*.

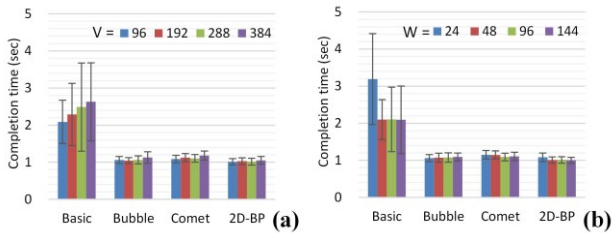

Figure 5. Task completion time across techniques with varied V (a) and W (b).

Error Rate

Results showed a main effect of *Technique* ($F_{1.77,2.39}$=197.208, p<.001), *W* ($F_{3,45}$=3.848, p=.016) and *V* ($F_{3,45}$=4.564, p=.007) on error rate. No significant interaction effects were exhibited (all p>.05). Pairwise comparisons showed significant differences across all pairs of techniques (p<.05) except *Comet* vs. *Bubble Cursor* (p=.457). The lowest error rate was achieved by *2D-BayesPointer* (12.1%), followed by *Comet* (19.6%), *Bubble Cursor* (20.6%) and *Basic* (57.2%). **Figure 6** shows error rates across *Technique* with varied *W* and *V*.

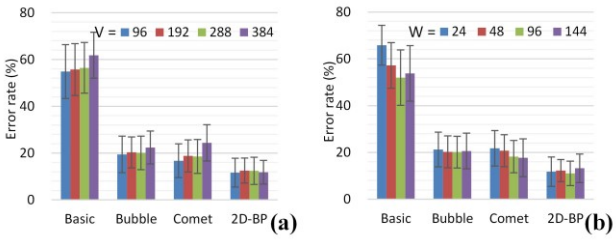

Figure 6. Error rates across techniques with varied V (a) and W (b).

In summary, compared with the two state-of-the-art target selection techniques, *2D-BayesPointer* outperformed *Comet* 4.4% in pointing speed, while it outperformed both *Bubble Cursor* 38.2% and *Comet* 41.2% in pointing accuracy.

STUDY 3: MODELING ERROR RATES IN A GAME INTERFACE

Error Rate Model

The error rate of a pointing task is defined as the percentage of endpoints that fall outside the target among all pointing attempts. With endpoint distribution predicted by 2D *Ternary-Gaussian* model, we can further calculate the error rate for pointing a rectangular target via multivariate normal cumulative distribution function (CDF) [44] as follow:

$$P(x_{top}, x_{right}) = \frac{1}{2\pi\sigma_t\sigma_n}\int_{-\infty}^{x_{top}}\int_{-\infty}^{x_{right}} \exp(-\frac{z}{2}), \quad (16)$$

where

$$z = \frac{(x_{top}-\mu_t)^2}{\sigma_t^2} + \frac{(x_{right}-\mu_n)^2}{\sigma_n^2}. \quad (17)$$

$P(x_0, x_1)$ represents the probability that X falls into the range from -∞ to x_0 in horizontal direction, and from -∞ to x_1 in vertical direction. Then the error rate is the probability that X falls out of left (x_{left}), right (x_{right}), bottom (x_{bottom}) and top (x_{top}) boundaries of the target:

$$ErrorRate = 1-[P(x_{right}, x_{top}) - P(x_{left}, x_{bottom})], \quad (18)$$

We use the `mvncdf` function provided in `MATLAB` to calculate $P(x_{top}, x_{right})$ and $P(x_{left}, x_{bottom})$ for us.

For targets with other shape such as ellipse, we can obtain the error rate by either integrating CDF of differential rectangles over the shape, or computing CDF over convex regions, please see [35] for details.

Experiment

We conducted an experiment to explore the feasibility of using the 2D *Ternary-Gaussian* model to predict error rates in real-world applications. We used a popular game named *Ant Smasher* (**Figure 7** (a)) as the testbed. In the game, a player had to kill all ants running to the picnic blanket by tapping them with the finger. The game involved selecting multiple 2D rectangle targets, and required fast searching and reaction abilities of players.

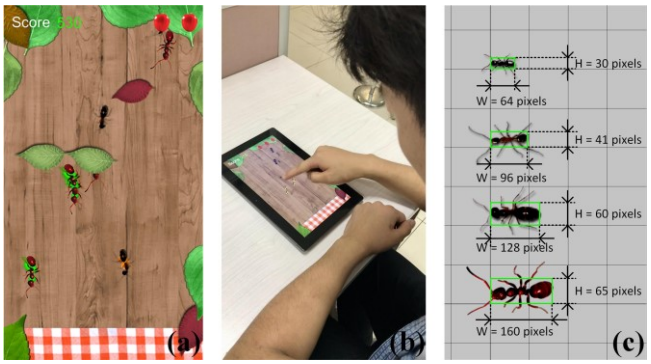

Figure 7. The tested game *Ant Smasher*. a) The interface; b) a participant playing the game; c) sizes of the bounding boxes of the ants.

The game was implemented on a Microsoft Surface tablet computer with a 10.6 inches touch screen (**Figure 7** (b)). In the game, ants randomly appeared and moved from the top of the screen to the blanket on the bottom. Players had to tap the ants to kill them before they reached the blanket, otherwise the players lost one life (three in total) for each ant. The ant was killed if the players tapped inside of a bounding box that had a same size as the ants' main body not including limbs, as showed in **Figure 7** (c).

Methods

Participants

Twelve subjects (6 females and 6 males, average age 26.9) were recruited to participate in the experiment. All subjects were right-handed and were familiar with computer and touch screen.

Design and Procedure

The ants had 4 different sizes (width × height) including 64 × 30, 96 × 41, 128 × 60 and 160 × 65 pixels, and 4 different speeds including 96, 192, 384 and 768 pixels/sec. New ants spawned per 1.75 second to 0.25 second. All conditions were randomized when new ant spawned. Player gained 20 to 40 scores when they killed an ant, as the player got higher score, new ant would be spawned faster and more likely to move with faster speed. Ants were spawned at a random location on the top of screen, and moved in a random direction heading to the bottom. The moving direction was dynamically fixed in a range preventing the ant from moving out of the screen before they reached the bottom.

Each participant had to complete 4 gameplays. In total, we had 4 trials × 12 participants = 72 trials. Participants could take a break between trials. It took about 12 minutes for each participant to finish the experiment. Participants practiced one trail before starting the formal study.

In a trial, the participant clicked a button to start the game. Once the game started, participants should keep playing until they lost all the three lives. All endpoints, no matter succeeded or failed, were recorded.

Measures

We measured the error rates for all 16 size × velocity conditions. As there are multiple targets on the screen, to collect the endpoints and calculate the error rate for a specified target, we had to determine that which target a tap event belong to. We set the target nearest to the endpoint as the intended target. Then we calculated the error rate for a condition as the number of failed selections divide by total number of selections for the targets in this condition.

Results

We used the `nlinfit` function provided in `MATLAB` to estimate the coefficients of 2D *Ternary-Gaussian* model with the data collected in the game. Since the ants were rectangle targets, target height was used in the formula of σ_n. With the CDF mentioned earlier, our model estimated error rates for all the 16 conditions of ants, and fitted the data well with a 0.94 R^2.We further performed a repeated two-fold cross-validation to test the generalizability of our model. The model coefficients were obtained over the data of 6 randomly chosen subjects and tested on the rest 6. Over 100 iterations, we obtained average MAE of 3.5% as displayed in **Figure 8**.

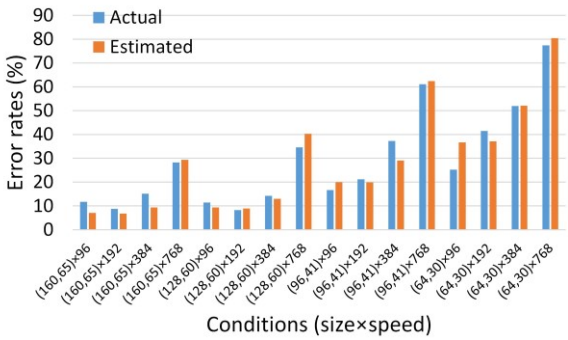

Figure 8. Actual and estimated error rates for 16 conditions.

As shown in **Figure 8**, the error rate increased when the target speed increased and when the target size decreased. This trend has been well described by our model. According to the results, although the game involved additional interferences such as the highly required quick reaction and visual search ability of players, our model performed well and showed robustness across conditions.

CONCLUSION

The model proposed in this paper precisely described the distribution of the endpoints for targets moving in two-dimensional space. Results showed that our model fit the empirical data well and showed robustness across devices. The model reflected general mechanisms of the catching movement and specific factors of devices as well. When using our model to assist selection of moving targets, it outperformed other two state-of-the-art techniques. We also demonstrated how our model can be used to predict error rates in a game interface design. The model successfully described the fact that error rate increases with target speed and decreases with target size.

We found that when users point on 2D circular moving targets, in the direction perpendicular to the velocity, the mean of endpoints is close to target center and the standard deviation is related to target speed. The effects of target speed on selection uncertainty may related to the inaccuracy of hand's rapid motion and the delay of human sensory-motor system. However, this paper has limited considerations about the influences of device settings, target shape and time constraints on selection uncertainty, which need to be further explored in the future.

ACKNOWLEDGMENTS

We sincerely thank Shumin Zhai for his scientific guidance in this study. This work was funded by National Key R&D Program of China (Grant No. 2016YFB1001405), and National Natural Science Foundation of China (Grant No. 61802379), and Key Research Program of Frontier Sciences, CAS (Grant No. QYZDY-SSW-JSC041), and partially supported by CAS Pioneer Hundred Talents Program.

REFERENCES

[1] Brouwer A M, Middelburg T, Smeets J B J, et al. 2003. Hitting moving targets: A dissociation between the use of the target's speed and direction of motion. *Experimental Brain Research*. 152, 3, 368-375. http://dx.doi.org/10.1007/s00221-003-1556-8

[2] Brouwer A, Brenner E, Smeets J B, et al. 2002. Hitting moving objects: is target speed used in guiding the hand. *Experimental Brain Research*, 143, 2, 198-211. http://dx.doi.org/10.1007/s00221-001-0980-x

[3] Johnny Accot and Shumin Zhai. 1997. Beyond Fitts' law: models for trajectory-based HCI tasks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '97), 295-302. http://dx.doi.org/10.1145/1120212.1120376

[4] 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

[5] 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

[6] 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

[7] Eli Brenner, Jeroen B J Smeets,. 1996. Hitting moving targets: Co-operative control of 'when'and 'where', *Human Movement Science*. 15, 1, 39-53. http://dx.doi.org/10.1016/0167-9457(95)00036-4

[8] Watson A B, Robson J G. 1981. Discrimination at threshold: labelled detectors in human vision. *Vision Res*, 21, 1115-1122. http://dx.doi.org/10.1016/0042-6989(81)90014-6

[9] Watson, A. B., & Ahumada, A. J. 1985. Model of human visual-motion sensing. *Journal of The Optical Society of America A-optics Image Science and Vision*, 2, 2, 322-342. http://dx.doi.org/10.1364/JOSAA.2.000322

[10] 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

[11] 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

[12] Crossman E, Goodeve P J. 1983. Feedback control of hand-movement and Fitts' law. *The Quarterly Journal of Experimental Psychology Section A*, 35, 2, 251-278. http://dx.doi.org/10.1080/14640748308402133

[13] Todorov Emanuel and Jordan I. Michael. 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

[14] 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

[15] Francis G, Kim H. 2001. Perceived motion in orientational afterimages: direction and speed, *Vision research*. 41, 2, 161-172. http://dx.doi.org/10.1016/S0042-6989(00)00242-X

[16] Stépán, Gábor. 2009. Delay effects in the human sensory system during balancing. *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences*. 367, 1195-212. http://dx.doi.org/10.1098/rsta.2008.0278

[17] Schweigart G, Mergner T, Barnes G. 2003. Object motion perception is shaped by the motor control mechanism of ocular pursuit. *Experimental brain research*, 148, 3, 350-365. http://dx.doi.org/10.1007/s00221-002-1306-3

[18] 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

[19] 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

[20] 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

[21] 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

[22] Jin Huang, Feng Tian, Xiangmin Fan, Xiaolong (Luke) Zhang, and Shumin Zhai. 2018. Understanding the

Uncertainty in 1D Unidirectional Moving Target Selection. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (CHI '18), ACM, to be determined. http://dx.doi.org/10.1145/3173574.3173811

[23] Jin Huang, Xiaolan Peng, Feng Tian, Hongan Wang, and Guozhong Dai. 2018. Modeling a target-selection motion by leveraging an optimal feedback control mechanism. Science in China Series F: Information Sciences 61, 4 (2018), 044101. http://dx.doi.org/10.1007/s11432-017-9326-8

[24] Bairstow P J. 1987. Analysis of hand movement to moving targets. *Hum Mov Sci*, 6, 205–231. http://dx.doi.org/10.1016/0167-9457(87)90013-3

[25] Bootsma R J, Wieringen PCW van. 1990. Timing an attacking forehand drive in table tennis. *J Exp Psychol Hum Percep Perform*, 16, 21-29. http://dx.doi.org/10.1037//0096-1523.16.1.21

[26] 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

[27] Nakayama K. 1985. Biological image motion processing: a review. *Vision research*, 25, 5, 625-660. http://dx.doi.org/10.1016/0042-6989(85)90171-3

[28] 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

[29] Byungjoo Lee, Sunjun Kim, and Antti Oulasvirta. 2018. Moving Target Selection: A Cue Integration Model. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (CHI '18), to appear. http://dx.doi.org/10.1145/3173574.3173804

[30] 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

[31] I Scott MacKenzie and Poika Isokoski. 2008. Fitts' throughput and the speed-accuracy tradeoff. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '08), 1633–1636. http://dx.doi.org/10.1145/1357054.1357308

[32] Matthews N, Luber B, Qian N, et al. 2001. Transcranial magnetic stimulation differentially affects speed and direction judgments. *Experimental Brain Research*, 140, 4, 397-406. http://dx.doi.org/10.1007/s002210100837

[33] Thompson P. 1983. Discrimination of moving gratings at and above detection threshold. *Vision Res*, 23, 1533-1538. http://dx.doi.org/10.1016/0042-6989(83)90166-9

[34] Senanayake R, Goonetilleke R S. 2016. Pointing Device Performance in Steering Tasks. *Perceptual and Motor Skills*, 122, 3, 886-910. http://dx.doi.org/10.1177/0031512516649717

[35] Paul N. Somerville. 1999. A Fortran 90 Program for Evaluation of Multivariate Normal and Multivariate t Integrals Over Convex Regions. *Journal of Statistical Software*. 3, 4, 1-10. http://dx.doi.org/10.18637/jss.v003.i04

[36] Jeroen B J Smeets, Eli Brenner. 1995. Perception and action are based on the same visual information: distinction between position and velocity. *Journal of Experimental Psychology: Human Perception and Performance*, 21, 1, 19. http://dx.doi.org/10.1037//0096-1523.21.1.19

[37] Mackenzie I S. 2015. Fitts' Throughput and the Remarkable Case of Touch-Based Target Selection. *International conference on human-computer interaction*, 238-249. http://dx.doi.org/10.1007/978-3-319-20916-6_23

[38] MacKenzie, I. S. 1992. Fitts' law as a research and design tool in human-computer interaction. *Human-Computer Interaction*, 7, 91-139. http://dx.doi.org/10.1207/s15327051hci0701_3

[39] 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

[40] 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

[41] 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

[42] Franklin D W, Wolpert D M. 2011. Computational Mechanisms of Sensorimotor Control. *Neuron*, 2011, 72, 3, 425-442. http://dx.doi.org/10.1016/j.neuron.2011.10.006

[43] Zhou X, Cao X, Ren X, et al. 2009. Speed-Accuracy Tradeoff in Trajectory-Based Tasks with Temporal Constraint.In *Proceedings of the SIGCHI Conference*

on Human Factors in Computing Systems (CHI '09), 906-919. http://dx.doi.org/10.1007/978-3-642-03655-2_99

[44] Drezner, Z. 1994. Computation of the Trivariate Normal Integral. *Mathematics of Computation*, 63, 289–294. http://dx.doi.org/10.2307/2153409

[45] 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

[46] Paul N. Somerville. 1999. A Fortran 90 Program for Evaluation of Multivariate Normal and Multivariate t Integrals Over Convex Regions. *Journal of Statistical Software*. 3, 4, 1-10. http://dx.doi.org/10.18637/jss.v003.i04