

Modeling Touch Point Distribution with Rotational Dual Gaussian Model

Yan Ma
yanma1@cs.stonybrook.edu
Stony Brook University
Stony Brook, NY, USA

IV Ramakrishnan
ram@cs.stonybrook.edu
Stony Brook University
Stony Brook, NY, USA

Shumin Zhai
zhai@acm.org
Google
Mountain View, CA, USA

Xiaojun Bi
xiaojun@cs.stonybrook.edu
Stony Brook University
Stony Brook, NY, USA

ABSTRACT

Touch point distribution models are important tools for designing touchscreen interfaces. In this paper, we investigate how the finger movement direction affects the touch point distribution, and how to account for it in modeling. We propose the Rotational Dual Gaussian model, a refinement and generalization of the Dual Gaussian model, to account for the finger movement direction in predicting touch point distribution. In this model, the major axis of the prediction ellipse of the touch point distribution is along the finger movement direction, and the minor axis is perpendicular to the finger movement direction. We also propose using *projected* target width and height, in lieu of nominal target width and height to model touch point distribution. Evaluation on three empirical datasets shows that the new model reflects the observation that the touch point distribution is elongated along the finger movement direction, and outperforms the original Dual Gaussian Model in all prediction tests. Compared with the original Dual Gaussian model, the Rotational Dual Gaussian model reduces the RMSE of touch error rate prediction from 8.49% to 4.95%, and more accurately predicts the touch point distribution in target acquisition. Using the Rotational Dual Gaussian model can also improve the soft keyboard decoding accuracy on smartwatches.

CCS CONCEPTS

• **Human-centered computing** → **Human computer interaction (HCI)**; Modeling.

KEYWORDS

Touch input; modeling

ACM Reference Format:

Yan Ma, Shumin Zhai, IV Ramakrishnan, and Xiaojun Bi. 2021. Modeling Touch Point Distribution with Rotational Dual Gaussian Model. In *The 34th Annual ACM Symposium on User Interface Software and Technology (UIST '21, October 10–14, 2021, Virtual Event, USA*

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

UIST '21, October 10–14, 2021, Virtual Event, USA

© 2021 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 978-1-4503-8635-7/21/10.

<https://doi.org/10.1145/3472749.3474816>

'21), October 10–14, 2021, Virtual Event, USA. ACM, New York, NY, USA, 13 pages. <https://doi.org/10.1145/3472749.3474816>

1 INTRODUCTION

Models for predicting the distribution of touch points have served as important tools to design touchscreen interfaces and interaction techniques. One of such models is the Dual Gaussian model [5] which predicts the touch point distribution in a target acquisition task with the target width W and height H . It assumes that touch points follow a bivariate Gaussian distribution, which is the sum of two Gaussian distributions: one Gaussian is governed by the width and height of the target, and the other Gaussian represents the input uncertainty of finger touch. More specifically, the Dual Gaussian model assumes that the touch point distribution X can be calculated as follows:

$$X \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma), \quad (1)$$

where $\boldsymbol{\mu}$ is the center of the target, and Σ is related to the width W and height H of the target:

$$\Sigma = \begin{bmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{bmatrix} = \begin{bmatrix} a + bW^2 & 0 \\ 0 & c + dH^2 \end{bmatrix}, \quad (2)$$

where a, b, c , and d are empirically determined parameters. The Dual Gaussian model has been used to determine the element sizes for touchscreen UI design, to determine the intended target in target acquisition tasks, and serves as the likelihood model in soft keyboard decoding.

The Dual Gaussian model is simple and easy to use. However, it ignores an important factor: the finger movement direction. It is known that the movement direction affects the shape of the distribution of end points: end points tend to be elongated along the movement direction [16, 19]. Could we factor in the finger movement direction in modeling the touch point distribution? If we could, would it improve the model fitness over the existing Dual Gaussian model? These are the research questions we investigated. In this paper, we propose to factor in the finger movement direction θ in the Dual Gaussian model by hypothesizing that the one axis of the prediction ellipse of the distribution is along the finger movement direction, and the other axis is perpendicular to the finger

movement direction. The co-variance matrix then becomes:

$$\Sigma_R = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{xy} & \sigma_y^2 \end{bmatrix} \quad (3)$$

where

$$\begin{aligned} \sigma_x^2 &= \cos^2 \theta (a + bW_p^2) + \sin^2 \theta (c + dH_p^2) \\ \sigma_{xy} &= \cos \theta \sin \theta (a + bW_p^2) - \cos \theta \sin \theta (c + dH_p^2) \\ \sigma_y^2 &= \sin^2 \theta (a + bW_p^2) + \cos^2 \theta (c + dH_p^2) \end{aligned}$$

where W_p and H_p are called projected target width and height, which are the lengths of the line segments by projecting the target along the two axes of the prediction ellipse. The coefficients a , b , c , and d are empirically determined parameters.

We call the modified model the *Rotational Dual Gaussian model*. Our evaluation on three empirical datasets shows that the Rotational Dual Gaussian model outperforms the original Dual Gaussian model in all prediction tests. Compared with the original Dual Gaussian model, the new model reduces the Root Mean Square Error (RMSE) of touch error rate prediction in target acquisition task from 8.49% to 4.95%; it more accurately predicts the touch point distribution in target acquisition; using the Rotational Dual Gaussian model reduces the soft keyboard decoding error rates on both smartwatches and smartphones.

2 RELATED WORK

Our work builds on previous research on understanding finger touch, modeling touch point distribution, and keyboard decoding.

2.1 Understanding Finger Touch

As the dominant input modality in mobile computing, touch input has been widely investigated by a considerable amount of research. Researchers have investigated soft button user performance compared to hard buttons with different functions of implement (finger vs. stylus), feedbacks (auditory and tactile-vibrato), and button size [24]. Holz and Baudisch's research [18] showed that the touch points have consistent offsets from the target, and the offsets were affected by angles (pitch, roll, and yaw) between the finger and the touch surface. Their subsequent studies [19] showed that users' perceived contact points were above actual contact points along the finger's axis and could be approximated by the center of the fingernail. Other factors such as the "fat finger" problem, which is caused by occlusion, also affect touch accuracy [18, 19, 30–32].

2.2 Modeling Touch Point Distribution

A sizable amount of previous research showed touch points followed bivariate Gaussian distributions. A study [33] conducted by Wang and Ren compared the properties of touch input among all five fingers, showing touch points followed bivariate Gaussian distributions and the variance differs across fingers. Previous research [3, 17] also revealed that for text entry on a touch screen keyboard of a phone-sized device, touch points followed bivariate Gaussian distributions, and the means of the distributions were close to the intended key center, but often with small offsets in different directions. The dual Gaussian hypothesis [5, 6] proposed by Bi and Zhai

identified and separated the two sources of end point variance: one is the speed-accuracy trade-off in the human motor system, and the other is the absolute imprecision of finger touch due to the "fat" finger. Ample empirical evidence [4, 5, 23, 25] has validated that the dual Gaussian model improves performance in modeling reciprocal target acquisition tasks.

While pointing models were typically derived to predict the movement time (*MT*), several previous works also presented models for predicting success rate (or error rate) in pointing tasks [7, 28, 35, 36, 38]. Meyer et al.'s work [28] first predicted the error rate, but it did not consider the effect of moving speed on precision [39]. Wobbrock et al. [35, 36] derived an error model from Fitts' law to predict the precision for 1D pointing tasks with mouse and stylus input, which accounted for the speed-accuracy tradeoff in the human motor system. Bi and Zhai [7] derived a prediction model from the dual Gaussian hypothesis [5] to predict the success rate for off-screen-start target acquisition tasks while accounting for both the speed-accuracy tradeoff and the ambiguity brought by finger touch. The validity of their work was further extended to predict success rate for on-screen-start target acquisition tasks [38].

Previous research revealed that touchpoints over a moving target also follow Gaussian distributions. Huang et al.'s research [21] showed that endpoints in 1D unidirectional moving target acquisition tasks followed a Ternary-Gaussian distribution that contains three Gaussian components: the first reflected the absolute uncertainty of a motor system including the input device; the second was caused by the motion of the target; the third was governed by target size. Their subsequent work [22] extended the Ternary-Gaussian model to model the uncertainty in touchpoint distribution over 2D moving targets. They also extended their work to a Quaternary-Gaussian model [20] that measured the endpoint uncertainty in crossing-based moving target selection.

Building upon previous works, we extended the dual Gaussian model [5] by considering movement direction and predicted the error rate for 2D target acquisition tasks with the Rotational Dual Gaussian model.

3 ROTATIONAL DUAL GAUSSIAN MODEL

We propose a Rotational Dual Gaussian model to factor in the finger movement direction in predicting the touch point distribution. The main difference between the Rotational Dual Gaussian and the original Dual Gaussian model is that the Rotational Dual Gaussian model reflects the observation that end point distribution is often elongated along the finger movement direction. In this section, we first briefly review the Dual Gaussian model, and then describe how we advance it to become the Rotational Dual Gaussian model.

3.1 Original Dual Gaussian Model

The original Dual Gaussian model assumes that touch point distribution in a target acquisition task follows a bivariate Gaussian distribution, along the visual target coordinate of x (horizontal) and y (vertical) directions. More specifically, it states that the touch point distribution can be modeled with the following model:

$$X \sim \mathcal{N}(\mu, \Sigma), \quad (4)$$

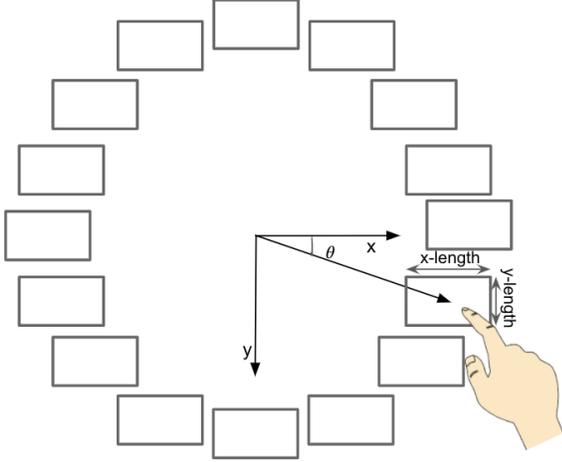


Figure 1. An illustration of movement angle. θ is the angle between the moving direction and the positive x-axis of the screen coordinate system.

where μ and Σ are the mean and co-variance matrix of the bivariate distribution, which can be calculated as:

$$\mu = (\mu_x, \mu_y) = (0, 0), \quad (5)$$

$$\Sigma = \begin{bmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{bmatrix} = \begin{bmatrix} a + bW^2 & 0 \\ 0 & c + dH^2 \end{bmatrix}. \quad (6)$$

As shown in Equation 5 and 6, the Dual Gaussian model assumes that the variance in x and y directions are independent, and the variance in each direction is quadratically related to target width (or height) with the linear coefficient being 0. It also assumes that the mean of the distribution is at the center of the target.

3.2 Deriving the Rotational Dual Gaussian Model

One weakness of the original Dual Gaussian model is that it assumes the correlation coefficient between x and y is always 0, regardless of the finger movement direction. Empirically this is inconsistent with the observation that the end point distribution tends to be elongated along the pointer movement direction. Theoretically, the original Dual Gaussian model is only logical when the movement direction is aligned with the horizontal x dimension. Recall that the original Fitts' law study [12] was on the univariate relationship between movement time and movement amplitude constraint (W) along the movement direction. Accot and Zhai's study [1] shows constraint orthogonal to the movement direction is much less dominant and takes on average around 100 ms less time to conform to. However it is also necessary to keep the UI coordinate in the x and y visual space rather than in the variable motor control space depending on the movement direction. We propose the Rotational Dual Gaussian model to address this conflict.

Similar to the original Dual Gaussian model, the Rotational Dual Gaussian model assumes that touch points follow a bivariate Gaussian distribution. The main difference is that it assumes that major axis of the prediction ellipse (e.g. 95% prediction ellipse) points to

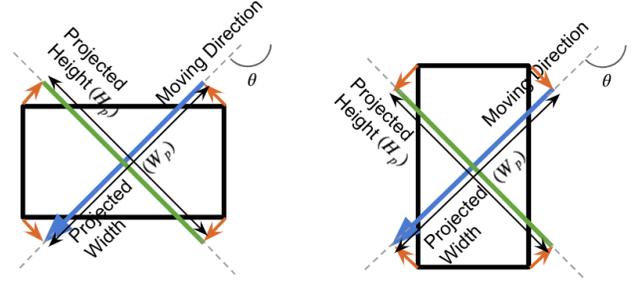


Figure 2. An illustration of using projected target width (W_p) and height (H_p) as amplitude and directional constraints. W_p (in blue) is the length of the line segment of target size projected on the movement direction. H_p (in green) is the length of the line segment of target size projected on the direction perpendicular to the movement direction.

the finger movement direction, and the other axis points to the direction perpendicular to the finger movement direction. This means that one eigenvector of the co-variance matrix points to the direction of movement direction of finger, and the other eigenvector points to the direction perpendicular to the movement direction.

We formally describe the Rotational Dual Gaussian model as follows. Given the finger movement direction θ , which is the angle between the positive x -axis and the movement direction (Figure 1), the Rotational Dual Gaussian model assumes that the touch points followed a bivariate Gaussian distribution X :

$$X \sim \mathcal{N}(\mu, \Sigma_R), \quad (7)$$

where μ is the same as the mean estimation in the original Dual Gaussian model (Equation 5), which is the center of the target, and Σ_R is the covariance matrix, which is calculated as follows.

To calculate Σ_R , we first perform eigendecomposition, which represents the matrix in terms of its eigenvalues and eigenvectors. With eigendecomposition, Σ_R becomes:

$$\Sigma_R = VLV^{-1} \quad (8)$$

, where V is a 2×2 matrix whose two columns are the two eigenvectors of Σ_R , and L is a 2×2 diagonal matrix whose values along the diagonals are eigenvalues of the two eigenvectors. L is also referred to as scale matrix.

As previously explained, we assume the two eigenvectors point to the directions along and perpendicular to the finger movement directions. Therefore, the two eigenvectors can be expressed as:

$$v_x = [\cos \theta, \sin \theta]^T \quad (9)$$

$$v_y = [-\sin \theta, \cos \theta]^T. \quad (10)$$

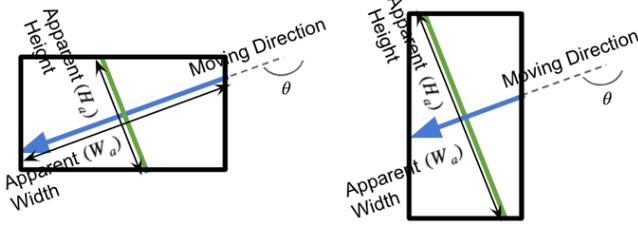


Figure 3. An illustration of using apparent target width (W_a) and height (H_a) as amplitude and directional constraints. W_a (in blue) is the length of the line segment intersecting the target size on the movement direction. H_a (in green) is the length of the line segment intersecting the target on the direction perpendicular to the movement direction.

The matrix V , whose columns are the two eigenvectors, can then be expressed as:

$$V = [v_x, v_y] \quad (11)$$

$$= \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \quad (12)$$

The scale matrix L defines the variance of the touch points along the two axes of the $(1 - \alpha)$ prediction ellipse. Similar to the assumption in the original Dual Gaussian model, we assume the variance along each axis is quadratically related to the size of the target projected to the axis (Figure 2). More specifically, L is defined as:

$$L = \begin{bmatrix} a + bW_p^2 & 0 \\ 0 & c + dH_p^2 \end{bmatrix} \quad (13)$$

where W_p is the projected target width which is the length of the line segment formed by projecting the target to the finger movement direction (Figure 2), and H_p is the projected target height which is the length of the line segment formed by projecting the target to the direction perpendicular to the finger movement direction (Figure 2). The coefficients $a, b, c,$ and d are empirically determined values.

Replacing V , and L in Equation 8 with Equations 12, and 13, we have

$$\Sigma_R = VL V^{-1} \quad (14)$$

$$= \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} a + bW_p^2 & 0 \\ 0 & c + dH_p^2 \end{bmatrix} \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \quad (15)$$

$$= \begin{bmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{xy} & \sigma_y^2 \end{bmatrix} \quad (16)$$

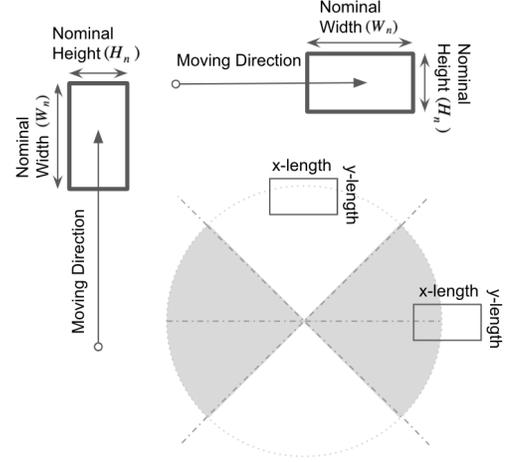


Figure 4. An illustration of using nominal target width (W_n) and height (H_n) as amplitude and directional constraints. If the movement direction falls within the grey area, x -length is W_n and y -length is H_n ; if the direction is within the white area, y -length is W_n and x -length is H_n .

where

$$\sigma_x^2 = \cos^2 \theta (a + bW_p^2) + \sin^2 \theta (c + dH_p^2)$$

$$\sigma_{xy} = \cos \theta \sin \theta (a + bW_p^2) - \cos \theta \sin \theta (c + dH_p^2)$$

$$\sigma_y^2 = \sin^2 \theta (a + bW_p^2) + \cos^2 \theta (c + dH_p^2)$$

Equation 16 relates the co-variance matrix of the Rotational Dual Gaussian model to the target width, height, and finger movement direction.

The Rotational Gaussian model becomes the regular Dual Gaussian models when the θ is 0° , or 180° . When the $\theta = 0^\circ$, we have $W_p = W$ and $H_p = H$. Plugging these values into the Equation 16, the co-variance matrix becomes

$$\Sigma_R = \begin{bmatrix} a + bW^2 & 0 \\ 0 & c + dH^2 \end{bmatrix} \quad (17)$$

which is exactly the same as the covariance matrix defined in the original Dual Gaussian model (Equation 6).

3.3 Prediction of Correlation Coefficient

The Rotational Dual Gaussian hypothesis models the co-variance matrix of touch point distribution given target size and movement direction. Hence, it can also model the Pearson's correlation coefficient ρ between variable x and y . The following equation shows another way to express the co-variance matrix in Equation 16

$$\Sigma_R = \begin{bmatrix} \sigma_x^2 & \rho \sigma_x \sigma_y \\ \rho \sigma_x \sigma_y & \sigma_y^2 \end{bmatrix} \quad (18)$$

Comparing Equation 18 and Equation 16 yields,

$$\rho = \frac{\sigma_{xy}}{\sigma_x \sigma_y} \quad (19)$$

Model		a	b	c	d	WAIC	RMSE
Original Dual Gaussian Model		2.403 [2.335, 2.469]	0.017 [0.016, 0.018]	2.295 [2.211, 2.379]	0.016 [0.015, 0.017]	500410.01	0.085
Rotational Dual Gaussian Model	a) W_n, H_n	2.508 [2.423, 2.597]	0.024 [0.023, 0.025]	1.938 [1.887, 1.988]	0.013 [0.011, 0.012]	496655.82	0.073
	b) W_a, H_a	2.105 [2.022, 2.185]	0.026 [0.025, 0.027]	1.723 [1.673, 1.777]	0.013 [0.013, 0.014]	496183.65	0.066
	c) W_p, H_p	2.043 [1.976, 2.108]	0.017 [0.017, 0.018]	1.476 [1.434, 1.516]	0.010 [0.010, 0.010]	492654.91	0.049

Table 1: Mean and 95% Credible Interval of model parameters for 4 model candidates. WAIC measures prediction accuracy giving consideration of model complexity. The smaller the value, the better a model. RMSE of error rate shows the Rotational Dual Gaussian model with projected widths and heights performed the best in predicting accuracy of target acquisition tasks.

where

$$\sigma_x = \sqrt{\cos^2 \theta (a + bW_p^2) + \sin^2 \theta (c + dH_p^2)}$$

$$\sigma_y = \sqrt{\sin^2 \theta (a + bW_p^2) + \cos^2 \theta (c + dH_p^2)}$$

$$\sigma_{xy} = \cos \theta \sin \theta (a + bW_p^2) - \cos \theta \sin \theta (c + dH_p^2)$$

Equation 19 predicts the correlation (ρ) between x and y .

3.4 Two alternatives of Rotational Dual Gaussian Model

In addition to using projected target width W_p and height H_p in Equation 16, we also propose the following two alternatives of defining the variance of the touch points, as they have been used to represent the finger movement constraints in the previous work [2, 23].

- Apparent Target Width and Height (Figure 3). This option is to use the apparent width W_a and height H_a to replace W_p and H_p in Equation 16. Apparent width W_a is the length of the line segment intersected within the target along the line parallel to the finger movement direction and crossing the target center, and the apparent height H_a is the length of the line segment perpendicular to the finger movement direction and crossing the target center. This definition is the same as the apparent width and height definition used in the previous work [2, 23]
- Nominal Width and Height (Figure 4). As another baseline or approximation, this option is to use nominal width (x -length) and height (y -length) of a target to replace W_p and H_p in Equation 16. The width and height defined with this option are referred to as W_n and H_n , respectively. More specifically, if θ falls within the ranges of $[0^\circ, 45^\circ]$, $[135^\circ, 225^\circ]$, $[315^\circ, 360^\circ]$, we use x -length to replace W_p (i.e., $W_n = x$ -length), and y -length to replace H_p (i.e., $H_n = y$ -length). If θ is with $[45^\circ, 135^\circ]$ and $[225^\circ, 315^\circ]$, we use y -length to replace W_p (i.e., $W_n = y$ -length) and x -length to replace H_p (i.e., $H_n = x$ -length). This option essentially simplifies the W_p and H_p

calculation of angled conditions to their nearest vertical or horizontal conditions.

In sum, the main difference between the Rotational Dual Gaussian and the original Dual Gaussian model is that the Rotational Dual Gaussian model replaces the co-variance matrix Σ (Equation 6) with the new co-variance matrix Σ_R (Equation 16). We have proposed three options of calculating the co-variance matrix of the distribution, namely using (1) projected width W_p and height H_p , (2) apparent width W_a and height H_a , or (3) nominal width W_n and height H_n . Other part of the Rotational Dual Gaussian model stays the same with the original Dual Gaussian Model.

Next, we evaluate the proposed Rotational Dual Gaussian model in target acquisition tasks [23], in typing tasks on smartphone [3], and typing tasks on smartwatches.

4 EVALUATION ON 2D TARGET ACQUISITION TASKS

We first evaluated whether the Rotational Dual Gaussian model improved the accuracy in predicting the distribution of touch points and target selection accuracy over the original Dual Gaussian model in 2D target acquisition tasks. We conducted the evaluation on a publicly available dataset about 2D target acquisition with finger touch – Ko et al’s Dataset [23].

4.1 Ko et al’s dataset

The empirical 2D touch pointing dataset was collected by Ko et al. [23]. It had 3 distances, 16 finger movement directions, and 21 different target sizes (width \times height). The experiment included target acquisition data from 18 participants, and each participant contributed 3301 trials on average. In total, it included 59427 target selection trials. Following the practice of Ko et al., we excluded trials whose selection time fell beyond 3 std (1.08% of total trials), and we ended up having 58785 trials. We performed the Mardia’s Multivariate Normality test [27] for the 336 (width \times height \times angle) conditions. Mardia’s test showed that the touch points in 69% of the conditions followed Gaussian distribution. Touch points in 98% of the conditions passed either the Skewness test or the Kurtosis test, indicating that their skewness or Kurtosis measure was close to a Gaussian distribution.

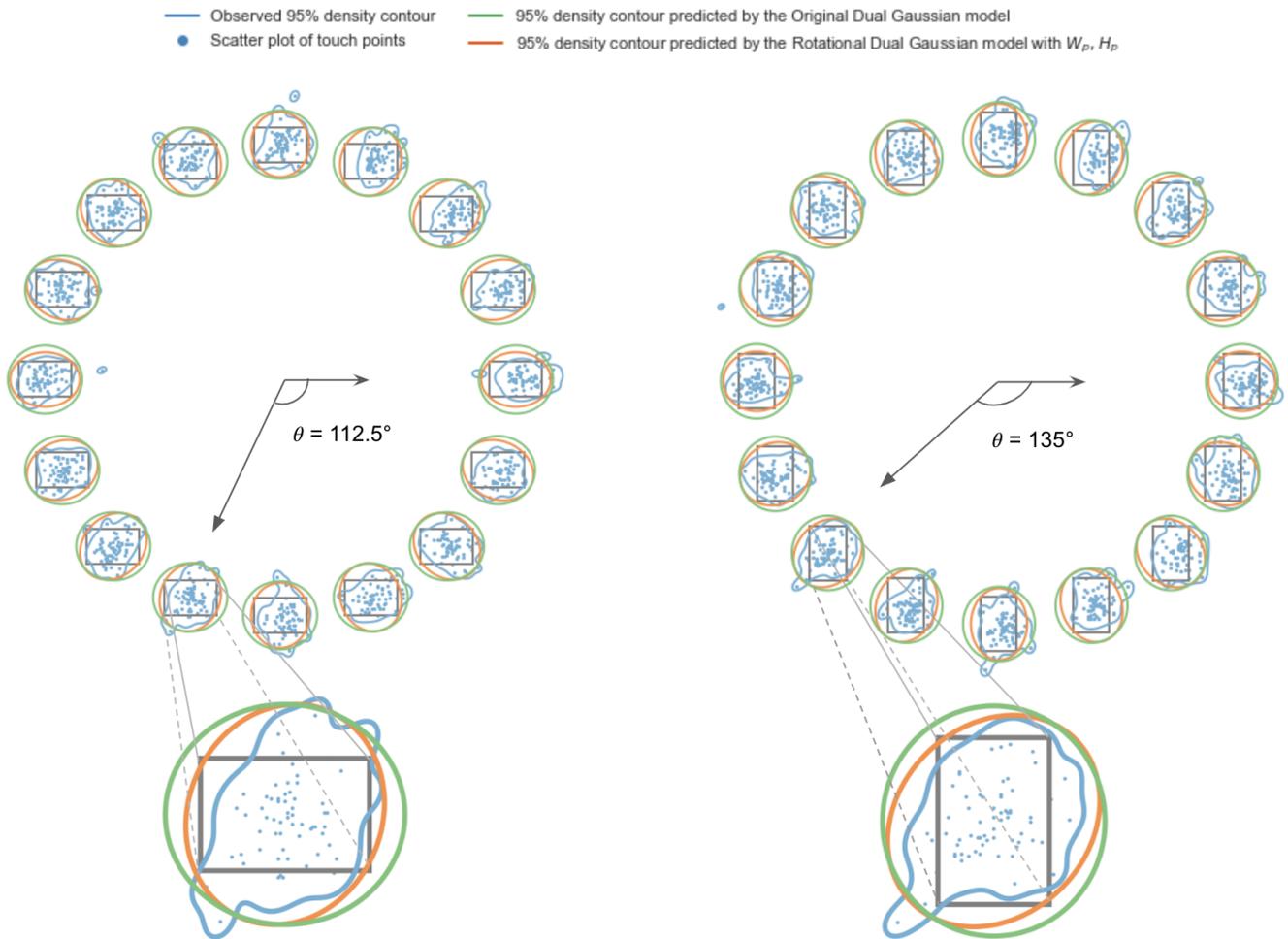


Figure 5. Prediction of 95% density contours for two conditions. Left: target size is 6×4 and the movement angle to the enlarged target is 112.5° . Right: target size is 4×6 and the movement angle to the enlarged target is 135° . This figure shows the predicted contours by the Rotational Dual Gaussian model (in orange) revealed that touch points tend to be elongated along the movement direction, which resembled the observed touch point distribution (in blue). The predictions made by the original Dual Gaussian model (in green) remained the same across different movement angles, resulting in poor fitting compared to the Rotation Dual Gaussian model.

4.2 Model Candidates

There were four model candidates:

- (1) Original Dual Gaussian model (Equation 6),
- (2) Rotational Dual Gaussian model with projected width W_p and height H_p (Equation 16).
- (3) Rotational Dual Gaussian model with apparent width W_a and height H_a .
- (4) Rotational Dual Gaussian model with nominal width W_n and height H_n .

4.3 Fitting Models

We employed a Bayesian computational method to estimate parameters a , b , c , and d of both model candidates. We used Stan [8], a probabilistic programming language for Bayesian modeling

and statistical inference, to implement the models and obtain the estimation of parameter distributions.

We constructed the two models as follows. After we defined the models and passed the data, Stan inferred the posterior distribution of model parameters through Hamiltonian Monte Carlo (HMC), which is a Markov Chain Monte Carlo (MCMC) sampling method. Since we did not possess prior knowledge of parameters, we assumed uniform distributions as priors for all parameters. We used the default setting of Stan to fit the model, which employed 4 chains, and each chain included 1000 iterations in sampling. All R-hat convergence diagnostics were close to 1.0, indicating that between and within estimates for parameters agree so that the Markov chain converged to the estimated distributions. We summarized the means of posterior distributions of model parameters in Table 1.

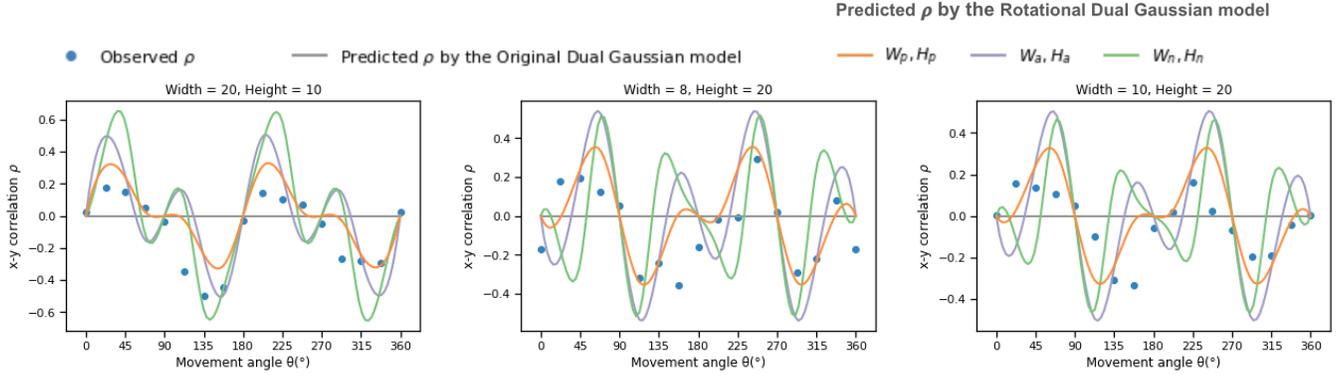


Figure 6. Prediction on x-y correlations (ρ) vs. movement angles (θ) in 3 target size (width \times height) conditions. The Rotational Dual Gaussian model successfully predicts the effect of movement angle on x-y correlations. Rotational model with projected widths and heights outperforms its alternatives. The original Dual Gaussian model assumes ρ is 0, which does not capture the changes of ρ as the movement direction θ changes.

Model	Original Dual Gaussian Model	Rotational Dual Gaussian Model		
		W_p, H_p	W_a, H_a	W_n, H_n
RMSE of predicted ρ	0.218	0.159	0.218	0.243

Table 2: Root Mean Square Error of predicted ρ . The Rotational Dual Gaussian Model with projected widths and heights outperforms three other model candidates.

4.4 Model Comparison

We examined the prediction accuracy of the four model candidates with the following four metrics.

Information Criteria. We first compared the prediction accuracy of the Rotational Dual Gaussian model with the original Dual Gaussian model using the Widely Applicable Information Criterion (WAIC) [14]. Table 1 shows that the Rotational Dual Gaussian model with projected widths and heights has the lowest WAIC, indicating that it has the strongest predictive power of touch point distribution among the four model candidates.

Error rate. We used the means of the posterior parameter distributions as parameters and calculated the observed and predicted error rates for 21 target sizes. More specifically, we calculated the predicted error rate using a sampling method. Given a target and the estimated touch point distribution, we first generated 10000 touch points from the estimated distribution and calculated the error rate as the percentage of touch points falling outside the boundaries of the target. We repeated this process 10 times for each target and used the average value over these 10 times as the predicted error rate for a given target. In 19 out of 21 width \times height conditions, the Rotational Dual Gaussian model with projected widths and heights outperforms the original Dual Gaussian model by predicting an error rate closer to the observed error rate. We also calculated the Root Mean Square Error (RMSE) between the predicted and observed error rates for the four model candidates. Table 1 shows that the Rotational Dual Gaussian model with projected widths and heights has the lowest RMSE of error rate in target acquisition tasks.

Prediction on Touch Point Distributions. We further inspected the performance of the Rotational Dual Gaussian model with projected widths and heights and the original Dual Gaussian model by comparing the 95% density contours predicted by both models with

the observed 95% density contours. We first simulated data from both models and calculated the probability density function (PDF) for the simulated and the observed data. Then, we obtained the 95% density contours from the PDFs. We generated the observed PDF using kernel density estimation (KDE) [11], which is a non-parametric method of estimating the PDF of a random variable. The simulated 95% density contours of touch points should resemble the observed ones if the model fits. We visually evaluated the two models by comparing the 95% density contours of touch points across (width, height, angle) tuples, as shown in Figure 5.

More specifically, for each width \times height \times angle condition, we draw 2000 sample points using each of the two models and plotted the 95% density contours against the 95% density contour of the observed touch points. Figure 5 shows the results for two conditions: (a) width = 6, height = 4 and (b) width = 4, height = 6. Compared to the original Dual Gaussian model, the Rotational Dual Gaussian model successfully predicts the orientation of the ellipse. It captures the tendency of touch points distributions elongated along the finger movement direction, and hence better predicts the shape of touch point distribution.

Prediction on x-y Correlations ρ . Additionally, we also evaluated the four model candidates by examining their prediction of Pearson's correlation coefficient between x and y coordinates of touch points over targets with different sizes. We followed Equation 19 to predict the x-y correlation coefficient and plotted it against the movement angle for 336 width \times height \times angle conditions in Ko et al's dataset [23]. Figure 6 shows three representative cases where (a) width = 10, height = 10, (b) width = 8, height = 20, and (c) width = 10, height = 20. As seen in the figure, x and y coordinates of touch points tend to present a positive correlation when $\theta \in (0^\circ, 90^\circ)$ or $\theta \in (180^\circ, 270^\circ)$ and a negative correlation when $\theta \in (90^\circ, 180^\circ)$ or $\theta \in (270^\circ, 360^\circ)$. The target width and height

affect how the x-y correlation changes with angle. Figure 6 shows that all three forms of the Rotational Dual Gaussian model successfully captured how the x-y correlations changes with the movement direction, but the magnitude of the changes is predicted differently. The correlation curves of the Rotational Dual Gaussian model with projected widths and heights better captured the trend of observed correlations relating to the movement angle than its two alternatives. Table 2 summarized the Root Mean Square Error (RMSE) of ρ prediction. As shown, Rotational Dual Gaussian model with W_p and H_p has lowest RMSE.

5 EVALUATION ON SMARTWATCH-BASED TYPING TASKS

To further evaluate the models, we tested them on a smartwatch typing dataset, as typing action can be viewed as a sequence of target acquisitions. We used each model in a statistical decoder, to test which model performed the best in decoding. Because of the shape (three rows of up to 10 keys) and letter arrangement (left and right alternation) on Qwerty keyboards, movements directions of one finger typing on such a layout tend to only slightly deviate from the horizontal axis. We therefore should expect small but still significant improvement of the Rotational Dual Gaussian model as the spatial model of decoding. To provide unbiased evaluation, we used the parameter values estimated on Ko et al's dataset [23] for each model (Table 1), and the typing dataset was viewed as a testing dataset.

5.1 Participants and Apparatus

We recruited 14 participants (4 females) aged from 23 to 32 (mean = 27.2, std = 2.9). All participants were right-handed and had experience with typing on a touchscreen keyboard. Participants performed typing tasks on a TicWatch S with Android API 28. The smartwatch is 44.958 mm in diameter and 12.954 mm thick. Participants wore the smartwatch on their non-dominant hand and typed with their index finger of the dominant hand throughout the experiment.

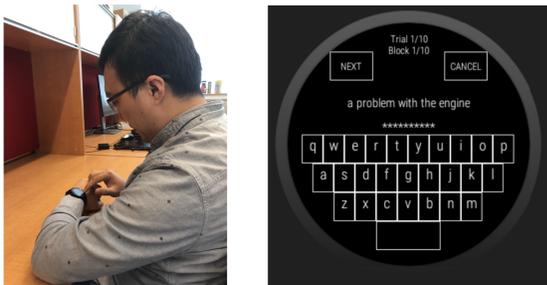


Figure 7. Left: a participant in the study. Right: a screenshot of the typing tasks.

Similar to the previous study [3], we designed a typing task where participants typed short phrases on a custom Android application. We randomly selected short and shortened phrases from the phrase set designed by MacKenzie and Soukoreff (M&S) [26, 37] for this study. As shown in Figure 7, we implemented an Android application that displayed a sequence of phrases, a simple QWERTY

soft keyboard, a NEXT button, and a CANCEL button. The key width was 3 millimeters for letters and 9 millimeters for the space key, while the key height was 4 millimeters. We preferred a simple interface to avoid complex visual elements having a positive or negative impact on the user's behavior. We recorded the positions of touch points that can be easily aligned with target characters.

This application displayed one short phrase at a time. Users completed typing a phrase by clicking on each letter key that appeared in the phrase, including the space key. After typing the entire phrase, users were required to click on the NEXT button to proceed to the next phrase. If users realized that they had input the wrong word in the middle of a phrase, they were allowed to retype the whole phrase by clicking on the CANCEL button. Meanwhile, this application logged the x and y coordinates of each touch point in millimeters along with the intended character in the presented phrase, including space, the NEXT button, and the CANCEL button. We also recorded the time that each touch appeared. The application logged touch points on "touch down" events. When users touched the soft keyboard, an asterisk appeared on the screen to provide limited feedback since we wanted to eliminate the influence of using a character detection algorithm, which was required to display the input character, on users' behavior.

5.2 Design and Procedure

Participants completed the study in 30 minutes on average and entered 110 short phrases while seated in an office environment. We divided the 110 phrases into 11 blocks, each containing 10 short phrases. Participants could take optional short breaks up to 2 minutes after each block, and the first block was a warm-up block. Similar to the previous study [3], before the study started, we instructed the participants to type "as accurately and naturally as possible." We collected a total of 29673 labeled touch points, excluding touch points from warm-up phases.

5.3 Results

As touch point distribution models also serve as spatial models in the statistical decoder of a soft keyboard, we compared the performance of four models in decoding accuracy.

Statistical Decoding Principle. The statistical decoder [6, 13, 15, 29, 34] works as follows. Given a set of n independent touch points on the keyboard $S = \{s_1, s_2, \dots, s_n\}$ and the lexicon L , the decoder will determine the best word W^* based on:

$$W^* = \arg \max_{W \in L} P(W|s_1, s_2, \dots, s_n) \quad (20)$$

Applying the Bayes' rule yields,

$$P(W|s_1, s_2, \dots, s_n) = \frac{P(s_1, s_2, \dots, s_n|W)P(W)}{P(s_1, s_2, \dots, s_n)} \quad (21)$$

where $P(s_1, s_2, \dots, s_n)$ is invariant across words. Therefore, substituting Equation 21 to Equation 20 is equivalent to the following:

$$W^* = \arg \max_{W \in L} P(s_1, s_2, \dots, s_n|W)P(W) \quad (22)$$

where $P(W)$ is estimated from a language model that in the current study is a unigram model trained from COCA Corpus [9, 10], and $P(s_1, s_2, \dots, s_n|W)$ is calculated as follows.

Model	Original Dual Gaussian Model	Rotational Dual Gaussian Model			Literal Strings
		W_p, H_p	W_a, H_a	W_n, H_n	
Mean of WER (SD)	28.89% (0.093)	28.54% (0.094)	28.62% (0.096)	28.71% (0.094)	80.31% (0.110)

Table 3: The mean (SD) of word-level error rates using the four model candidates and the literal strings in typing tasks on a smartwatch.

Participant	WER of Using the Original Dual Gaussian Model	WER of Using the Rotational Dual Gaussian Model			WER of Literal Strings
		W_p, H_p	W_a, H_a	W_n, H_n	
		1	9.91%	9.45%	
2	11.97%	11.44%	11.44%	11.97%	58.78%
3	40.32%	40.55%	40.55%	40.78%	92.86%
4	22.81%	22.58%	22.58%	22.58%	88.40%
5	32.87%	32.64%	32.64%	32.87%	71.26%
6	26.04%	25.35%	25.58%	25.58%	79.03%
7	36.55%	36.29%	36.55%	36.29%	84.77%
8	29.95%	29.72%	29.72%	29.95%	76.50%
9	34.56%	34.33%	34.56%	34.56%	93.55%
10	35.48%	35.48%	35.48%	35.48%	86.87%
11	23.50%	23.04%	23.04%	23.04%	61.29%
12	37.10%	37.10%	37.56%	37.56%	91.01%
13	35.71%	34.56%	34.56%	34.79%	87.33%
14	27.65%	26.96%	27.19%	27.19%	76.50%

Table 4: Word error rates per user on the smartwatch dataset collected by us. Literal strings use only key boundaries to determine which letter is typed without using any language model in decoding.

Model	Original Dual Gaussian Model	Rotational Dual Gaussian Model			Literal Strings
		W_p, H_p	W_a, H_a	W_n, H_n	
Mean of WER (SD)	3.35% (0.016)	3.21% (0.016)	3.23% (0.016)	3.33% (0.016)	30.06% (0.178)

Table 5: The mean (SD) of word-level error rates using the four model candidates and the literal strings in typing tasks on smartphone.

Assume the word W is composed of a sequence of n characters $\{c_1, c_2, \dots, c_n\}$. Since we assume s_1, s_2, \dots, s_n are independent, we obtain:

$$P(s_1, s_2, \dots, s_n | W) = \prod_{i=1}^n P(s_i | c_i) \quad (23)$$

The model that predicts $P(s_i | c_i)$ is also referred to as a spatial model, which can be approximated by each of the four touch point distribution models.

Using original Dual Gaussian model to calculate $P(s_i | c_i)$. Let (x_i, y_i) denotes the coordinates of s_i and (μ_{i_x}, μ_{i_y}) denotes the center of key c_i on the soft keyboard. W_i and H_i represent the width and height of key c_i on the keyboard. Applying the original Dual Gaussian model in Equation 5 and Equation 6, we have

$$P(s_i | c_i) = \frac{1}{2\pi\sqrt{(a+bW_i^2)}\sqrt{(c+dH_i^2)}} \exp\left[-\frac{z}{2}\right] \quad (24)$$

where

$$z \equiv \frac{(x_i - \mu_{i_x})^2}{a + bW_i^2} + \frac{(y_i - \mu_{i_y})^2}{c + dH_i^2}$$

Using Rotational Dual Gaussian model to calculate $P(s_i | c_i)$. Let (x_i, y_i) denotes the coordinates of s_i and (μ_{i_x}, μ_{i_y}) denotes the center of key c_i on the soft keyboard. W_p^i and H_p^i represent the projected width and height of key c_i on the keyboard. Let θ_i denotes the angle of the movement from the previous touch point s_{i-1} to the current target key center (μ_{i_x}, μ_{i_y}) . Applying the Rotational Dual Gaussian model in Equation 16, we have

$$P(s_i | c_i) = \frac{1}{2\pi\sigma_{i_x}\sigma_{i_y}\sqrt{1-\rho_i^2}} \exp\left[-\frac{z}{2(1-\rho_i^2)}\right] \quad (25)$$

Participant	WER of Using the Original Dual Gaussian Model	WER of Using the Rotational Dual Gaussian Model			WER of Literal Strings
		W_p, H_p	W_a, H_a	W_n, H_n	
		1	5.26%	4.97%	
2	1.77%	1.77%	2.06%	2.06%	11.50%
3	3.19%	2.98%	2.98%	3.40%	18.09%
4	2.31%	1.73%	2.31%	2.31%	14.12%
5	0.79%	0.79%	0.39%	0.39%	10.63%
6	5.03%	5.03%	4.86%	5.03%	43.38%
7	4.43%	4.06%	4.06%	4.43%	55.13%
8	1.86%	1.86%	1.69%	1.69%	19.49%
9	5.20%	5.20%	5.20%	5.20%	57.20%
10	3.76%	3.76%	3.76%	3.94%	31.84%

Table 6: Word-level error rates per user using the four model candidates and the literal strings in typing tasks on a smartphone.

where

$$z \equiv \frac{(x_i - \mu_{ix})^2}{\sigma_{ix}^2} - \frac{2\rho_i(x_i - \mu_{ix})(y_i - \mu_{iy})}{\sigma_{ix}\sigma_{iy}} + \frac{(y_i - \mu_{iy})^2}{\sigma_{iy}^2}$$

$$\rho_i = \frac{\cos \theta_i \sin \theta_i (a + bW_p^2) - \cos \theta_i \sin \theta_i (c + dH_p^2)}{\sigma_{ix}\sigma_{iy}}$$

$$\sigma_{ix} = \sqrt{\cos^2 \theta_i (a + bW_p^2) + \sin^2 \theta_i (c + dH_p^2)}$$

$$\sigma_{iy} = \sqrt{\sin^2 \theta_i (a + bW_p^2) + \cos^2 \theta_i (c + dH_p^2)}$$

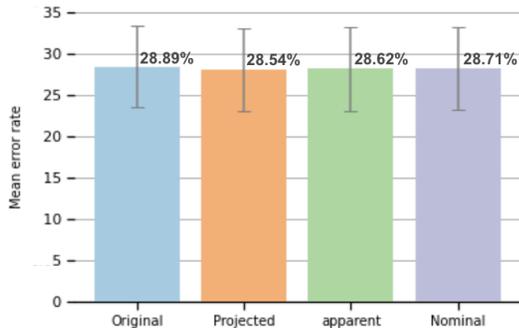


Figure 8. A comparison of WER for different models on a smartwatch. The gray error bars show 95% confidence intervals.

Word error rate (WER) [13] is a standard way of measuring decoding accuracy in a word-based task. Table 3 and Figure 8 show the decoding accuracy measured by word error rate of using four models, and also the literal string error rate [13] which uses only key boundaries to determine which letter is typed and used no language model in decoding. As shown, the mean WER across all participants was 28.54% (SD = 0.09) for the Rotational Dual Gaussian model with projected widths and heights, 28.89% (SD = 0.09) for the original Dual Gaussian model, and 80.31% (SD = 0.11) for the literal strings.

The Rotational Dual Gaussian model further reduced the decoding error rate than the original Dual Gaussian model. Table 4 shows the decoding error rates for each user with the four model candidates. The Rotational Dual Gaussian model with projected widths and heights reduced the WER for 11 out of 14 users. For 2 out of 14 users, the WER stayed the same, and the WER slightly increased for only 1 user. The increase in decoding accuracy was small because the language model contributed to correct most errors. Repeated-measures ANOVA showed the decoding method had a main effect on edit distance ($F(3, 39) = 7.409, p = 0.0005$). Pairwise comparison with Bonferroni adjustment showed the difference between error rates of using the Rotational Dual Gaussian model with projected widths and heights and the original Dual Gaussian model were significant ($p < 0.05$).

6 EVALUATION ON A SMARTPHONE TYPING DATASET

Besides testing models on smartwatch-based typing tasks, we also tested them on a smartphone-based typing dataset, the Azenkot and Zhai's Dataset [3].

6.1 Azenkot and Zhai's Dataset

This empirical text entry dataset was collected to investigate touch behavior with different postures on virtual smartphone keyboards by Azenkot and Zhai [3]. They recruited 32 participants for a between-subjects study, where 11 subjects entered text with two thumbs, 11 with one thumb, and 10 with one index finger. They collected x and y pixel coordinates of touch points during text entry on a custom Android application, which they called the Keyboard Touch Collector (KTC). KTC displayed a sequence of short phrases and a simple keyboard. When a user clicked on the KTC keyboard, an asterisk appeared on the screen as limited visual feedback. They collected 86888 labeled touch points in total, excluding touch points from warm-up phases. We used their 24126 labeled touch points collected from one index finger in this evaluation.

6.2 Results

We followed the same procedures to evaluate the performance of the four model candidates in a statistical decoder on Azenkot and Zhai’s Dataset [3]. Table 5 shows the decoding accuracy of using four model candidates and the literal string accuracy. The mean of WERs across all participants was 3.21% (SD = 0.016) for the Rotational Dual Gaussian model with projected widths and heights, 3.35% (SD = 0.016) for the original Dual Gaussian model, and 30.06% (SD = 0.178) for the literal strings. Table 6 and Figure 9 show the decoding error rates for each user using the four model candidates. The Rotational Dual Gaussian model with projected widths and heights reduced the WER for 4 out of 10 participants, and the WER remained unchanged for the rest 6 participants. The differences between error rates using the Rotational Dual Gaussian model and the original Dual Gaussian model were not significant ($F(3, 27) = 1.851, p = 0.162$). The result was not significant since the language model and the original Dual Gaussian model were capable of correcting most of the errors.

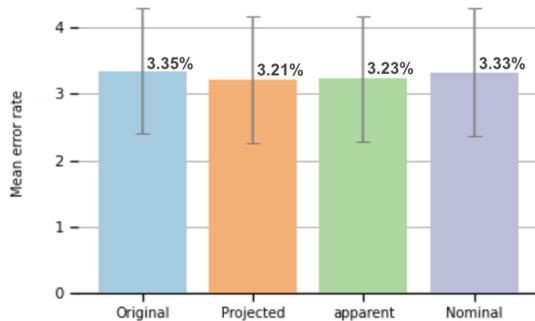


Figure 9. A comparison of WER for different models on a smartphone. The gray error bar shows 95% confidence intervals.

7 GENERAL DISCUSSION

As a key enabler to modern mobile computing and a main difference from the desktop interface technologies, touch input has attracted significant empirical and theoretical studies in HCI literature [e.g. 6, 19, 33]. Finger touch input has the advantage of being direct therefore intuitive, but suffers from the lack of mouse pointer level precision. The most rigorous model accounting for the “fat finger” imprecision to date is the Dual Gaussian model of finger touch [5, 6]. However, the original Dual Gaussian model did not consider the case therefore weakened its prediction when the movement direction is misaligned with the primary target constraint dimension. The model’s independent variables were the nominal target width W in the horizontal x dimension and nominal target height H in the vertical y dimension only, regardless of the movement direction θ . We generalize this line of modelling work with the Rotational Dual Gaussian model whose independent variables also include movement direction θ . Through mathematical derivation and empirical testing on several data sets, we demonstrated the value and validity of the generalized model, particularly along the following three aspects.

7.1 Modeling distribution of touch points

The Rotational Dual Gaussian model with projected widths and heights performs the best in modeling touch point distribution. Its prediction accuracy on Ko et al’s dataset [23] measured by WAIC is the strongest among all the four model candidates including the original Dual Gaussian model and the two alternatives of the Rotational Dual Gaussian model. Compared to the original Dual Gaussian model, it reduces the Root Mean Square Error of prediction error rate from 8.49% to 4.95% in on-screen-starting target acquisition tasks.

7.2 Applications of the Rotational Dual Gaussian model

The Rotational Dual Gaussian model proposed in this paper advances the Dual Gaussian Distribution model by factoring in the finger movement direction. It also advances the previous research from modeling 1D or circular targets to modeling 2D rectangular targets. Such an advancement has high practical value as rectangular targets are more commonly seen on current touchscreen interfaces than 1D or circular targets. Examples of 2D rectangular targets include buttons, icons, hyperlinks, and menu items. The proposed model benefits the following applications. First, given target widths and heights, the Rotational Dual Gaussian model more accurately predicts the finger touch accuracy than the existing Dual Gaussian model. Therefore, it can be used by UI designers in designing and evaluating touchscreen user interfaces.

Second, using the Rotational Dual Gaussian model as the likelihood model in soft keyboard decoding can improve the decoding accuracy. Replacing the original Dual Gaussian model with the selected Rotational Dual Gaussian model in a statistical decoder reduces the mean decoding error rate from 28.89% to 28.54% on smartwatches and from 3.35% to 3.21% on smartphones. The improvements in decoding accuracy are small for two types of reasons. The first is due to the Qwerty layout on which movement directions do not vary much away from the horizontal direction. The second reason is due to the use of the language model that has a significant contribution to reducing the error rate and the original Dual Gaussian model is already able to correct most errors. When applying the Rotational and the original Dual Gaussian model to smartwatches, the differences in decoding accuracy are still statistically significant. Using the Rotational Dual Gaussian model on a smartwatch reduced the decoding error rates for 11 out of 14 users, and kept the error rate unchanged for 2 out of 14 users. Although the magnitude of improvement is small, it is generalizable, and using the Rotational Dual Gaussian model to replace the original Dual Gaussian model has almost no cost. As text entry is a vital task of touchscreen interaction, any improvements in the error rate will positively affect the user’s experience. The promising results showed it is worth adopting the Rotational Dual Gaussian model to decoding, especially for small soft keyboards (e.g., keyboards on smartwatches).

Third, it can be used to predict the touch accuracy for on-screen-starting pointing actions: moving finger from one place on the screen to select a target located at another place on the screen. The on-screen-starting pointing is an important part of smartphone interaction [38], such as (1) moving finger from the bottom (or top)

of a screen to tap a 'Like' button after scrolling through a social networking service feed, or (2) successively inputting check marks on a questionnaire.

7.3 Effects of finger movement direction on distribution

Parameter estimation showed that $a = 2.043$ is larger than $b = 1.476$ and $c = 0.017$ is larger than $d = 0.010$ in the Rotational Dual Gaussian model. This is consistent with previous findings where end points tend to vary along the movement direction. In the Rotational Dual Gaussian model, a and b are parameters factoring variation along the movement direction, while c and d account for the direction perpendicular to the movement direction. A larger a and b enables the Rotational Dual Gaussian model to better capture the shape of touch point distribution.

Another implication of the Rotational Dual Gaussian model is that projected target size (W_p model), which is obtained from projecting the target onto the moving direction and the direction perpendicular to the moving direction, has a good performance in modeling touch point distribution. Both W_p and W_a models convert target width and height into constraints along and perpendicular to the movement direction. However, W_p and H_p values are always greater than or equal to the corresponding W_a and H_a values. It shows that W_a and H_a underestimate both directional and amplitude constraints, compared with W_p and H_p . Nevertheless, the W_a model still outperforms the W_n and the original Dual Gaussian models, indicating it is a good approximation for modeling touch pointing.

Our investigation shows that finger movement direction leads to the rotation of the touch point distribution such that the two symmetrical axes of the prediction ellipse approximately fall onto the movement direction and the direction perpendicular to the movement direction. Effects of the movement direction on the orientation of touch points distribution also explain the observation that the x-y correlation ρ of touch points changes with movement angle. When the angle falls into the first or the third quadrant, the x-y correlation of touch points is positive; a movement angle in the second or the fourth quadrant implies a negative correlation between x and y coordinates of touch points. The Rotational Dual Gaussian model is able to capture how ρ will change as the finger movement direction θ changes.

7.4 Limitations of the Rotational Dual Gaussian model

The Rotational Dual Gaussian model works under the assumption that the movement direction θ is known. Cases where θ is unknown are beyond the scope of this model and should be handled by the original Dual Gaussian model. Nevertheless, the Rotational Dual Gaussian model as the latest refinement of the Dual Gaussian family of finger touch models is conceptually more logical because it handles movement amplitude and directional errors separately. As such our first research task was to prove it was indeed more accurate than the original Dual Gaussian models. We also went beyond that by examining its increased power in soft keyboard decoding, which the motor control model is just a part of the total decoding process.

The proposed Rotational Dual Gaussian model was trained for the index finger only, but generalizing it to different fingers is straightforward since the process of deriving the model has no dependence on the input finger type. When generalizing the model to other fingers, the parameters (a , b , c , and d in Equation 3) should be re-estimated with empirical data to account for the different levels of ambiguity caused by different input fingers. Previous research [6] shows that the touch point distributions between index finger and thumb have only small differences. We expect the difference in model parameters of the Rotational Dual Gaussian model for different fingers would also be small. If the input finger is unknown, one possible option is to use the parameters estimated from data aggregated across different input fingers.

8 CONCLUSION

We propose the Rotational Dual Gaussian model, a model that predicts the distribution of touch points in target acquisition tasks. It advances the original Dual Gaussian model [5] by taking into account the finger movement direction θ . The main difference from the original Dual Gaussian model is that the co-variance matrix becomes:

$$\Sigma_R = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{xy} & \sigma_y^2 \end{bmatrix} \quad (26)$$

where

$$\sigma_x^2 = \cos^2 \theta (a + bW_p^2) + \sin^2 \theta (c + dH_p^2)$$

$$\sigma_{xy} = \cos \theta \sin \theta (a + bW_p^2) - \cos \theta \sin \theta (c + dH_p^2)$$

$$\sigma_y^2 = \sin^2 \theta (a + bW_p^2) + \cos^2 \theta (c + dH_p^2)$$

where θ is the finger movement direction, W_p and H_p are projected target width and height along the finger movement direction, and the direction perpendicular to the finger movement direction.

The evaluation on three datasets shows that the Rotational Dual Gaussian model is more accurate than the original Dual Gaussian model. In target acquisition tasks, the Rotational Dual Gaussian model reduces the Root Mean Square Error of error rate prediction from 8.49% to 4.95% over the original Dual Gaussian Model; it also more accurately predicts the correlation coefficient ρ between x and y coordinates of touch points. Using the Rotational Dual Gaussian model also improves the soft keyboard decoding accuracy than using the original Dual Gaussian model for typing on smartwatches, and the difference between using these two models is statistically significant.

ACKNOWLEDGMENTS

We thank anonymous reviewers for their insightful comments, and our user study participants. This work was supported by NSF awards 1805076, 1936027, 2113485, 1815514, and NIH award R01EY030085.

REFERENCES

- [1] 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*. 73–80.

- [2] 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*. 193–200.
- [3] Shiri Azenkot and Shumin Zhai. 2012. Touch behavior with different postures on soft smartphone keyboards. In *Proceedings of the 14th international conference on Human-computer interaction with mobile devices and services*. 251–260.
- [4] Nikola Banovic, Varun Rao, Abinaya Saravanan, Anind K Dey, and Jennifer Mankoff. 2017. Quantifying Aversion to Costly Typing Errors in Expert Mobile Text Entry. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. 4229–4241.
- [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*. 1363–1372.
- [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*. 51–60.
- [7] Xiaojun Bi and Shumin Zhai. 2016. Predicting finger-touch accuracy based on the dual Gaussian distribution model. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. 313–319.
- [8] Bob Carpenter, Andrew Gelman, Matthew Hoffman, Daniel Lee, Ben Goodrich, Michael Betancourt, Marcus Brubaker, Jiqiang Guo, Peter Li, and Allen Riddell. 2017. Stan : A Probabilistic Programming Language. *Journal of Statistical Software* 76 (01 2017). <https://doi.org/10.18637/jss.v076.i01>
- [9] Mark Davies. 2008. The corpus of contemporary American English: 450 million words, 1990–present.
- [10] Mark Davies. 2018. The corpus of contemporary American English: 1990–present.
- [11] Richard A Davis, Keh-Shin Lii, and Dimitris N Politis. 2011. Remarks on some nonparametric estimates of a density function. In *Selected Works of Murray Rosenblatt*. Springer, 95–100.
- [12] Paul M Fitts. 1954. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of experimental psychology* 47, 6 (1954), 381.
- [13] Andrew Fowler, Kurt Partridge, Ciprian Chelba, Xiaojun Bi, Tom Ouyang, and Shumin Zhai. 2015. Effects of language modeling and its personalization on touchscreen typing performance. In *Proceedings of the 33rd annual ACM conference on human factors in computing systems*. 649–658.
- [14] Andrew Gelman, Jessica Hwang, and Aki Vehtari. 2014. Understanding predictive information criteria for Bayesian models. *Statistics and computing* 24, 6 (2014), 997–1016.
- [15] Joshua Goodman, Gina Venolia, Keith Steury, and Chauncey Parker. 2002. Language modeling for soft keyboards. In *Proceedings of the 7th international conference on Intelligent user interfaces*. 194–195.
- [16] Tovi Grossman and Ravin Balakrishnan. 2005. A probabilistic approach to modeling two-dimensional pointing. *ACM Transactions on Computer-Human Interaction (TOCHI)* 12, 3 (2005), 435–459.
- [17] Niels Henze, Enrico Rukzio, and Susanne Boll. 2012. Observational and experimental investigation of typing behaviour using virtual keyboards for mobile devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2659–2668.
- [18] Christian Holz and Patrick Baudisch. 2010. The generalized perceived input point model and how to double touch accuracy by extracting fingerprints. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 581–590.
- [19] Christian Holz and Patrick Baudisch. 2011. Understanding touch. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2501–2510.
- [20] Jin Huang, Feng Tian, Xiangmin Fan, Huawei Tu, Hao Zhang, Xiaolan Peng, and Hongan Wang. 2020. Modeling the Endpoint Uncertainty in Crossing-Based Moving Target Selection. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–12.
- [21] Jin Huang, Feng Tian, Xiangmin Fan, Xiaolong 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*. 1–12.
- [22] Jin Huang, Feng Tian, Nianlong Li, and Xiangmin Fan. 2019. Modeling the Uncertainty in 2D Moving Target Selection. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*. 1031–1043.
- [23] Yu-Jung Ko, Hang Zhao, Yoonsang Kim, IV Ramakrishnan, Shumin Zhai, and Xiaojun Bi. 2020. Modeling Two Dimensional Touch Pointing. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. 858–868.
- [24] Seungyon Lee and Shumin Zhai. 2009. The performance of touch screen soft buttons. In *Proceedings of the SIGCHI conference on human factors in computing systems*. 309–318.
- [25] Yuexing Luo and Daniel Vogel. 2014. Crossing-based selection with direct touch input. In *Proceedings of the SIGCHI conference on human factors in computing systems*. 2627–2636.
- [26] I Scott MacKenzie and R William Soukoreff. 2003. Phrase sets for evaluating text entry techniques. In *CHI'03 extended abstracts on Human factors in computing systems*. 754–755.
- [27] Kanti V Mardia. 1970. Measures of multivariate skewness and kurtosis with applications. *Biometrika* 57, 3 (1970), 519–530.
- [28] David E Meyer, Richard A Abrams, Sylvan Kornblum, Charles E Wright, and JE Keith Smith. 1988. Optimality in human motor performance: ideal control of rapid aimed movements. *Psychological review* 95, 3 (1988), 340.
- [29] Keith Vertanen, Haythem Memmi, Justin Emge, Shyam Reyal, and Per Ola Kristensson. 2015. VelociTap: Investigating fast mobile text entry using sentence-based decoding of touchscreen keyboard input. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. 659–668.
- [30] Daniel Vogel and Ravin Balakrishnan. 2010. Occlusion-aware interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 263–272.
- [31] Daniel Vogel and Patrick Baudisch. 2007. Shift: a technique for operating pen-based interfaces using touch. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 657–666.
- [32] Daniel Vogel and G ry Casiez. 2012. Hand occlusion on a multi-touch tabletop. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2307–2316.
- [33] Feng Wang and Xiangshi Ren. 2009. Empirical evaluation for finger input properties in multi-touch interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 1063–1072.
- [34] Daryl Weir, Henning Pohl, Simon Rogers, Keith Vertanen, and Per Ola Kristensson. 2014. Uncertain text entry on mobile devices. In *Proceedings of the SIGCHI conference on human factors in computing systems*. 2307–2316.
- [35] 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*. 1613–1622.
- [36] Jacob O Wobbrock, Alex Jansen, and Kristen Shimohara. 2011. Modeling and predicting pointing errors in two dimensions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 1653–1656.
- [37] Jacob O Wobbrock and Brad A Myers. 2006. Analyzing the input stream for character-level errors in unconstrained text entry evaluations. *ACM Transactions on Computer-Human Interaction (TOCHI)* 13, 4 (2006), 458–489.
- [38] Shota Yamanaka and Hiroki Usaba. 2020. Rethinking the Dual Gaussian Distribution Model for Predicting Touch Accuracy in On-screen-start Pointing Tasks. *Proceedings of the ACM on Human-Computer Interaction* 4, ISS (2020), 1–20.
- [39] 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 (2004), 823–856.