KeyTime: Super-Accurate Prediction of Stroke Gesture Production Times

Luis A. Leiva[1] **Daniel Martín-Albo**[2] **Réjean Plamondon**[3] **Radu-Daniel Vatavu**[4]

[1]Sciling, Spain [2]WIRIS Math, Spain [3]École Polytechnique de Montréal, Canada

[4]MintViz Lab, MANSiD Research Center, University Ştefan cel Mare of Suceava, Romania

name@sciling.com dmas@wiris.com rejean.plamondon@polymtl.ca vatavu@eed.usv.ro

ABSTRACT

We introduce KeyTime, a new technique and accompanying software for predicting the production times of users' stroke gestures articulated on touchscreens. KeyTime employs the principles and concepts of the Kinematic Theory, such as lognormal modeling of stroke gestures' velocity profiles, to estimate gesture production times significantly more accurately than existing approaches. Our experimental results obtained on several public datasets show that KeyTime predicts user-independent production times that correlate $r = .99$ with groundtruth from just one example of a gesture articulation, while delivering an average error in the predicted time magnitude that is 3 to 6 times smaller than that delivered by CLC, the best prediction technique up to date. Moreover, KeyTime reports a wide range of useful statistics, such as the trimmed mean, median, standard deviation, and confidence intervals, providing practitioners with unprecedented levels of accuracy and sophistication to characterize their users' a priori time performance with stroke gesture input.

ACM Classification Keywords

H.5.2 Information Interfaces and Presentation: User Interfaces; I.5.2 Pattern Recognition: Design Methodology

Author Keywords

Human Performance; Gesture Synthesis; Kinematic Theory; Rapid Prototyping; Touch Gestures; Stroke Gestures

INTRODUCTION

Stroke gestures, such as those produced on touchscreens, have gained considerable interest in the research and practice of user interface design as excellent shortcuts for specific functions or sequences thereof [6,21,38,52]; e.g., a letter "B" drawn on the home screen of a smartphone could be mapped to a tedious sequence of commands, such as "go to `Settings` → `Battery and Performance` → `Power Settings` and toggle the `Balanced` option." Compared to traditional input techniques based on item selection from menus, stroke gestures are not only faster, but they also reduce users' cognitive load and

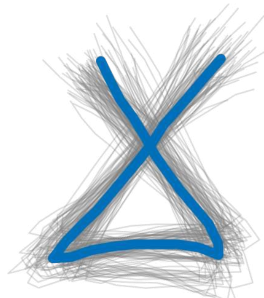

Figure 1. KeyTime predicts users' performance for a given gesture by computing a representative, user-independent prototype of that gesture (thick line) using synthetic articulations (thin gray lines) automatically derived from a single example provided by the designer. Additionally, unlike other techniques, KeyTime delivers a wide palette of predictors for production times, such as variances and confidence intervals.

visual attention [6,53] and increase usability [21,24]. Moreover, stroke gestures represent the only effective input technique for some user categories to operate touchscreen devices effectively, such as users with visual impairments [17]. In fact, popular screen readers, such as VoiceOver and Google TalkBack, already employ a large variety of such stroke gesture commands, e.g., a two-finger flick down reads the screen from the VoiceOver cursor to the end of the current page.[1]

The production time of a stroke gesture, i.e., how long it takes users, on average, to produce a 2D handwritten trace on a touch-sensitive surface, is one essential aspect of user performance with gesture input [9,11,37,47]. Such insightful information about users' performance represents a valuable asset for practitioners to inform gesture design directly, e.g., what are the fastest gestures to produce [6,11] or indirectly, e.g., what are the easiest gestures to execute from a given set [37,47]. In fact, the research literature has repeatedly stressed the importance of designing gesture commands that are easy to understand, learn, and recall and, consequently, fast to articulate [6,11,19,38,53]. Moreover, gesture production time turned out to be an excellent predictor of users' subjective perceptions of the difficulty to articulate stroke gestures [37, 47]. In this context, it is important for user interface designers to be able to estimate a priori, as accurately as possible, users' input performance in order to save considerable time and effort demanded by subsequent user evaluations and/or gesture set redesigns.

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

ISBN 978-1-4503-5620-6/18/04…$15.00

DOI: https://doi.org/10.1145/3173574.3173813

[1]`https://apple.com/voiceover/info/guide/1131.html`

The problem of predicting gesture production times has been addressed in the community with various techniques, from simple estimation rules [16] and training procedures [47] to complex models of the geometry of stroke gesture paths [9]. However, today's best technique to predict the production times of stroke gestures, known as the Curves, Lines, and Corners (CLC) model [9], was derived more than 10 years ago when the state of the art in stroke gesture processing, analysis, and recognition was not as mature as it is today [3,21,36,39,40, 43,44] and, unavoidably, came with important limitations in terms of flexibility and accuracy [9,11]. For instance, CLC can only provide a single prediction value,[2] which is insufficient to characterize the variation in gesture articulation within and between users [3,43,44] (low flexibility). Also, although being accurate at providing a relative ranking of gestures according to their production times [9], CLC is known to overestimate the actual magnitudes of predicted times [9,11,47] (low accuracy). Therefore, better models of gesture articulation in the time domain are needed to assist designers with accurate predictions of users' time performance with gesture input on touchscreens.

In this paper, we introduce KeyTime, a new technique that relies on the principles of the Kinematic Theory [31,32,33] to predict the production times of stroke gestures. In its simplest form, KeyTime delivers a very accurate prediction of the user-independent production time of a given gesture type, e.g., the expected time of the gesture shown in Figure 1 is 655 ms, which is off by just 55 ms from the actual measured time compared to CLC's estimation of $1{,}793$ ms. In its true strength, however, KeyTime computes a variety of predictors for production times: mean, median, trimmed and winsorized means, variance, standard deviation, standard error, and confidence intervals that, together, deliver a rich and sophisticated characterization of users' time performance with stroke gesture input; e.g., the 95% confidence interval predicted by KeyTime for the gesture shown in Figure 1 is $[634, 677]$ ms.

A practical example

In the following, we show with a straightforward example the advantage of estimating gesture production times not with a single value, but rather with ranges of variation, expressed in this case with the standard deviation and 95% confidence interval. Imagine a designer who wants to decide which of three gestures, A, B, or C, to use for popping up a contextual menu in some graphical interface. Knowing that the task will be executed frequently by users, the associated gesture command should be "fast and easy to perform." By employing a production time estimator such as CLC [9], the designer finds that gesture A will take approximately 972 ms to perform, gesture B will take $1{,}090$ ms, while C will be executed in about $2{,}886$ ms.[3] Based on this information, the designer decides to assign gesture A to the high-priority contextual menu function and use gesture B for another, less frequent task. Given that A is faster than B, A is also going to be perceived by users easier to execute than B according to the first rule of estimating gesture difficulty of Vatavu et al. [47] (p.101), so the designer can rest assured that both design criteria are met. The application is released and, surprisingly, feedback reports start coming in that users struggle with the contextual menu feature. The designer looks at the application logs and notices that gesture A not only is performed slower than B but also with larger variability (SD $= 925$ ms, $\mathrm{CI}_{95\%} = [1202, 1402]$ ms), while gesture B has narrower variation (SD $= 606$ ms, $\mathrm{CI}_{95\%} = [886, 1017]$ ms). In fact, users are much more precise at executing gesture B than A, an outcome that unfortunately impacted negatively the usability of the interface. In this case, the designer did not dispose of sufficient information and, consequently, adopted a suboptimal decision. Other similar examples can be easily imagined by the reader, such as the time predictor being not accurate enough [11] or its parameters needing to be tweaked to specific users to maximize prediction accuracy [9] (p.1502).

[2] We will use the terms "prediction" and "estimation" interchangeably in this work to denote "an approximation of a result."

[3] Actual values computed on the GDS dataset [52] (A: "triangle", B: "pigtail", C: "star"), which we analyze later in this paper.

Contributions

To enable effective gesture designs, the production time of a gesture needs to be accurately estimated. This process, however, requires computing both measures of central tendency (i.e., the *expected* production time of a gesture) and, equally important, measures of variation as well; i.e., how much are users expected to deviate their production times from the mean? Given that users are known to vary their gesture articulations [3,44], it also causes variation in their production times. Thus, the mean time taken alone cannot provide a realistic picture of the expected user performance. As shown before in the literature, the CLC model can be far off [11] and, unfortunately, the HCI community has not been offered yet a more accurate estimator. Our new technique is not only far more accurate than the state of the art [9,16], but the extra information given by location and dispersion-based measures can tell the practitioner the range in which the mean time is likely to lie and also how much to expect individual times to deviate from the mean. Therefore, with this work, we touch a key aspect by showing that the problem of time estimation is not a punctual one (i.e., a single value), but it should rather be approached from the perspective of variation, for which the "interval" (e.g., 95% or 99% CIs, ±SDs, etc.) is the appropriate concept to employ.

Our contributions in this work are as follows:

1. We introduce KeyTime, a new technique grounded on the solid foundation of the Kinematic Theory [31,32,33] to compute accurate predictions of stroke gesture production times, articulated e.g. on a touchscreen.
2. We present an evaluation of KeyTime on three public datasets ($14{,}240$ unistroke gestures collected from 35 participants), and we show that KeyTime outperforms state-of-the-art techniques [9,16] both in terms of relative and absolute prediction accuracy.
3. We also show how KeyTime can be used to compute a wide range of useful predictors of location (e.g., mean, median, 95% and 99% confidence intervals) and dispersion (e.g., variance and standard deviation), one of the unique features of KeyTime, unmatched by and unattainable with any of the techniques before it.

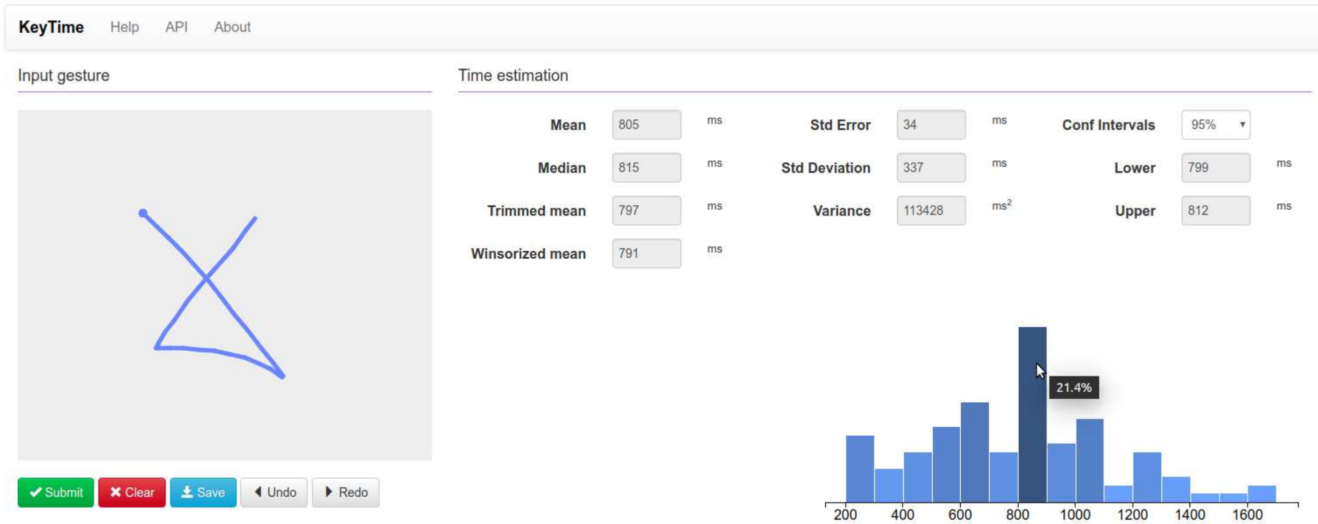

Figure 2. The KeyTime web application reports a wide range of statistics to characterize user-independent stroke gesture production times from just one provided example. The application is available at `https://luis.leiva.name/keytime/`.

Predicting production times with KeyTime

Before proceeding further, we take a brief moment to illustrate how KeyTime operates and how easy it is for designers and researchers to access a wide range of time statistics regarding users' a priori gesture input performance. Suppose that a designer wishes to implement a "delete" command with the gesture depicted in Figure 1 (which is actually the "delete" gesture of the GDS dataset [52] that we borrow here to illustrate our example). The designer wants to know how fast users will be able to produce this gesture. Normally, the task would be challenging, because of the within and between-users variation in producing gestures, as documented in the literature [3,43,44]. However, KeyTime turns the prediction task into a simple, one-click procedure (see Figure 2), as follows:

a. The designer draws the "delete" gesture in free form *once*, using the KeyTime web interface and clicks on the 'Submit' button.
b. KeyTime automatically computes a wide range of statistics that characterize the user-independent time performance expected for the provided gesture type; see Figure 2.

The results are available in the web application, as shown in Figure 2 or, for custom setups, they can be queried using KeyTime's JSON RESTful API, as shown in Figure 3.

```
HTTP/1.1 200 OK
Connection: close
```

```
{
    "errors": null,
    "result": {
        "confidence_intervals": {
            "99%": [796, 814],
            "95%": [799, 812],
            "90%": [800, 811]
        },
        "max": 1776,
        "mean": 805
        "median": 815,
        "min": 129,
        "range": 1647,
        "standard_deviation": 337,
        "standard_error": 34,
        "trimmed_mean": 797,
        "values": [884, 841, 626, ...],
        "variance": 113428,
        "winsorized_mean": 791,
    }
}
```


Figure 3. KeyTime API response for the gesture shown in Figure 2.

RELATED WORK

We review in this section prior work that examined user performance with stroke gesture input. We also discuss applications of this prior work to gesture recognition, synthesis, and analysis, highlighting the importance of production time as a key feature for gesture-based user interfaces.

Evaluating users' stroke gesture input performance

Researchers have employed a variety of measures to characterize users' performance with stroke gesture input. For example, Blagojevic et al. [8] examined 114 distinct gesture features to inform the design of an accurate feature-based statistical recognizer. Other researchers looked for representative features to depict various aspects of users' performance. For example, Anthony et al. [3] evaluated gesture articulation consistency, and showed that users are highly consistent (evaluations within-users showed .91 on a scale of 0 to 1), but also highly individual (between-users consistency was .55). The study also reported a log-linear relationship with the number of strokes: less consistency was observed for gestures that were produced with more strokes.

Gesture features and measures were also used to inform the design of gesture sets. For example, Long et al. [25] were interested in gesture shapes that would be easy for users to learn and recall. They found that users' perceptions of gestures' visual similarity were related to several gesture features (such as gesture length, area, or various angles), and derived a model for perceived gesture similarity that correlated $r = .56$ with groundtruth.

Researchers have also employed gesture measures to understand differences in users' input performance between various user categories or input conditions. For example, Vatavu et al. [46] employed "path accuracy" measurements to quantify deviations from ideal straight-line paths for drag-and-drop gestures produced by small children on touchscreens. Kane et al. [18] examined specific gesture features, such as "location accuracy" or "line steadiness" to contrast touch gestures produced by people with and without visual impairments. Tu et al. [41] also introduced various specialized geometric and kinematic features, such as "axial symmetry" or "intersecting points deviation," to understand the differences between gestures produced with pen and finger.

Interested in a more sophisticated characterization of the amount of variation in articulation along the gesture path, Vatavu et al. [44] introduced a set of relative accuracy measures of performance to describe the geometric, kinematic, and articulation accuracy of stroke gestures with respect to canonical gesture templates, such as those present in recognizers' training sets. A follow-up work introduced "gesture heatmaps" as colorful visualizations of stroke gesture paths to facilitate understanding of recognition errors and to highlight differences in articulation under various conditions [45].

Such gesture measures and features introduced by prior work have proven very useful to characterize various aspects of user performance with stroke gesture input, as well as to inform gesture user interface design. However, another line of work has focused on a more fundamental understanding of human movement during stroke gesture production by recurring to key aspects from the motor control theory. We discuss this work in the following section.

Time performance models for stroke gesture input

Simple forms of stroke-based input, such as pointing and item selection from menus, have been extensively studied with Fitts' law and its variations [7,13,51], the steering law [1], or the Keystroke-Level Model (KLM) [10]. However, more complex stroke input techniques, such as handwriting or free-form gesture paths drawn on touchscreens, need more sophisticated models to characterize human performance effectively. Comprehensive surveys in this area are provided by Quinn and Zhai [36] and Müller et al. [29].

Isokoski [16] proposed a first-order rank model for stroke gestures that used the number of approximating line segments as a predictor of that gesture's shape complexity. Although Isokoski's model did not attempt to quantify production time explicitly, it was nevertheless found to predict the relative ranking of gestures by their production times with reasonable accuracy [16]. In contrast, Cao and Zhai's CLC model [9] was specifically designed to predict the actual magnitudes of stroke gesture production times. The CLC model operates by dividing the gesture shape into curves, straight lines, and corners, for which production times are estimated individually. The predicted time for the gesture is computed as the sum of the individual production times needed to articulate each of the gesture's elementary parts. The CLC model works very well as a first-order predictor, but it tends to overestimate production times [11,47], presumably because of its inability to compensate for users' articulation skills [9]. Nevertheless, CLC represents the state of the art in predicting stroke gesture production times.

Other gesture models have addressed specific application domains for stroke gesture input, such as text entry. For example, Quinn and Zhai [36] developed a model of gesture production that can predict realistic gesture trajectories for arbitrary shapewriting tasks. The model employs "statistical via-points" located in each key traveled by the finger with distributions that reflect the sensorimotor noise and speed-accuracy tradeoff while typing. However, Quinn and Zhai's model assumes interaction with a keyboard layout and does not predict absolute movement time [29].

Viviani et al. [48,49] were among the first researchers to investigate the fundamentals of human handwriting and drawing behavior, which led to the $2/3$ power law of curvature that connected the speed of articulation with the curvature of the produced stroke. An interesting line of research has been the application of minimization principles to motor control, among which Flash and Hogan's minimum-jerk theory [14] has become particularly influential in the HCI literature [36]. This theory argued that human trajectories chosen by the motor system converge toward path smoothness, and can be used to successfully predict the bell-shaped velocity profiles of articulated strokes observed experimentally. Further investigations of human movements showed that lognormal-based models, such as those postulated by the Kinematic Theory [31,32,33], are arguably the most accurate descriptors of human movements known today, compared to which "other models can be considered as successive approximations" [12].

The Kinematic Theory is strongly supported mathematically by an extension of the Central Limit Theorem [35], and has been extensively verified experimentally [30]. In the context set by this theory, gestures are planned in advance in terms of their spatial organization described by a map of "virtual targets." This map is activated as a sequence of commands and the human peripheral system reacts to these commands. The overall gesture trajectory is the result of the time superimposition of the different velocity profiles, and the articulation fluency is directly linked to the quality of this superimposition.

The Kinematic Theory has recently found many applications to stroke gesture input. For example, the "Gestures à Go Go" (G3) application [22,26] was introduced to synthesize stroke gestures from just a single example provided by the designer. Leiva et al. [22,23,24] evaluated the articulation characteristics of synthetic stroke gestures under various conditions, such as pen vs. finger input, slow vs. fast speed, or for various user categories. They showed that synthetic gestures possess the same characteristics as gestures produced by actual users [20,23], with direct application to training gesture recognizers efficiently and to synthesize stroke gestures across user categories [24]. In this work, we rely on the concepts and principles of the Kinematic Theory to introduce the KeyTime technique.

Summary
Next to recognition accuracy, the production time of a gesture represents the essential aspect that determines human performance *directly*: fast gestures make for ideal shortcuts [6]. The overall goal of every user interface designer is to optimize input efficiency, which has been traditionally evaluated with error rates and task times. For the specific domain of stroke gestures, however, production time represents an instance of "task time" and, consequently, stands as a direct measure of input efficiency. Moreover, production time also describes other aspects of user performance with stroke gesture input, *indirectly*. For example, Rekik et al. [37] and Vatavu et al. [47] found that production times correlate very strongly ($r \geq .95$) with users' perceptions of the difficulty to articulate gestures.

In this context, it is important for designers to be able to estimate users' stroke gesture time performance a priori to inform their gesture set designs. In this work, we show that existing models [9,16] are neither accurate or precise and, consequently, new techniques are needed to the predict gesture production times reliably. KeyTime comes to address this need by delivering very accurate, rich, user-independent time predictions of users' performance with stroke gesture input.

KEYTIME
We introduce in this section our new technique for predicting the production times of stroke gestures by relying on the theoretical concepts and practical principles of the Kinematic Theory. KeyTime employs an internal model, i.e., the Sigma-Lognormal model ($\Sigma\Lambda$) of the Kinematic Theory [31,32], to describe users' gesture articulations in the time domain as an optimum set of lognormal-based velocity profiles, and then uses the model to synthesize as many articulation variations as possible for a given gesture type, reflective of the actual articulations of the users [22,23]. The production times of all synthesized gestures are then compiled into an accurate, user-independent estimate of that gesture's production time. Before we describe the KeyTime technique, we review the core concepts of the Kinematic Theory, such as computing the $\Sigma\Lambda$ model from one single articulation sample [22], the concept of virtual targets [31], and computation of the model parameters, referred to as the *central* and *peripheral* parameters [31,32].

Technical overview of the Kinematic Theory
The Kinematic Theory is a solid framework for studying human movement production, which has been recently adopted in HCI for stroke gesture synthesis and recognition [22,23,24]. The latest instantiation of this framework is the Sigma-Lognormal model [34], which was demonstrated to outperform many other models [12,33].

The Kinematic Theory assumes that a complex handwritten trace, e.g., a character, word, signature, or stroke gesture, is composed of a series of primitives[4] connecting a sequence of "virtual targets", such as those illustrated in Figure 4. The virtual targets correspond to near-zero-velocity peaks in the gesture strokes and are automatically computed by the $\Sigma\Lambda$ model [31,32]. These primitives form the "action plan" of the user in relation to a specific gesture type which, by means of the neuromuscular network, will produce a path trajectory in the form of a handwritten trace on the touch-sensing surface.

[4]The Kinematic Theory uses the term "stroke" to denote what we call "primitive" in this paper. In HCI, we refer to a gesture stroke as the sequence of points bewteen two consecutive pen-down and pen-up events.

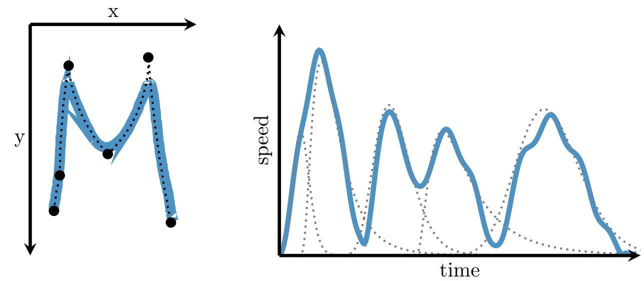

Figure 4. Left: A gesture stroke (thick line) is described by a series of primitives (dotted arcs) that connect virtual targets (black dots). Right: primitives are described in terms of their lognormal velocity profiles.

The $\Sigma\Lambda$ model computes the velocity profile of each primitive ($\vec{v}_i$) according to the lognormal function illustrated below, which is defined by a set of central (D,t_0,θ) and peripheral (μ,σ) parameters of $\Sigma\Lambda$ [31]:

$$\begin{aligned}\|\vec{v}_i(t)\| &= D_i \Lambda(t; t_{0_i}, \mu_i, \sigma_i^2) \\ &= \frac{D_i}{\sigma_i\sqrt{2\pi}(t - t_{0_i})} \exp\left(\frac{-[\ln(t - t_{0_i}) - \mu_i]^2}{2\sigma_i^2}\right)\end{aligned} \tag{1}$$

$$\vec{v}(t) = \sum_{i=1}^{N} \vec{v}_i(t) = \sum_{i=1}^{N} \begin{bmatrix}\cos\phi_i(t) \\ \sin\phi_i(t)\end{bmatrix} D_i \Lambda(t; t_{0_i}, \mu_i, \sigma_i^2) \tag{2}$$

$$\phi_i(t) = \theta_{s_i} + \frac{\theta_{e_i} - \theta_{s_i}}{2}\left[1 + \text{erf}\left(\frac{\ln(t - t_{0_i}) - \mu_i}{\sigma_i\sqrt{2}}\right)\right] \tag{3}$$

We employ the $\Sigma\Lambda$ extractor from Martín-Albo et al. [28] to compute the parameter values that best fit the observed velocity profiles. Once the gesture primitives are modeled, perturbations can be added to the model parameters in order to produce different gesture variations, as follows:

$$p_i^* = p_i + n_{p_i} \tag{4}$$

where $p_i = \{\mu_i, \sigma_i, D_i, \theta_{s_i}, \theta_{e_i}\}$ denote the $\Sigma\Lambda$ parameters and $n_{p_i} = \mathcal{U}(-n_i, n_i)$ the noise applied to each primitive according to a uniform distribution centered around that particular $\Sigma\Lambda$ parameter [23]. For example, perturbations in μ and σ mimic peripheral noise, e.g., a user who articulates the same gesture slightly different each time; perturbations in D and θ refer to central fluctuations that occur in the position of the virtual targets of the action plan from one articulation to the next.

For more details, we refer the reader interested in the $\Sigma\Lambda$ mathematical formulation to Plamondon et al. [31,32,34] and to Leiva et al. [22,23,24] and Martín-Albo et al. [27,28] for applications to stroke gesture input and handwriting analysis, respectively. These works also include diverse studies on noise variation and how it impacts recognition performance.

The KeyTime technique

KeyTime provides a prediction ($\hat{t}$) of the expected production time (t) of a given gesture type g, e.g., 1,250 ms for a "square" gesture, starting from one sample of g that can be provided by the designers themselves. The gesture sample is used to synthesize many different potential articulations of g as in Leiva et al. [22], each articulation having different yet realistic production times. This technique was demonstrated to produce human-like synthetic gestures that possess the same articulation characteristics as actual gestures produced by real users, with comparable variations in path length, area size, production time, and articulation speed [20,23].

Following the principles of the Kinematic Theory, KeyTime constructs a model for gesture g as the summation of its stroke primitives [31]; see Figure 4 on the previous page. The ending time of a single primitive is given by:

$$t_{e_i} = t_{0_i} + \exp(\mu_i + 3\sigma_i) \tag{5}$$

At this time, 99.97% of the trajectory distance pertaining to the gesture stroke has been already covered by the ith primitive. Then, the production time of the modeled gesture is $t_{e_L} - t_{0_1}$, where L denotes the last primitive.

Let n be different variations generated by KeyTime for gesture g, for which the corresponding production times computed with the approach described above are t_i, $i = 1..n$. Starting from these values, KeyTime computes a prediction of the expected production time of g as follows:

$$\hat{t} = \mathcal{F}(t_1, t_2, \ldots, t_n) \tag{6}$$

where $\mathcal{F}$ is a positive real-valued function. In this work, we implement and evaluate the following variants for $\mathcal{F}$:

1. The arithmetic **mean** ($\hat{t}_{\mathrm{M}}$) of the arguments represents the average production time of all the synthesized versions of gesture g:

$$\hat{t}_{\mathrm{M}} = \frac{1}{n}\sum_{i=1}^{n} t_i \tag{7}$$

2. The **median** ($\hat{t}_{\mathrm{Mdn}}$) is the middle value $i = \frac{n+1}{2}$, assuming all values are sorted, or, equivalently, the second quartile of t_i, $i = 1..n$. Unlike the mean, however, the median is much less affected by outliers and, consequently, it should provide a better estimate of the expected production time given t_i.

3. The **20%-trimmed mean** ($\hat{t}_{.20}$) is computed by sorting the values of the arguments t_i, removing the lowest 20% as well as the highest 20%, and then averaging the values that remain in the center, i.e., 60% of the data. The 20%-trimmed mean represents a robust measure of location [50] especially when distributions are skewed. Although various trimming percents can be applied (e.g., 5%, 10%, etc.), a 20% trimming is generally considered to deliver a considerable advantage over no trimming at all (i.e., the mean) or over the median (50% trimming) [50]. We compute the 20%-trimmed mean as follows:

$$\hat{t}_{.20} = \frac{1}{n}\sum_{i=[0.2n]}^{[0.8n]} t_{\sigma(i)} \tag{8}$$

where $t_{\sigma(i)}$ denotes the ascending ordered list of the t_i values and $[\cdot]$ is the integer part function.

4. The **winsorized mean** ($\hat{t}_{\mathrm{W}}$) replaces the lowest and highest p% of the t_i values with the smallest and highest non-trimmed value, i.e., the smallest p% of the observations are pulled up to the smallest t_i value coming right next, while the largest p% are pulled down to the largest value just before them. We use $p = 20\%$ for the same reasons presented previously. The winsorized mean is the average of the winsorized values:

$$\hat{t}_{\mathrm{W}} = \frac{1}{n}\sum_{i=1}^{n} t'_{\sigma(i)} \tag{9}$$

where t'_i equals t_i for values located in the center (60% of the values), $t_{[0.2n]}$ for the lowest 20% of the values ($i < [0.2n]$), and $t_{[0.8n]}$ for the highest 20% ($i > [0.8n]$).

Table 1 shows a calculation example of these four measures. While the arithmetic mean employs all the values to produce an estimate of central tendency, the 20%-trimmed mean discards the values in gray (12, 28 and 90, 192, respectively), and the winsorized mean replaces those values with the next eligible ones (40 and 87), highlighted in orange. This example illustrates how various measures of central tendency are more or less representative of the underlying "true" distribution.

Sample time values t_i ($i = 1..10$)	$\hat{t}_{\mathrm{M}}$	$\hat{t}_{\mathrm{Mdn}}$	$\hat{t}_{.20}$	$\hat{t}_{\mathrm{W}}$
12, 28, 40, 48, 48, 58, 85, 87, 90, 192	68.8	53.0	61.0	62.0

Table 1. The mean ($\hat{t}_{\mathrm{M}}$), median ($\hat{t}_{\mathrm{Mdn}}$), 20%-trimmed mean ($\hat{t}_{.20}$), and the winsorized mean ($\hat{t}_{\mathrm{W}}$) of a set of 10 values.

In this work, we use $n = 100$ for the number of synthetic gestures and corresponding production time values t_i, $i = 1..n$, that KeyTime synthesizes to predict the production time of a given gesture type. Although in theory n could be chosen as large as desired, previous work has shown that values of $n > 100$ do not necessarily contribute to substantial variation in the pool of synthetic gestures that are being generated with the $\Sigma\Lambda$ model; see Leiva et al. [22] (p.19).

EVALUATION

We conducted an experiment to evaluate the accuracy of KeyTime for predicting the production time of stroke gestures.

Experiment design

We manipulated one independent variable, TIME-PREDICTOR, with three levels: (1) KeyTime, our new technique, (2) CLC, the best competitor from the literature [9], and (3) groundtruth, the control condition, represented by the actual production times of gestures articulated by users. The performance of the time predictors was evaluated with the following accuracy measures, acting as dependent variables in our experiment:

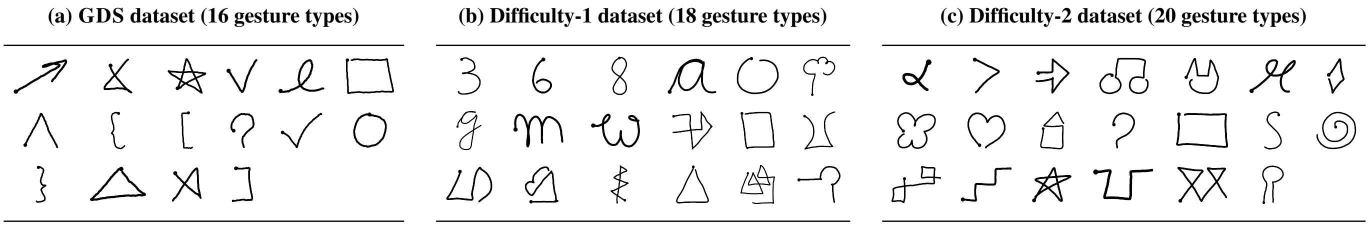

Figure 5. Gesture types (actual examples) from the GDS, Difficulty-1, and Difficulty-2 datasets [47,52].

1. RANKING-ACCURACY (or RELATIVE-ACCURACY) evaluates the extent to which a TIME-PREDICTOR is able to deliver the correct ranking of gestures according to their production times. For example, if the mean production times of the "square" and "arrow" gesture types are 1,778 ms and 1,900 ms, respectively, and their predicted production times also respect this relative order, i.e., $\hat{t}_{\text{square}} < \hat{t}_{\text{arrow}}$, then the relative prediction is accurate. For the general case with more than two gestures, the ranking accuracy can be directly evaluated against groundtruth times using Spearman's rank correlation coefficient r_s. The closer r_s to 1, the more accurate the TIME-PREDICTOR is for reporting the relative order of gesture production times.
2. ABSOLUTE-ACCURACY evaluates the extent to which a TIME-PREDICTOR delivers the correct magnitude of the expected production time of a given gesture type. The closer the predicted time to the groundtruth time, the more accurate the TIME-PREDICTOR is.

Datasets

We employed the following publicly available gesture datasets:

1. The **GDS** dataset [52] contains 4,800 samples of 16 distinct gesture types (see Figure 5a) performed by ten participants with a stylus on an iPAQ Pocket PC. Because participants were asked to articulate gestures at three different speeds (slow, medium, and fast), we predicted production times separately for each articulation speed, which corresponds to using three sub-datasets in our analysis:

   1.1. **GDS-fast**: 1,600 gestures performed by ten participants at fast speed (10 executions per participant per gesture type). Participants received the instruction "go as fast as you can."

   1.2. **GDS-medium**: 1,600 gestures performed at medium speed by the same participants (10 executions per participant per gesture type). Participants received the instruction "balance speed and accuracy."

   1.3. **GDS-slow**: 1,600 gestures performed at slow speed by the same participants (10 executions per participant per gesture type). Participants received the instruction "be as accurate as possible."

2. The **Difficulty-1** dataset [47] contains 5,040 samples of 18 distinct gesture types (Figure 5b) performed by 14 participants with a stylus on a Wacom DTU-710 display (20 executions per participant per gesture type).
3. The **Difficulty-2** dataset [47] contains 4,400 samples of 20 distinct gesture types (Figure 5c) performed by 11 participants with a stylus on a Wacom DTU-710 display (20 executions per participant per gesture type).

In total, we evaluate the prediction performance of KeyTime on 14,240 samples of 48 distinct gesture types collected under various conditions [47,52] from 35 participants. These gesture types represent a good mixture of geometrical shapes, letters, digits, and symbols with a large variety and wide range of complexity levels (assessed using Isokoski's shape complexity measure [16] between 2 for the "check" [52] and "right arrow" [47] gestures up to 10 for the "triangles chain" symbol [47]), and a good balance between familiar (i.e., known and practiced) and non-familiar (i.e., first time seen) symbols [47].

Methodology

CLC production times were generated with the PlayCLC application[5] provided as companion to the CLC paper [9]. PlayCLC enables the designer to create gesture models as a sequence of lines and curves, from which it automatically computes a prediction of that gesture's production time.

In our own testing, we observed that results vary with the designer's ability to draw curves. To address this aspect, three researchers used PlayCLC to create models for the gestures in Figure 5 and the resulted production times were averaged for each gesture type.

foreach gesture $g \in [1..G]$ **do**
    foreach participant $p \in [1..P]$ **do**
        foreach execution $e \in [1..E]$ **do**
            $\mathcal{T} \leftarrow$ generate N samples of type g from e using $\Sigma\Lambda$
            $\hat{t} \leftarrow$ AVERAGE($\mathcal{T}$)
            $t_{\text{true}} \leftarrow t_e \in \{P \cap p\}$

Figure 6. The user-independent, leave-one-out cross-validation procedure employed in our evaluation experiment. The function AVERAGE from this pseudocode implements any variant of KeyTime measures.

We computed production times for KeyTime using the leave-one-out cross-validation procedure depicted in Figure 6, which considers each execution e from each gesture g produced by each participant p as the representative gesture sample from which the $\Sigma\Lambda$ model parameters are computed. Notice also that the true time is not known by KeyTime.

[5] `http://www.cs.toronto.edu/~caox/PlayCLC/PlayCLC.htm` (now defunct, but accessible via `http://web.archive.org`).

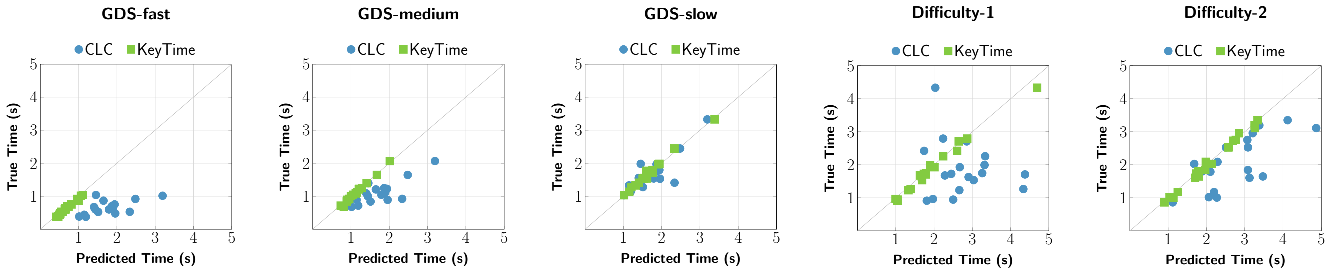

Figure 7. Predicted production times for each gesture from the evaluation datasets. In these figures, we show the arithmetic mean ($\hat{t}_{\mathrm{M}}$) for KeyTime.

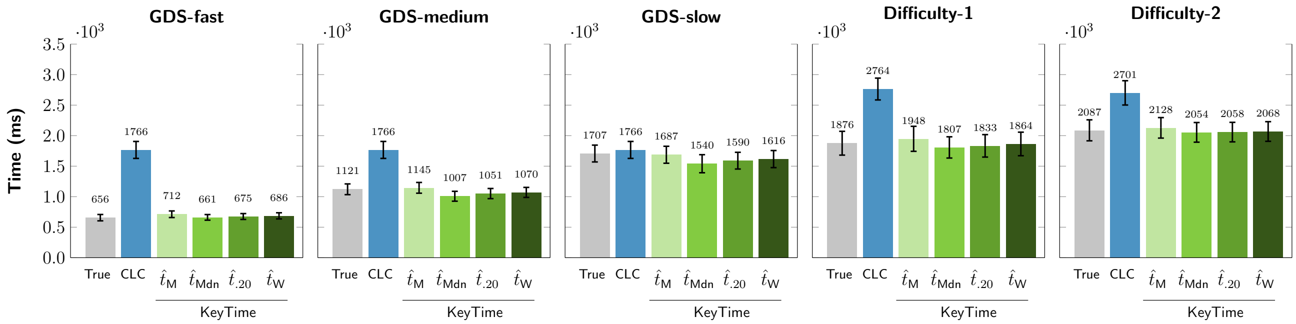

Figure 8. Predicted production times delivered by CLC and KeyTime vs. groundtruth (True). Error bars denote the standard error of the mean.

RESULTS

We start our analysis of the results by discussing the relative accuracy of production times delivered by each TIME-ESTIMATOR. Then, we analyze the accuracy of the absolute magnitudes of predicted times, which is a much stronger performance criterion.

Ranking accuracy: the relative order of production times

Figure 7 plots predicted times delivered by KeyTime and CLC vs. groundtruth for each dataset. It is easy to see that CLC values tend to accumulate below the diagonal line, a result that confirms previous observations from the literature about CLC overestimating the magnitude of predicted times [9,11]. KeyTime, however, seems to deliver predictions that are very close to the diagonal line, which in turn represents the performance of an ideal time predictor. We verified these initial observations with Spearman correlations between CLC, Isokoski's shape complexity measure, and KeyTime, respectively, against groundtruth data; see Table 2. While the correlation coefficients for CLC reached a maximum of .620 ($p < .01$), all coefficients for KeyTime's $\hat{t}$ predictor were above .979 ($p < .001$) with a maximum of .992 for the GDS-medium dataset. Fisher tests revealed statistically significant differences between CLC and KeyTime correlation coefficients for all datasets ($p < .001$).

Absolute accuracy: the magnitude of production times

Figure 8 illustrates the magnitudes of production times predicted by KeyTime and CLC, respectively, vs. groundtruth for each evaluation dataset. Overall, KeyTime delivered time predictions that were much closer to the actual production times than the predictions delivered by CLC; e.g. for a groundtruth average of 1,876 ms in the Difficulty-1 dataset KeyTime provided an estimation of $\hat{t}_{\mathrm{W}} = 1{,}864$ ms, whereas CLC estimated 2,764 ms. As noted before, CLC predictions tended to overestimate the magnitude of production times, except for the GDS-slow dataset, for which participants were actually instructed to spend more time to articulate gestures as accurately as possible. KeyTime predictions were very close to the groundtruth for all datasets, with average error offsets from as low as 5 ms reached by $\hat{t}_{\mathrm{M}}$ for the GDS-fast dataset.

A one-way ANOVA procedure showed a statistically significant effect of TIME-ESTIMATOR for four out of our five datasets; see Table 3. Post-hoc pairwise t-tests (Bonferroni corrected) revealed better performance of all implementations of KeyTime ($\hat{t}_{\mathrm{M}}$, $\hat{t}_{\mathrm{Mdn}}$, $\hat{t}_{.20}$, $\hat{t}_{\mathrm{W}}$) compared to CLC for all datasets ($p < .05$) excepting GDS-slow, in which case all time estimators performed equally ($p > .05$). In any case, we found no statistically significant differences between KeyTime and true times, which builds our confidence that KeyTime does not only produce more accurate time predictions than CLC, but that KeyTime's predictions are also on par with users' actual time performance with stroke gesture input. Additionally, we found no statistically significant differences between the various implementations of KeyTime predictors.

These results show that KeyTime is much more accurate than its direct competitor CLC. Our user-independent evaluations also confirm previous results from the literature that one gesture sample is sufficient to generate a reliable $\Sigma\Lambda$ model of the gesture path [22] that, when instantiated with representative values for the model parameters [23], can generate synthetic gestures with human-like appearance and characteristics [20,23]. In the next section, we show how KeyTime

Characteristics of the evaluation datasets					Spearman correlations					
Dataset	**Distinct gestures**	**Total gestures**	**Num. participants**	**Average gesture time**	**Isokoski**	**CLC**	**KeyTime predictors**			
							$\hat{t}_\text{M}$	$\hat{t}_\text{Mdn}$	$\hat{t}_{.20}$	$\hat{t}_\text{W}$
1. GDS-fast	16	1,600	10	667 ms	.64 **	.517 *	**.991** ***	.976 ***	**.991** ***	.988 ***
2. GDS-medium	16	1,600	10	1153 ms	.63 **	.614 *	**.992** ***	.935 ***	.947 ***	.938 ***
3. GDS-slow	16	1,600	10	1761 ms	.65 **	.620 **	**.979** ***	.867 ***	.947 ***	.961 ***
4. Difficulty-1	18	5,040	14	1878 ms	.81 ***	.013 *n.s.*	.975 ***	**.979** ***	.975 ***	.975 ***
5. Difficulty-2	20	4,400	11	2088 ms	.78 ***	.627 **	.985 ***	.974 ***	.986 ***	**.992** ***
				Average performance	.70	.478	.984	.946	.969	.971
				Best (max) performance	.81	.627	.991	.979	.991	.992

Table 2. Spearman correlation coefficients (r_s) between predicted production times and groundtruth. Gesture datasets are ordered by their average gesture production times. The highest correlation coefficients are highlighted for each dataset. Statistical significance is denoted as follows: $p < .05$ (*), $p < .01$ (**), and $p < .001$ (***).

Dataset	ANOVA (F-test)	p-value	η_p^2
1. GDS-fast	$F_{(5,90)} = 36.77$	$p < .001$	.67
2. GDS-medium	$F_{(5,90)} = 8.92$	$p < .001$	.33
3. GDS-slow	$F_{(5,90)} = 0.35$	$p > .050$	.02
4. Difficulty-1	$F_{(5,102)} = 3.82$	$p < .010$	.16
5. Difficulty-2	$F_{(5,114)} = 2.23$	$p < .050$	.09

Table 3. Statistical results for production times predicted by KeyTime's $\hat{t}_\text{M}$, $\hat{t}_\text{Mdn}$, $\hat{t}_{.20}$, $\hat{t}_\text{W}$, CLC, and groundtruth.

provides a rich variety of statistical predictions for production times, unmatched by any technique before it.

TOWARD RICHER PREDICTIONS OF PRODUCTION TIME

KeyTime employs a set of production time measurements t_i, $i = 1..n$, corresponding to the n synthetic gesture articulations that it generates under the hood for a given gesture type g. These individual measurements are employed to deliver a richer characterization of users' time performance with stroke gesture input, as follows:

1. **Variance and standard deviation.** KeyTime employs the n time estimates to compute the unbiased sample variance (s^2) and unbiased standard deviation (s) of the predicted production times for a given gesture type, as follows:

$$s^2 = \frac{1}{n-1} \sum_{i=1}^{n} \left(t_i - \hat{t}\right)^2 \tag{10}$$

$$s = \sqrt{s^2} \tag{11}$$

   By replacing $\hat{t}$ in these equations with any of the other time predictors described in the 'KeyTime' section, our technique provides up to four different computations of variance and standard deviation for the predicted production times. In this section, we discuss and evaluate the variance of the median, as it leads to better difference estimates between groups while retaining good statistical power [42].

2. **Confidence intervals.** KeyTime also computes confidence intervals using the unbiased standard deviation s, as follows:

$$\text{CI}_z = \left(\hat{t} - z^* \frac{s}{\sqrt{n}}, \hat{t} + z^* \frac{s}{\sqrt{n}}\right) \tag{12}$$

   where $z^* = 1.645$ for 90% CIs, $z^* = 1.96$ for 95% CIs, and $z^* = 2.576$ for 99% CIs.

Figure 9 illustrates 95% confidence intervals predicted by KeyTime vs. actual groundtruth for the production times of the gestures from each dataset. For example, the 95% confidence interval predicted by KeyTime's $\hat{t}_\text{Mdn}$ for the GDS-fast dataset was $[580{,}769]$ ms, very close to the actual interval of $[554{,}757]$ ms. As we expected, given KeyTime's very accurate predictions so far, the Levene's test for equality of variances was not statistically significant on any of our evaluation datasets ($0.03 < W < 0.08$, $p > .05$).

These results reconfirm that KeyTime is able to provide unprecedented levels of accuracy to characterize users' stroke gesture production times. Moreover, KeyTime is able to deliver sophisticated predictions for production times beyond a single estimation point (such as the mean), by reporting accurate estimations of the variance, standard deviation, range, standard error, and customized confidence intervals for predicted production times.

DISCUSSION

KeyTime delivers accurate predictions of users' stroke gesture production times with no effort required from designers. In this section, we discuss several practical aspects of using KeyTime with our companion web application, and we point to some limitations, but also opportunities for future work.

KeyTime as a practical tool over the web

KeyTime is available as a web application at `https://luis.leiva.name/keytime/`. Designers draw in free form the gesture type for which they wish to obtain time prediction data (see Figure 2) and the web application computes and reports all KeyTime's location and variation estimators. One requirement of KeyTime is that the gesture example provided by the designer should be reconstructed with high quality, as defined by the signal-to-noise ratio (SNR) [22]. Previous work suggested that SNR values below 15 dB denote poor execution quality [2,22,23] and, in such cases, the input gesture should be discarded. To address this aspect, the KeyTime application alerts the designer when the provided gesture example does not have enough quality to generate synthetic gestures. This

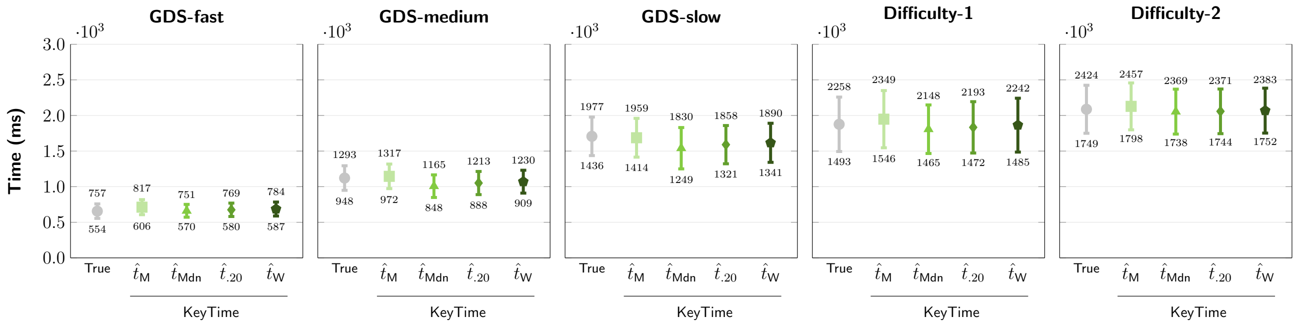

Figure 9. Confidence intervals (at 95%) for groundtruth times (True) and production times predicted by KeyTime implementations.

preliminary validation represents an important feature of KeyTime, which comes as a direct consequence of the fact that lognormal velocity profiles are the ultimate impulse response of a human movement [33]. However, we found this situation to appear extremely rarely in practice: out of the $14{,}240$ stroke gestures that we evaluated in this work, only 22 samples had SNR$<$ 15 dB, which represents a merely 0.15% of the datasets.

Accurate estimations from one gesture sample only

KeyTime only needs *one* gesture example to deliver accurate predictions of stroke gesture production times, as we showed in this work with our leave-one-out cross-validation methodology. Although using only one gesture example could be seen as a limitation (i.e., time predictions are bound to the sample gesture provided by the designer), our experiments revealed that KeyTime is a very accurate user-independent predictor, reporting production times very close in magnitude to the actual groundtruth data. This performance is explained by the fact that the gesture synthesizer employed by KeyTime under the hood [22] uses generic, user-independent value ranges for the $\Sigma\Lambda$ parameters, which were empirically derived from and validated for many user categories by prior work [15,22,24,27]. Concretely, we used in Equation 4 the following noise values [15,22]: $n_\mu = 0.15$, $n_\sigma = 0.35$, $n_D = 0.25$, $n_\theta = 0.3$. Although we should note that different values may be needed for different user categories, such as gestures articulated by visually impaired users [24].

Limitations and future work

We evaluated KeyTime on three public stroke gesture datasets representing a large total number of $14{,}240$ samples, which gives us strong confidence in the reliability of our results. Even if all the gestures from the evaluation datasets were collected using styli [47,52], we are confident that our results transfer to touch gestures articulated with the finger as well. In support of this statement, we refer to previous recent work that actually compared synthetic versions of gestures articulated with the finger and the stylus with the same technique as ours involving the $\Sigma\Lambda$ model of the Kinematic Theory [23]. Concretely, empirical results from Leiva et al. [20,22,23] showed that stroke gestures synthesized from examples collected using the finger or the stylus are similar (i.e., non-statistically significant differences) in terms of their geometric, kinematic, and articulation characteristics [44]. In this context, it is reasonable to conclude that our results do transfer to finger touch gestures as well, but further evaluations are always recommendable, which are left for future work as a sensible reconfirmation of KeyTime's performance. More interesting future work includes extending KeyTime's applicability to estimate the production times of multistroke and multitouch gestures, i.e., gestures composed of more than one stroke and/or articulated with more than one finger, given the large interest for such gestures types [3,4,5,43]. Also, bimanual gestures, performed with both hands touching the surface in parallel and/or in sequence [37], represent another challenging direction for further extension of KeyTime. We leave these interesting explorations as an opportunity for future work.

CONCLUSION

KeyTime is a new high-performing technique informed by the Kinematic Theory that delivers very accurate predictions of users' stroke gesture production times. Through careful evaluations, we showed that KeyTime's predictions are very close to the actual production times of stroke gestures articulated by real users. Moreover, KeyTime only requires one gesture example that designers can produce themselves, and is readily available to any practitioner both as an online application and a RESTful JSON API. KeyTime also raises the bar for future research on stroke gesture time prediction and analysis by delivering a wide palette of predictors of location and dispersion for production times. It is our hope that KeyTime will provide researchers, designers, and practitioners with unprecedented levels of accuracy and sophistication to characterize their users' a priori time performance with stroke gesture input, informing better gesture user interface designs.

ACKNOWLEDGMENTS

R. Plamondon acknowledges support from NSERC-Canada grant RGPIN-2015-06409. R.-D. Vatavu acknowledges support from the project PN-III-P2-2.1-PED-2016-0688 (209PED/2017) financed by UEFISCDI, Romania.

REFERENCES

1. Johnny Accot and Shumin Zhai. 1997. Beyond Fitts' law: Models for trajectory-based HCI tasks. In *Proc. CHI '97*.
2. Abdullah Almaksour, Eric Anquetil, Réjean Plamondon, and Christian O'Reilly. 2011. Synthetic handwritten gesture generation using Sigma-Lognormal model for evolving handwriting classifiers. In *Proc. IGS '11*.

3. Lisa Anthony, Radu-Daniel Vatavu, and Jacob O. Wobbrock. 2013. Understanding the consistency of users' pen and finger stroke gesture articulation. In *Proc. GI '13*.
4. Lisa Anthony and Jacob O. Wobbrock. 2010. A lightweight multistroke recognizer for user interface prototypes. In *Proc. GI '10*.
5. Lisa Anthony and Jacob O. Wobbrock. 2012. $N-protractor: a fast and accurate multistroke recognizer. In *Proc. GI '12*.
6. Caroline Appert and Shumin Zhai. 2009. Using strokes as command shortcuts: Cognitive benefits and toolkit support. In *Proc. CHI '09*.
7. Xiaojun Bi, Yang Li, and Shumin Zhai. 2013. Ffitts law: Modeling finger touch with Fitts' law. In *Proc. CHI '13*.
8. Rachel Blagojevic, Samuel Hsiao-Heng Chang, and Beryl Plimmer. 2010. The power of automatic feature selection: Rubine on steroids. In *Proc. SBIM '10*.
9. Xiang Cao and Shumin Zhai. 2007. Modeling human performance of pen stroke gestures. In *Proc. CHI '07*.
10. Stuart K. Card, Thomas P. Moran, and Allen Newell. 1980. The keystroke-level model for user performance time with interactive systems. *Commun. ACM* **23**(1).
11. Steven J. Castellucci and I. Scott MacKenzie. 2008. Graffiti vs. Unistrokes: An empirical comparison. In *Proc. CHI '08*.
12. Moussa Djioua and Réjean Plamondon. 2009. Studying the variability of handwriting patterns using the Kinematic Theory. *Hum. Mov. Sci.* **28**(5).
13. Paul M. Fitts. 1954. The information capacity of the human motor system in controlling the amplitude of movement. *J. Exp. Psychol.* **47**(6).
14. Tamar Flash and Neville Hogan. 1985. The coordination of arm movements: an experimentally confirmed mathematical model. *J. Neurosci.* **5**(7).
15. Javier Galbally, Réjean Plamondon, Julián Fierrez, and Javier Ortega-García. 2012. Synthetic on-line signature generation. Part II: Experimental validation. *Pattern Recogn.* **45**(7).
16. Poika Isokoski. 2001. Model for unistroke writing time. In *Proc. CHI '01*.
17. Shaun K. Kane, Jeffrey P. Bigham, and Jacob O. Wobbrock. 2008. Slide rule: Making mobile touch screens accessible to blind people using multi-touch interaction techniques. In *Proc. ASSETS '08*.
18. Shaun K. Kane, Jacob O. Wobbrock, and Richard E. Ladner. 2011. Usable gestures for blind people: Understanding preference and performance. In *Proc. CHI '11*.
19. Per Ola Kristensson and Shumin Zhai. 2004. SHARK^2: A large vocabulary shorthand writing system for pen-based computers. In *Proc. UIST '04*.
20. Luis A. Leiva. 2017. Large-scale user perception of synthetic stroke gestures. In *Proc. DIS '17*.
21. Luis A. Leiva, Vicent Alabau, Verónica Romero, Alejandro H. Toselli, and Enrique Vidal. 2014. Context-aware gestures for mixed-initiative text editing UIs. *Interact. Comput.* **27**(1).
22. Luis A. Leiva, Daniel Martín-Albo, and Réjean Plamondon. 2016. Gestures à Go Go: Authoring synthetic human-like stroke gestures using the kinematic theory of rapid movements. *ACM T. Intel. Syst. Tec.* **7**(2).
23. Luis A. Leiva, Daniel Martín-Albo, and Réjean Plamondon. 2017a. The Kinematic Theory produces human-like stroke gestures. *Interact. Comput.* **29**(4).
24. Luis A. Leiva, Daniel Martín-Albo, and Radu-Daniel Vatavu. 2017b. Synthesizing stroke gestures across user populations: A case for users with visual impairments. In *Proc. CHI '17*.
25. A. Chris Long, Jr., James A. Landay, Lawrence A. Rowe, and Joseph Michiels. 2000. Visual similarity of pen gestures. In *Proc. CHI '00*.
26. Daniel Martín-Albo and Luis A. Leiva. 2016. G3: bootstrapping stroke gestures design with synthetic samples and built-in recognizers. In *Proc. MobileHCI '16*.
27. Daniel Martín-Albo, Réjean Plamondon, and Enrique Vidal. 2014. Training of on-line handwriting text recognizers with synthetic text generated using the Kinematic Theory of rapid human movements. In *Proc. ICFHR '14*.
28. Daniel Martín-Albo, Réjean Plamondon, and Enrique Vidal. 2015. Improving sigma-lognormal parameter extraction. In *Proc. ICDAR '15*.
29. Jörg Müller, Antti Oulasvirta, and Roderick Murray-Smith. 2017. Control theoretic models of pointing. *ACM Trans. Comput.-Hum. Interact.* **24**(4).
30. Christian O'Reilly, Réjean Plamondon, Mohamed K. Landou, and Brigitte Stemmer. 2013. Using kinematic analysis of movement to predict the time occurrence of an evoked potential associated with a motor command. *Eur. J. Neurosci.* **37**(2).
31. Réjean Plamondon. 1995a. A kinematic theory of rapid human movements. Part I: Movement representation and control. *Biol. Cybern.* **72**(4).
32. Réjean Plamondon. 1995b. A kinematic theory of rapid human movements. Part II: Movement time and control. *Biol. Cybern.* **72**(4).
33. Réjean Plamondon, Adel M. Alimi, Pierre Yergeau, and Franck Leclerc. 1993. Modelling velocity profiles of rapid movements: a comparative study. *Biol. Cybern.* **69**(1).
34. Réjean Plamondon and Moussa Djioua. 2006. A multi-level representation paradigm for handwriting stroke generation. *Hum. Mov. Sci.* **25**(4–5).

35. Réjean Plamondon, Chunhua Feng, and Anna Woch. 2003. A kinematic theory of rapid human movement. Part IV: A formal mathematical proof and new insights. *Biol. Cybern.* **89**(2).

36. Philip Quinn and Shumin Zhai. 2016. Modeling gesture-typing movements. *Hum.-Comput. Interact.*. Advance online publication. DOI: 10.1080/07370024.2016.1215922.

37. Yosra Rekik, Radu-Daniel Vatavu, and Laurent Grisoni. 2014. Understanding users' perceived difficulty of multi-touch gesture articulation. In *Proc. ICMI '14*.

38. Quentin Roy, Sylvain Malacria, Yves Guiard, Eric Lecolinet, and James Eagan. 2013. Augmented letters: Mnemonic gesture-based shortcuts. In *Proc. CHI '13*.

39. Eugene M. Taranta, II, Mehran Maghoumi, Corey R. Pittman, and Joseph J. LaViola, Jr. 2016. A rapid prototyping approach to synthetic data generation for improved 2D gesture recognition. In *Proc. UIST '16*.

40. Eugene M. Taranta II, Amirreza Samiei, Mehran Maghoumi, Pooya Khaloo, Corey R. Pittman, and Joseph J. LaViola Jr. 2017. Jackknife: A reliable recognizer with few samples and many modalities. In *Proc. CHI '17*.

41. Huawei Tu, Xiangshi Ren, and Shumin Zhai. 2012. A comparative evaluation of finger and pen stroke gestures. In *Proc. CHI '12*.

42. John W. Tukey. 1977. *Exploratory data analysis*. Addison-Wesley Publishing Co.

43. Radu-Daniel Vatavu, Lisa Anthony, and Jacob O. Wobbrock. 2012. Gestures as point clouds: a $P recognizer for user interface prototypes. In *Proc. ICMI '12*.

44. Radu-Daniel Vatavu, Lisa Anthony, and Jacob O. Wobbrock. 2013. Relative accuracy measures for stroke gestures. In *Proc. ICMI '13*.

45. Radu-Daniel Vatavu, Lisa Anthony, and Jacob O. Wobbrock. 2014. Gesture heatmaps: Understanding gesture performance with colorful visualizations. In *Proc. ICMI '14*.

46. Radu-Daniel Vatavu, Gabriel Cramariuc, and Doina Maria Schipor. 2015. Touch interaction for children aged 3 to 6 years: Experimental findings and relationship to motor skills. *Int. J. Hum.-Comput. Stud.* 74.

47. Radu-Daniel Vatavu, Daniel Vogel, Géry Casiez, and Laurent Grisoni. 2011. Estimating the perceived difficulty of pen gestures. In *Proc. INTERACT '11*.

48. Paolo Viviani and Tamar Flash. 1995. Minimum-jerk, two-thirds power law, and isochrony: converging approaches to movement planning. *J. Exp. Psychol.* **21**(1).

49. Paolo Viviani and Carlo Terzuolo. 1982. Trajectory determines movement dynamics. *Neuroscience* **7**(2).

50. Rand Wilcox. 2012. *Modern Statistics for the Social and Behavioral Sciences*. Taylor & Francis Group, LLC, Boca Raton, FL, USA.

51. Jacob O. Wobbrock, Edward Cutrell, Susumu Harada, and I. Scott MacKenzie. 2008. An error model for pointing based on fitts' law. In *Proc. CHI '08*.

52. Jacob O. Wobbrock, Andrew D. Wilson, and Yang Li. 2007. Gestures without libraries, toolkits or training: A $1 recognizer for user interface prototypes. In *Proc. UIST '07*.

53. Shumi Zhai, Per Ola Kristensson, Caroline Appert, Tue H. Anderson, and Xiang Cao. 2012. Foundational issues in touch-surface stroke gesture design — an integrative review. In *Foundations and Trends in Human-Computer Interaction*. Vol. 5.