Chapter 1

Stroke Gesture Synthesis in Human-Computer Interaction

Luis A. Leiva

Aalto University

Daniel Martín-Albo

Independent researcher, WIRIS Math SL

Radu-Daniel Vatavu

MintViz Lab, MANSiD Research Center, University "Ştefan cel Mare" of Suceava

Réjean Plamondon

Laboratoire Scribens, Polytechnique Montréal

Gesture recognizers usually require a large number of examples to achieve good accuracy. To achieve this goal, a series of time-consuming and expensive experiments must be followed, e.g. preparing the lab, recruiting participants, data collection and labeling, and often reporting to review boards. Fortunately, the Kinematic Theory allows to easily bootstrap gesture data generation. The synthesized data, in turn, may enable further applications of interest. In this chapter, we review the foundations of synthetic *stroke gestures* generation; i.e. the synthesis of data sequences comprising 2D points and associated timestamps, derived e.g. from electronic pens and touchscreens. We show that synthesized gestures not only perform equally similar to gestures generated by human users, but also they "look and feel" the same. We also discuss how the synthesized gestures can be used to estimate *production time*, which is a fundamental measure of performance in Human-Computer Interaction. Ultimately, this work benefits researchers and designers who wish to create gesture-driven prototypes or use the synthesized data to build more sophisticated applications.

1. Introduction

Gestures are increasingly becoming a predominant input modality in today's graphical user interfaces (UIs). Gesture interaction is possibly one of the most researched areas in Human-Computer Interaction (HCI), with a long history that started as early as 1960, with the Sketchpad project [Sutherland (1963)] and the RAND tablet [Davis and Ellis (1964)]. Gestures can be mid-air (more prominent in gaming applications) or stroke based (more prominent in mobile applications). We are particularly interested in the latter type, since stroke gestures are becoming more and more relevant to mainstream products such as touchscreen-capable devices like smartphones and tablets; see Figure 1.

Fig. 1. Stroke gestures input is common on many devices with a wide variety of form factors, from smartphones and tablets to tiny touchscreens featured by some wearables, such as smartwatches, and to large interactive surfaces.

Stroke gestures represent the movement trajectory of one or more contact points on a sensitive surface. Stroke gestures are sometimes also called "pen gestures", "hand drawn marks", "hand drawn gestures", "hand markings", or "markings" [Zhai *et al.* (2012)]. Appert and Zhai (2009) demonstrated the cognitive advantage of stroke gestures in the area of command shortcuts, concluding that stroke gestures tend to give richer perceptual cues to the user, to form an association between the shape of the gesture and the meaning of the command. Stroke gestures also may improve the usability of UIs, by replacing standard shortcuts by more accessible triggers.

Stroke gestures have existed in the market for decades. Early examples of commercial products that successfully incorporated gestures are, e.g., PDAs like the Palm Pilot or the Apple Newton, and the Windows Tablet. These devices featured the Graffiti and Unistroke shorthand writing systems, which used a single stroke Roman letter-like gesture vocabulary. Stroke gestures are increasingly being incorporated to facilitate random access to smartphone contents, such as invoking a command hidden in an advanced settings menu or quickly searching for a friend's email in the contacts list. For example, drawing a letter "S" on a mobile phone screen can be used to search in the address book [DroidByDesign (2014)] or speed-dial some contacts [Li (2010a)]. Similarly, in a video game [POW Studios (2008)] players can draw circles to create shields, arcs to launch fireballs, and hearts to drink potions. More recently, the massive online game 'Harry Potter: Wizards Unite' requires players to draw stroke gestures to create spells and defeat enemies [WB Games and Niantic (2019)]. In addition, many modern tablets incorporate a stylus,[a] which allows for more precise and enriched gestures. Therefore, it is expected that stroke gestures will make a notable impact in consumers' lives.

In general, any application that is driven by gestures must rely on some recognition-based techniques. These techniques often require expert knowledge in pattern recognition or machine learning, something that is typically beyond the reach of many developers and UI designers. Furthermore, these techniques require a large pool of labeled training data, which is usually both time-consuming and expensive to acquire. Thus, it is important to investigate how to empower developers to (1) quickly collect and label gesture samples and (2) create accurate recognizers; both for improving UI usage and user experience.

This book chapter provides a compilation of our previous results on the application of the Kinematic Theory to stroke gestures synthesis [Leiva *et al.* (2016); Martín-Albo and Leiva (2016)], and summarizes for readers empirical results regarding the articulation characteristics of synthetic stroke gestures [Leiva *et al.* (2017a)], the perception of human observers comparing synthetic and authentic gesture articulations [Leiva (2017)], as well as applications to predicting users' gesture input performance [Leiva *et al.* (2018a,b)] and extensions towards modeling the articulation characteristics of gestures produced by various categories of users [Leiva *et al.* (2017b); Ungurean *et al.* (2018b,a)]. By compiling all these previous results in one informative and instructional book chapter, we hope to deliver readers a clear understanding of the application potential of synthetic gestures to support innovations and advances in stroke gestures recognition and analysis.

[a]For example: Apple iPad Pro, Microsoft Surface Pro 6, Samsung Galaxy Tab S4, Huawei MediaPad M5 Pro, Wacom Cintiq 22HD, and Lenovo Yoga Book, among others.

2. Related Research

In this section we review core areas that resemble the most to our work: approaches to gesture recognition and gesture bootstrapping.

2.1. *Gesture Recognition*

Gesture recognition has its own roots in sketching and handwriting recognition [Connell and Jain (2000); Deepu *et al.* (2004); Marzinkewitsch (1991); Rubine (1991)]. Classification methods include, among others: linear discriminant analysis [Rubine (1991)], template matching [Connell and Jain (2000)], decision trees [Belaid and Haton (1984)], neural networks [Marzinkewitsch (1991)], hidden Markov models [Koschinski *et al.* (1995)], parsing grammars [Costagliola *et al.* (2004)], support vector machines [Bahlmann *et al.* (2001)], principal component analysis [Deepu *et al.* (2004)], or *ad-hoc* recognizers [Leiva *et al.* (2013, 2014)].

In HCI, most gesture recognizers for prototyping UIs are based on the template matching (or instance-based) approach [Leiva *et al.* (2014)]: a query gesture is geometrically compared against a number of stored templates, using 1 nearest-neighbor for classification and either Euclidean distance or a Mean Square Error (MSE) score as dissimilarity measures. Template matchers are a very viable and a relatively simple solution for recognizing gestures, and can be adapted to personalized user gestures.

Popular examples of these template-based recognizers among the HCI literature are part of the so-called "$ family": $1 [Wobbrock *et al.* (2007)], $N [Anthony and Wobbrock (2010)], and their improved versions Protractor [Li (2010b)] and $N-Protractor [Anthony and Wobbrock (2012)], respectively. Vatavu *et al.* (2012) introduced $P, an articulation-invariant gesture recognizer that represents stroke gestures as clouds of 2D points, discarding thus stroke count, order, and direction, with the most recent instantiation being $Q, a quick and accurate recognizer for point clouds [Vatavu *et al.* (2018)].

For personalized, gesture-based interaction, it is hard to foresee what gestures an end-user would specify and what the distribution of these gestures will look like [Li (2010b)]. Nonetheless, they can be both time and space consuming on the computational side, given the size of the gesture vocabulary and the number of stored templates to define each gesture.

2.2. *Gesture Boostrapping*

Training a high-quality recognizer requires providing examples that illustrate sufficient variation to enable robust inference on unseen, future data. Example-based approaches like GRANDMA [Rubine (1991)], Agate [Landay and Myers (1993)], or Gesture Studio [Lü and Li (2013)] allow developers to create and test gestures by recording examples. There are a number of similar systems tailoring end-users, like EventHurdle [Kim and Nam (2013)], A CAPpella [Dey *et al.* (2004)], or GestIT [Spano *et al.* (2013)]. They support designers' explorative prototyping through programming by demonstration environments. Another strand of research is aimed at simplifying the process of designing gesture sets. For example, Gesture Script [Lü *et al.* (2014)], Gesture Marks [Ouyang and Li (2012)], Gestalt [Patel *et al.* (2010)], or CrowdLearner [Amini and Li (2013)]. Finally, we can find a number of competing systems aimed at creating synthetic 3D gestures as a means to improve gesture recognizers, including e.g. MAGIC [Ashbrook and Starner (2010); Kohlsdorf and Starner (2013)] and Gesture Follower [Caramiaux *et al.* (2014)]. MAGIC performs local perturbations to the resampled points of a gesture, whereas Gesture Follower introduce some variations to a gesture template using Viviani's curve formulation.

Overall, training data is the key factor to build a competitive gesture recognizer, for which most of the previously reviewed approaches have contributed to generating their own, without having to recruit participants and perform time-consuming user evaluations. They also have contributed to decreasing the number of iterations needed to build a fast and stable gesture recognition interface. However, there is no evidence that any of the previous works can produce human-like samples. Further, artificially generated samples usually achieve suboptimal results since they do not illustrate sufficient variation required for high-quality training [Almaksour *et al.* (2011); Plamondon *et al.* (2014); Reznakova *et al.* (2015)] and therefore the achieved error rates are typically higher when compared to training exclusively with human-generated data.

Besides the above-mentioned error rates, another important factor worth mentioning toward the adoption of one gesture recognizer over another is performance, typically represented by processor time and memory usage. It is here where the above-mentioned template matchers usually excel, and the main reason why we chose them to conduct our recognition experiments in the next sections. In addition, template matchers are very easy to convey, implement, and deploy on any platform for non-specialists whose objective is quickly enhancing interactivity and not dealing with the complexity of the underlying recognition algorithms.

3. Synthesizing Gestures

Many models have been proposed to study human movement production; e.g., models relying on neural networks [Bullock and Grossberg (1988)], equilibrium point models [Feldman (1966)], behavioral models [Thomassen *et al.* (1983)], coupled oscillator models [Hollerbach (1981)], kinematic models [Meyer *et al.* (1990)], or models exploiting minimization principles [Flash and Hogan (1985)]. Other models exploit the properties of various functions to reproduce human movements; e.g., exponentials [Plamondon and Lamarche (1986)], second order systems [van der Gon and Thuring (1965)], beta functions [Alimi (2003)], splines [Morasso *et al.* (1983)], Viviani's curves [Viviani and Flash (1995)], and trigonometrical functions [Maarse (1987)]. Among these approaches, the Kinematic Theory [Plamondon (1995)] provides a well-established and solid framework for the study of the production of human movements This framework takes into account different psychophysiological features, such as the neuromuscular response time, and has been shown to outperform many other approaches [Plamondon *et al.* (1993)]. The Sigma-Lognormal ($\Sigma\Lambda$) model [Plamondon and Djioua (2006)] is the latest instantiation of this framework, and very recently has been used to explore gesture recognition.

At a high-level representation, the Kinematic Theory assumes that a complex handwritten trace (such as a stroke gesture) is composed of a series of *primitives*[b] (circular arcs) connecting a sequence of *virtual targets*. This series of primitives conform the "action plan" of the user, which is fed through the neuromuscular network to produce a trajectory that leaves a handwritten trace; see Figure 2.

Under this framework, each ith gesture primitive is modeled according to a lognormal function representing their corresponding velocity profile. Each primitive is defined by a set of *central* parameters (D, t_0, θ) and *peripheral* parameters (μ, σ) [Plamondon (1995)]. The central parameters describe the articulated handwritten trajectory, whereas the peripheral parameters describe the reaction to such articulation. Concretely, D represents the overall size (amplitude) of the primitive, t_0 accounts for the

[b] In the gesture recognition literature, the term "stroke" denotes the trajectory between two consecutive pointer-down and pointer-up events. In the Kinematic Theory literature, a "stroke" is what we call "primitive" in this chapter.

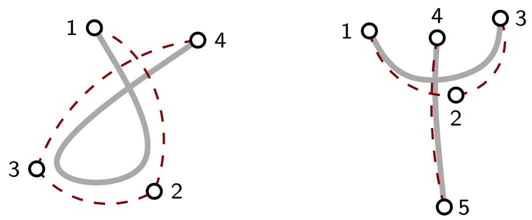

Fig. 2. A gesture stroke (solid lines) is described by the temporal overlap of a series of primitives (dashed arcs) connecting a sequence of virtual targets (numbered circles). Each primitive is described by a lognormal velocity profile.

handwriting start time, θ informs about rotation and direction, and finally μ and σ denote the mean and variance of the lognormal function, respectively, reflecting the neuromuscular response.

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

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

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

Then, a $\Sigma\Lambda$ extractor computes the parameter values that best explain the observed velocity profiles [Martín-Albo *et al.* (2015)]. Once the gesture primitives are modeled, perturbations can be added to the model parameters in order to produce different gesture variations [Leiva *et al.* (2017a)]:

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

where $p_i = \{\mu_i, \sigma_i, D_i, \theta_i\}$ denote the $\Sigma\Lambda$ parameters, with $n_{p_i} = \mathcal{U}(-n_i, n_i)$ being the noise applied to each primitive, according to a uniform distribution (i.e., a rectangular distribution with constant probability) centered around the expected human variability ranges [Galbally *et al.* (2012b); Leiva *et al.* (2017a)]: $n_\mu = n_\sigma = 0.1$, $n_D = 0.15$, $n_\theta = 0.06$. Figure 3 shows a series of actual examples of the synthetic gestures produced with the Kinematic Theory.

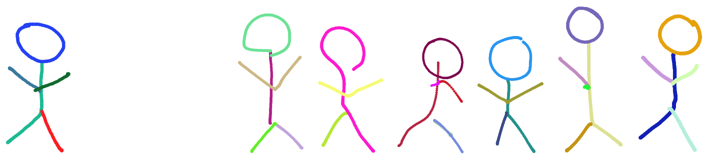

Fig. 3. Examples of gestures synthetized with the Kinematic Theory, using a single human example as input.

Previous works have demonstrated the connection between the distortion of the $\Sigma\Lambda$ model parameters and the intra-variability found in human handwriting [Djioua and Plamondon (2009)]. Combining both types of variations reflects real-life situations like performing the same movement under different psychophysiological conditions. 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 [Leiva *et al.* (2016, 2017a)]. Finally, we should mention that we decided to not add perturbations to the t_0 parameter, since it is very sensitive to fluctuations [Djioua and Plamondon (2009); Leiva *et al.* (2016)]. Nevertheless, perturbations in t_0 have been suggested to reflect changes in the sequence of command instantiation e.g. due to a decrease in attention or cognitive neuromotor fatigue [Djioua and Plamondon (2009)]. Therefore, further analysis of the t_0 parameter is left as an opportunity for future work.

4. Gesture Performance Analysis

The first experiment compared the performance of synthetic gestures with that of human samples under user-independent tests in terms of articulation speed, input device, and gesture variability. The user-independent setting is the more generalizable scenario, however the interested reader may consult user-dependent tests and a follow-up evaluation in our previous work [Leiva *et al.* (2016)].

We synthesized two popular datasets in HCI: GDS [Wobbrock *et al.* (2007)] and MMG [Anthony and Wobbrock (2012)]. Both datasets include examples of simple and complex gestures produced with different devices and under different execution speeds. Therefore these datasets are a relevant testbed for conducting replicable research on stroke gestures.

On the one hand, the GDS dataset comprises 5,280 *unistroke* gestures (16 classes). Ten users provided 10 samples per class at 3 articulation speeds (slow, medium, fast) using an iPAQ Pocket PC (stylus as input device). On the other hand, the MMG dataset comprises 9,600 *multistroke* gestures (16 classes). Twenty users provided 10 samples per class at 3 articulation speeds (same as in GDS) using either finger (half of the users) or stylus as input device on a Tablet PC. Both GDS and MMG datasets are available at http://depts.washington.edu/madlab/proj/dollar/. The synthesized versions of both datasets are available at https://luis.leiva.name/g3/#datasets.

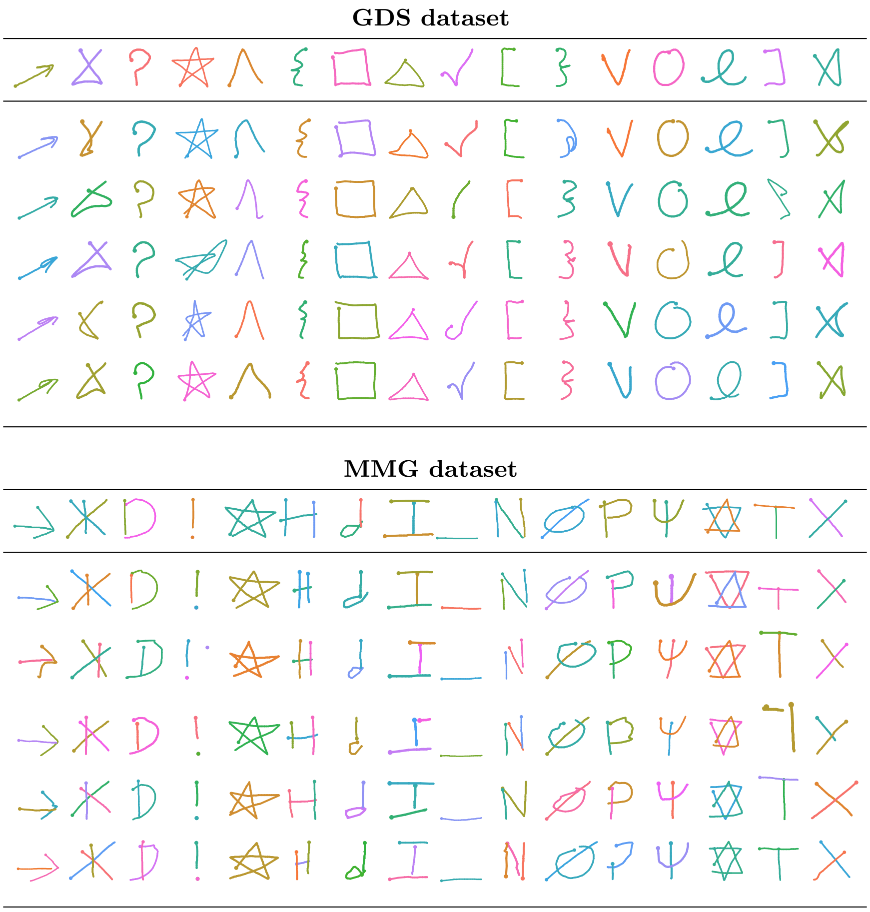

Fig. 4. Five synthesized gestures using one human example (top row) as input.

4.1. *Impact of Articulation Speed*

We sought to analyze whether gesture articulation speed leads to a difference in classification error rates between human and synthetic templates. The GDS dataset (unistroke gestures) was analyzed with the $1 recognizer, whereas the MMG dataset (multistroke gestures) was analyzed with the $P recognizer. Both recognizers were initialized with 10 templates, as suggested in previous work [Anthony and Wobbrock (2010); Li (2010b); Wobbrock *et al.* (2007)]. Table 1 summarizes this experiment.

As can be observed, error rates are very small ($< 1\%$), indicating that the recognizers were successful in classifying correctly each gesture instance. The error rates for the synthesized gestures were slightly higher, therefore we ran two-tailed paired t-tests to investigate further these differences. We applied the Bonferroni correction, to counteract Type I errors as a result of multiple comparisons. The statistical tests revealed no statistically significant differences on recognition errors for any of the articulation speeds, suggesting thus that synthetic gestures can be recognized with the same confidence as their human counterparts, regardless the execution speed.

Table 1. Effect of articulation speed on error rates (in %).

Type		**GDS**			**MMG**	
	slow	med.	fast	slow	med.	fast
Human	0.09	0.06	0.18	0.03	0.02	0.06
Synthetic	0.20	0.27	0.90	0.17	0.36	0.56

4.2. *Impact of Input Device*

We also sought to analyze whether the input device leads to a difference in classification error rates between human and synthetic templates. This analysis was performed over the MMG dataset, which is the one that provides two data splits (finger and stylus). We used the $P recognizer with 10 templates, as in the previous experiment. Table 2 summarizes this experiment.

As can be observed, error rates are very small ($< 0.5\%$), indicating that the recognizers were successful in classifying correctly each gesture instance. The error rates for the synthesized gestures were slightly higher, therefore we ran two-tailed paired t-tests (Bonferroni corrected) to investigate further these differences. The statistical tests revealed no statistically significant differences on recognition errors for any of the articulation speeds, suggesting thus that gestures can be successfully synthesized using either a stylus or the human finger as input.

Table 2. Effect of input device on error rates (in %).

Human		**Synthetic**	
finger	stylus	finger	stylus
0.03	0.04	0.43	0.29

4.3. *Impact of Gesture Variability*

Finally, we sought to analyze whether an increase in the amount of noise ξ introduced to the $\Sigma\Lambda$ model parameters leads to more variable synthetic gestures. We computed the mean squared error[c] between human and synthetic gestures for different number of synthesized samples using ξ from 0.0 (no variability) to 1.0 (maximum variability, in the allotted human ranges [Leiva *et al.* (2016)]). Table 3 summarizes this experiment.

As expected, it was found that synthetic samples are more variable as ξ increases. As expected, within-samples variability was found to increase as the number of requested synthetic samples increases. In general, we observed that requesting a small number of synthetic samples (10 samples per gesture) provides slightly less variable samples. For a given value of ξ, variability was found to increase as the number of requested synthetic samples increases, though we suspect it is because the MSE is underestimated for small batch sizes. Indeed, the standard error (Table 3) gets smaller as the number of samples gets larger, because the mean of a large sample is likely to be closer to the true population mean.

5. Gesture Similarity Analysis

To provide further evidence on the value of the Kinematic Theory as a means to generate synthetic stroke gestures, we conducted an online survey that measured the user perception toward gestures' human-likeness. We used the same datasets depicted in the previous section, both in their original and synthesized form. The survey is still available online at

[c]To ease computation, strokes were resampled in such a way that a human sample and its synthesized samples had the same length.

Table 3. Gesture variability in term of mean squared error (standard errors in parentheses).

N	**GDS**				**MMG**			
	$\xi = 0.0$		$\xi = 1.0$		$\xi = 0.0$		$\xi = 1.0$	
10	554.9	(56.6)	593.6	(57.8)	169.7	(9.5)	490.4	(38.2)
100	"	(17.8)	622.1	(20.5)	"	(3.0)	493.4	(10.7)
1000	"	(5.6)	621.4	(6.4)	"	(0.9)	498.7	(3.5)

https://g3.prhlt.upv.es/guessit/. Eventually, 236 participants took part in this study.

We defined the guessing accuracy as the user's ability to distinguish between human and synthetic samples; i.e., the proportion of gestures that were successfully classified by the user. Then, two types of errors[d] can be committed [Galbally *et al.* (2012a,b)]: (i) a synthetic gesture is mistaken with a real sample, measured by the False Real Rate (Type I error); and (ii) a real gesture is marked as synthetic, measured by the False Synthetic Rate (Type II error). The results are presented in Table 4.

Table 4. Accuracy and error rates (in %).

Accuracy	**False Real Rate**	**False Synthetic Rate**
49.65	29.84	20.40

A paired two-sample t-test (two-tailed alternative hypothesis) revealed that there is no difference between classified and misclassified gestures, proving thus the human-like appearance of synthetic samples.[e] In sum, participants could not tell human and synthetic gestures apart. Follow-up analyses [Leiva *et al.* (2017a); Leiva (2017)] provided further evidence that synthesized gestures are actually reflective of how users produce stroke gestures.

We also examined the intra-class variability of human gestures, distance-wise; i.e., how variable is a human gesture sample as compared to the rest of the human samples that belong to the same gesture class. The Pearson's correlation coefficient was found to be $\rho > 0.9$ in all datasets, and we observed that ρ decreased as ξ increased. This indicates that, as expected,

[d] The sum of False Real Rate (Type I error) and False Synthetic Rate (Type II error) yields the overall error rate (100 - Accuracy).
[e] We assume that humans are reasonable judges of realism, but it might not always be the case.

samples synthesized with a low variability degree look much more similar to the human samples from which they were generated. Taken together, these experiments suggest that the visual appearance of synthetic gestures is very similar and close to that of their human counterparts.

6. Production Times Analysis

As a practical application of the synthesized gestures with the Kinematic Theory, we report results on a fundamental topic in HCI. 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 [Castellucci and MacKenzie (2008); Cao and Zhai (2007); Vatavu *et al.* (2011); Rekik *et al.* (2014)]. 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 [Appert and Zhai (2009); Castellucci and MacKenzie (2008)] or indirectly, e.g., what are the easiest gestures to execute from a given set [Vatavu *et al.* (2011); Rekik *et al.* (2014)]. Moreover, gesture production time turned out to be an excellent predictor of users' subjective perceptions of the difficulty to articulate stroke gestures [Rekik *et al.* (2014); Vatavu *et al.* (2011)]. 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.

For this experiment, we compared the synthesized gesture production times against ther human counterparts [Leiva *et al.* (2018a,b)] using the same two datasets described in the previous experiments (GDS and MMG). The performance of $\Sigma\Lambda$ as time predictor was evaluated with the following accuracy measures:

Rank accuracy evaluates the extent to which the synthetic gestures deliver the correct *ranking* (Spearman's correlation) of gestures according to their production times. The closer the rank to 1, the better.

Absolute accuracy evaluates the extent to which the synthetic gestures deliver the correct *magnitude* (in ms) of the expected production time of a given gesture type. The closer the magnitude to the true production time, the better.

Production times were computed using a user-independent leave-one-out cross-validation procedure, which considers each execution from each gesture produced by each participant as the representative gesture sample to compare against the $\Sigma\Lambda$ model. The results of this experiment are shown in Table 5. In any case, we found no statistically significant differences between true and estimated times, which builds our confidence that synthesized gestures are on par with users' actual time performance with stroke gesture input.

Table 5. Production times analysis. Time in ms. SDs denoted in parentheses.

Dataset	**Device**	**Speed**	**True time**	**Rank acc.**	**Abs. acc.**
GDS	stylus	slow	1607 (679)	.979***	1587 (678)
		med.	1055 (438)	.992***	1078 (425)
		fast	618 (259)	.991***	670 (273)
MMG	stylus	slow	1011 (683)	.932***	1120 (638)
		med.	668 (511)	.900***	672 (408)
		fast	537 (440)	.788***	609 (339)
MMG	finger	slow	1038 (694)	.950***	940 (665)
		med.	688 (504)	.914***	805 (528)
		fast	553 (434)	.775***	730 (464)

Statistical significance levels are $p < .001$ in all cases.

We are thus confident that synthetic samples look like real ones at the geometric level. However, velocity profiles comprise subtle time-dependent relationships that so far have not been studied for stroke gestures. Therefore, How realistic are the velocity profiles synthesized? To answer this question, the plots below display the velocity profiles of a set of gestures chosen at random from each dataset. We can observe that the synthesized velocity profiles are often in line with the human velocity profiles, however the velocity range is typically smaller for the synthesized gestures. This happened especially to the synthesized samples from the GDS dataset, as show in Figure 5, and is a consequence of our current implementation of the $\Sigma\Lambda$ model, which transforms the original stream into constant frequency; i.e., the resulting synthesized coordinates are uniformly distributed in time. This resampling process allows to "fix" downsampled strokes that were acquired with under-resourced hardware [Leiva *et al.* (2016)] but an undesired side effect is that velocity profiles look smoother than usual. The MMG is a particularly bad dataset to analyze velocity profiles, because many gesture points have duplicated timestamps [Leiva *et al.* (2017a)], which makes it challenging to estimate velocity values. For example, if two consecutive points at timestep $i-1$ and i have the same timestamp, the time derivative at timestep i is zero and therefore the velocity value is infinite. In consequence, careful preprocessing must be performed. Figure 6 shows a couple of velocity profiles corresponding to gestures articulated with stylus and finger, respectively.

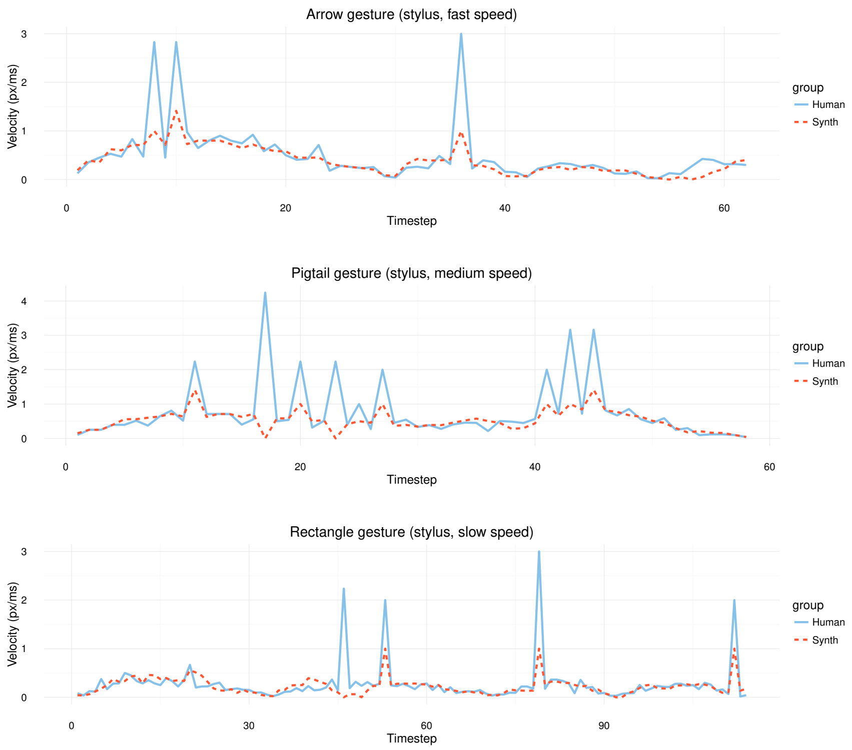

Fig. 5. Comparison of velocity profiles from samples in the GDS dataset.

7. Discussion

Users tend to be reluctant to invest time and effort upfront to train or adjust software before using it [Appert and Zhai (2009)]. Further, users are unwilling to provide more than a small set of samples for training [Li (2010b)]. Consequently, synthesizing techniques like ours are of high value, as they help to lower time and costs associated to recruiting users and subsequent data labeling. Furthermore, researchers can focus exclusively on UI design

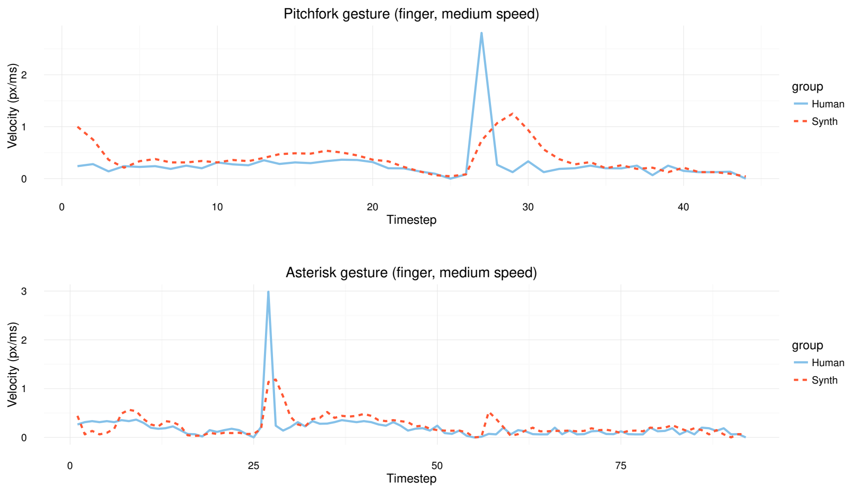

Fig. 6. Comparison of velocity profiles from samples in the MMG dataset.

rather than fret over machine learning concepts or toolkits that may not be available for their platform. Eventually, our web application [Martín-Albo and Leiva (2016)] can be used for rapid prototyping, allowing developers to define new gestures on demand. The interested reader can access it at https://g3.prhlt.upv.es.

Until now the "human likeness" of synthesized gestures was measured indirectly, intermediated by classification/recognition accuracy performance. Our studies are important because recent research has shown how different users produce different gesture articulations in various conditions. Thus, researchers and practitioners can be confident that synthesized gestures using the Kinematic Theory are actually reflective of how users produce stroke gestures.

One requirement for the $\Sigma\Lambda$ model to produce proper results is that the user-provided gesture example should be reconstructed with high quality, as defined by the signal-to-noise ratio (SNR). Previous work suggested that SNR values below 15 dB denote poor execution quality [Almaksour *et al.* (2011); Leiva *et al.* (2016, 2017a)] and, in such cases, the input gesture should be discarded. However, we found this situation to appear extremely rarely in practice: out of the $14,240$ stroke gestures that we evaluated in our experiments, only 22 samples had SNR$<$ 15 dB, which represents merely 0.15% of the data.

On a side note, the concepts concerning internal models of human movements have been well supported by behavioral studies in the field of sensory motor control. Overall, it is assumed that users are "ideal" motion planners who choose movement trajectories to minimize an expected loss [Trommershäuser *et al.* (2003); Quinn and Zhai (2018)]. Currently, we can find two compelling theories to describe those movements: the Minimization Theory [Flash and Hogan (1985)] and the Kinematic Theory [Plamondon (1995)]. Actually, it has been shown that their concepts are linked and describe, with different arguments, a model of velocity profiles [Djioua and Plamondon (2010); Leiva *et al.* (2017a)], the Minimization Theory being as a very good approximation of the lognormal description provided by the Kinematic Theory.

The previous instantiation of the Kinematic Theory (the Delta-Lognormal model) assumed that the production of a stroke requires the synergetic activation of two neuromuscular systems, one agonist and the other antagonist to the direction of the movement. These synchronous commands propagate in parallel across the two neuromuscular systems, each of which is described by a lognormal impulse response and has its own timing properties. On the contrary, the $\Sigma\Lambda$ model does not assume that the two neuromuscular systems are working in precisely opposite directions. The output velocity is thus described by a vectorial summation of the contribution of each neuromuscular system involved in the production of a stroke. This model is actually very general, and is not limited to a single stroke description [Plamondon and Djioua (2006); O'Reilly and Plamondon (2009)]. This corroborates the prediction of the Kinematic Theory, where it is theorized that the ideal impulse response of the human neuromotor system follows a lognormal response that results from the limiting behavior of a large number of interdependent neuromuscular networks, as stated in the first chapter of this book and referred to as the lognormality principle.

The fundamental advantage of our approach over others is that the $\Sigma\Lambda$ model only needs *one* user example to start synthesizing more gestures. Although using only one gesture example could be seen as a limitation (i.e., the results are bound to the sample gesture provided as seed), our experiments revealed that synthetic gestures are on par with their human counterparts. This performance is explained by the fact that our synthesizer [Leiva *et al.* (2016); Martín-Albo and Leiva (2016)] 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 [Galbally *et al.* (2012b); Leiva *et al.* (2016, 2017b); Martín-Albo *et al.* (2014)]. The interested reader can refer to these prior works to know other range values and how they may impact recognition performance. Although we should note that different values may be needed for different user categories, such as gestures articulated by visually impaired users [Leiva *et al.* (2017b)] or users with motor impairments [Ungurean *et al.* (2018b,a)].

We also have shown that the ΣΛ model delivers accurate predictions of users' stroke gesture production times with no effort required from designers. This is a practical application to enable effective gesture sets design. We should stress the fact that it is important to provide designers with 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 [Anthony *et al.* (2013); Vatavu *et al.* (2013)], it also causes variation in their production times. With synthesized gestures, we are able to deliver the extra information given by location and dispersion-based measures that 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.

Finally, in light of the analysis of the synthetic velocity profiles, we believe there is still room for improvement in how synthetic gestures are produced by the ΣΛ model. While the articulation of synthetic gestures, as reflected by the shape of their velocity profiles, is very much in line with their corresponding human velocity profiles, the ΣΛ model is often unable to capture *all* the compensatory micro-movements observed in human gesturing, very much like it happens in human handwriting. These subtle micro-movements are explained by the isochrony principle [Viviani and McCollum (1983)], which states that the velocity of a movement is proportionally linked to its linear extension (or trajectory) so as to permit the execution time to be maintained approximately constant [Freund (1986)]. Stroke gestures captured on commodity touchscreens, such as the ones we have analyzed in this work, have usually low temporal resolution and *asynchronous* timestamps. Therefore, their velocity profiles are much more challenging to reconstruct than, say, handwritten signatures performed on a high-resolution tablet with a high and fixed sampling rate. More research is thus needed to understand the articulation of stroke gestures on touch-capable devices.

8. Conclusion

Our experiments provide evidence against the implied alternate hypothesis of a difference between human and synthesized stroke gestures. Researchers and practitioners can be finally confident that the Kinematic Theory generates stroke gestures that not only perform equally similar to their human couterparts but also they look and feel the same. And while there is still room for improving how synthetic strokes gestures are articulated, it is reliable to generate synthetic datasets this way, since the overall performance and behavior of gesture samples will be consistently similar to that of actual users. This can be useful to researchers and practitioners in many ways.

As a practical application of the synthesized gestures, we have discussed how they can be used to estimate production times, which is one of the fundamental performance measures in HCI. We have left out other practical applications that also might be of interest to practitioners, such as synthesizing gestures across populations [Leiva *et al.* (2017b); Ungurean *et al.* (2018b)], evaluating the effect of hardware resolution on handwriting analysis [Martín-Albo *et al.* (2016b)], or detecting the "hidden" user intent in mouse cursor movements [Martín-Albo *et al.* (2016a)]. However, space precludes a complete treatment of all possible applications of synthetic gestures in HCI. Instead, the interested reader is redirected to the works referenced above.

In sum, the Kinematic Theory provides the HCI community with a reliable way to synthesize stroke gesture sets without having to expressly collect them from a large pool human subjects. However, we do not to encourage the substitution of human gestures by synthetic ones, but rather to provide an automated way to lower the need of recruiting a large number of users and subsequent data labeling.

References

Alimi, A. M. (2003). Beta neuro-fuzzy systems, in W. Duch and D. Rutkowska. eds., *TASK Quarterly J., Special Issue on Neural Networks*, Vol. 7.

Almaksour, A., Anquetil, E., Plamondon, R. and O'Reilly, C. (2011). Synthetic handwritten gesture generation using Sigma-Lognormal model for evolving handwriting classifiers, in *Proceedings of Biennial Conf. of the Intl. Graphonomics Society (IGS)*.

Amini, S. and Li, Y. (2013). CrowdLearner: Rapidly creating mobile recognizers using crowdsourcing, in *Proceedings of Annual ACM Symp. on User Interface Software and Technology (UIST)*.

Anthony, L., Vatavu, R.-D. and Wobbrock, J. O. (2013). Understanding the consistency of users' pen and finger stroke gesture articulation, in *Proceedings of Graphics Interface (GI)*.

Anthony, L. and Wobbrock, J. O. (2010). A lightweight multistroke recognizer for user interface prototypes, in *Proceedings of Graphics Interface (GI)*.

Anthony, L. and Wobbrock, J. O. (2012). $N-protractor: a fast and accurate multistroke recognizer, in *Proceedings of Graphics Interface (GI)*.

Appert, C. and Zhai, S. (2009). Using strokes as command shortcuts: Cognitive benefits and toolkit support, in *Proceedings of SIGCHI Conf. on Human Factors in Computing Systems (CHI)*.

Ashbrook, D. and Starner, T. E. (2010). MAGIC: A motion gesture design tool, in *Proceedings of SIGCHI Conf. on Human Factors in Computing Systems (CHI)*.

Bahlmann, C., Haasdonk, B. and Burkhardt, H. (2001). On-line handwriting recognition with support vector machines: A kernel approach, in *Proceedings of Intl. Workshop on Frontiers in Handwriting Recognition (IWFHR)*.

Belaid, A. and Haton, J.-P. (1984). A syntactic approach for handwritten formula recognition, *IEEE Trans. Pattern Anal. Mach. Intell.* **6**, 1.

Bullock, D. and Grossberg, S. (1988). The VITE model: a neural command circuit for generating arm and articulator trajectories, in *Dynamic Patterns in Complex Systems*.

Cao, X. and Zhai, S. (2007). Modeling human performance of pen stroke gestures, in *Proceedings of SIGCHI Conf. on Human Factors in Computing Systems (CHI)*.

Caramiaux, B., Montecchio, N., Tanaka, A. and Bevilacqua, F. (2014). Adaptive gesture recognition with variation estimation for interactive systems, *ACM Trans. Interact. Intell. Syst.* **4**, 4.

Castellucci, S. J. and MacKenzie, I. S. (2008). Graffiti vs. Unistrokes: An empirical comparison, in *Proceedings of SIGCHI Conf. on Human Factors in Computing Systems (CHI)*.

Connell, S. D. and Jain, A. K. (2000). Template-based on-line character recognition, *Pattern Recogn.* **34**, 1.

Costagliola, G., Deufemia, V., Polese, G. and Risi, M. (2004). A parsing technique for sketch recognition systems, in *Proceedings of Symp. on Visual Languages and Human-Centric Computing (VLHCC)*.

Davis, M. R. and Ellis, T. O. (1964). The RAND tablet: A man-machine graphical communication device, in *Proceedings of American Federation of Information Processing Societies (AFIPS)*.

Deepu, V., Madhvanath, S. and Ramakrishnan, A. G. (2004). Principal component analysis for online handwritten character recognition, in *Proceedings of Intl. Conf. on Pattern Recognition (ICPR)*.

Dey, A. K., Hamid, R., Beckmann, C., Li, I. and Hsu, D. (2004). A CAPpella: Programming by demonstration of context-aware applications, in *Proceedings of SIGCHI Conf. on Human Factors in Computing Systems (CHI)*.

Djioua, M. and Plamondon, R. (2009). Studying the variability of handwriting patterns using the Kinematic Theory, *Hum. Mov. Sci.* **28**, 5.

Djioua, M. and Plamondon, R. (2010). The limit profile of a rapid movement velocity, *Hum. Mov. Sci.* **29**, 1.

DroidByDesign (2014). Oftseen gestures, URL https://play.google.com/store/apps/details?id=com.proofbydesign.oftSeenGestures, accessed on May 2019.

Feldman, A. (1966). Functional tuning of the nervous system with control of movement or maintenance of a steady posture, *Biophysics* **11**, 1.

Flash, T. and Hogan, N. (1985). The coordination of arm movements: an experimentally confirmed mathematical model, *J. Neurosci.* **5**, 7.

Freund, H.-J. (1986). Time control of hand movements, *Prog. Brain Res.* **64**.

Galbally, J., Plamondon, R., Fierrez, J. and Ortega-García, J. (2012a). Synthetic on-line signature generation. Part I: Methodology and algorithms, *Pattern Recogn.* **45**, 7.

Galbally, J., Plamondon, R., Fierrez, J. and Ortega-García, J. (2012b). Synthetic on-line signature generation. Part II: Experimental validation, *Pattern Recogn.* **45**, 7.

Hollerbach, J. M. (1981). An oscillation theory of handwriting, *Biol. Cybern.* **39**, 2.

Kim, J.-W. and Nam, T.-J. (2013). EventHurdle: Supporting designers' exploratory interaction prototyping with gesture-based sensors, in *Proceedings of SIGCHI Conf. on Human Factors in Computing Systems (CHI)*.

Kohlsdorf, D. and Starner, T. E. (2013). MAGIC Summoning: Towards automatic suggesting and testing of gestures with low probability of false positives during use, *J. Mach. Learn. Res.* **14**, 1.

Koschinski, M., Winkler, H. J. and Lang, M. (1995). Segmentation and recognition of symbols within handwritten mathematical expressions, in *Proceedings of Intl. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*.

Landay, J. A. and Myers, B. A. (1993). Extending an existing user interface toolkit to support gesture recognition, in *Proceedings of INTERCHI'93 Conf. on Human Factors in Computing Systems*.

Leiva, L. A. (2017). Large-scale user perception of synthetic stroke gestures, in *Proceedings of the ACM Conf. on Designing Interactive Systems (DIS)*.

Leiva, L. A., Alabau, V., Romero, V., Toselli, A. H. and Vidal, E. (2014). Context-aware gestures for mixed-initiative text editing UIs, *Interact. Comput.* **27**, 1.

Leiva, L. A., Alabau, V. and Vidal, E. (2013). Error-proof, high-performance, and context-aware gestures for interactive text edition, in *Proceedings of Extended Abstracts on Human Factors in Computing Systems (CHI EA)*.

Leiva, L. A., Martín-Albo, D. and Plamondon, R. (2016). Gestures à Go Go: Authoring synthetic human-like stroke gestures using the kinematic theory of rapid movements, *ACM Trans. Intell. Syst. Technol.* **7**, 2.

Leiva, L. A., Martín-Albo, D. and Plamondon, R. (2017a). The Kinematic Theory produces human-like stroke gestures, *Interact. Comput.* **29**, 4.

Leiva, L. A., Martín-Albo, D., Plamondon, R. and Vatavu, R.-D. (2018a). KeyTime: Super-accurate prediction of stroke gesture production times, in *Proceedings of SIGCHI Conf. on Human Factors in Computing Systems (CHI)*.

Leiva, L. A., Martín-Albo, D. and Vatavu, R.-D. (2017b). Synthesizing stroke gestures across user populations: A case for users with visual impairments, in *Proceedings of SIGCHI Conf. on Human Factors in Computing Systems (CHI)*.

Leiva, L. A., Martín-Albo, D. and Vatavu, R.-D. (2018b). GATO: Predicting human performance with multistroke and multitouch gesture input, in *Proceedings of ACM Conf. on Human-computer interaction with mobile devices and services (MobileHCI)*.

Li, Y. (2010a). Gesture Search: A tool for fast mobile data access, in *Proceedings of Annual ACM Symp. on User Interface Software and Technology (UIST)*.

Li, Y. (2010b). Protractor: a fast and accurate gesture recognizer, in *Proceedings of SIGCHI Conf. on Human Factors in Computing Systems (CHI)*.

Lü, H., Fogarty, J. A. and Li, Y. (2014). Gesture Script: Recognizing gestures and their structure using rendering scripts and interactively trained parts, in *Proceedings of SIGCHI Conf. on Human Factors in Computing Systems (CHI)*.

Lü, H. and Li, Y. (2013). Gesture Studio: Authoring multi-touch interactions through demonstration and declaration, in *Proceedings of SIGCHI Conf. on Human Factors in Computing Systems (CHI)*.

Maarse, F. J. (1987). *The Study of Handwriting Movement: Peripheral Models and Signal Processing Techniques*, Swets & Zeitlinger.

Martín-Albo, D. and Leiva, L. A. (2016). G3: bootstrapping stroke gestures design with synthetic samples and built-in recognizers, in *Proceedings of ACM Conf. on Human-computer interaction with mobile devices and services (MobileHCI)*.

Martín-Albo, D., Leiva, L. A., Huang, J. and Plamondon, R. (2016a). Strokes of insight: User intent detection and kinematic compression of mouse cursor trails, *Inform. Process. Manag.* **56**, 6.

Martín-Albo, D., Leiva, L. A. and Plamondon, R. (2016b). On the design of personal digital bodyguards: Impact of hardware resolution on handwriting analysis, in *Proceedings of Intl. Conf. on Frontiers in Handwriting Recognition (ICFHR)*.

Martín-Albo, D., Plamondon, R. and Vidal, E. (2014). Training of on-line handwriting text recognizers with synthetic text generated using the Kinematic Theory of rapid human movements, in *Proceedings of Intl. Conf. on Frontiers in Handwriting Recognition (ICFHR)*.

Martín-Albo, D., Plamondon, R. and Vidal, E. (2015). Improving sigma-lognormal parameter extraction, in *Proceedings of Intl. Conf. on Document Analysis and Recognition (ICDAR)*.

Marzinkewitsch, R. (1991). Operating computer algebra systems by hand-printed input, in *Proceedings of Intl. Symp. on Algorithms and Computation (ISAAC)*.

Meyer, D. E., Keith-Smith, J. E., Kornblum, S., Abrams, R. A. and Wright, C. E. (1990). Speed-accuracy tradeoffs in aimed movements: Toward a theory of rapid voluntary action, *Atten. and Perform.* **13**, 23.

Morasso, P., Ivaldi, F. A. M. and Ruggiero, C. (1983). How a discontinuous mechanism can produce continuous patterns in trajectory formation and handwriting, *Acta Psychol.* **54**, 1.

O'Reilly, C. and Plamondon, R. (2009). Development of a Sigma-Lognormal representation for on-line signatures, *Pattern Recogn.* **42**, 12.

Ouyang, T. and Li, Y. (2012). Bootstrapping personal gesture shortcuts with the wisdom of the crowd and handwriting recognition, in *Proceedings of SIGCHI Conf. on Human Factors in Computing Systems (CHI)*.

Patel, K., Bancroft, N., Drucker, S. M., Fogarty, J., Ko, A. J. and Landay, J. (2010). Gestalt: Integrated support for implementation and analysis in machine learning, in *Proceedings of Annual ACM Symp. on User Interface Software and Technology (UIST)*.

Plamondon, R. (1995). A kinematic theory of rapid human movements. Part I: Movement representation and control, *Biol. Cybern.* **72**, 4.

Plamondon, R., Alimi, A. M., Yergeau, P. and Leclerc, F. (1993). Modelling velocity profiles of rapid movements: a comparative study, *Biol. Cybern.* **69**, 1.

Plamondon, R. and Djioua, M. (2006). A multi-level representation paradigm for handwriting stroke generation, *Hum. Mov. Sci.* **25**, 4–5.

Plamondon, R. and Lamarche, F. (1986). Modelization of handwriting: A system approach, in H. S. R. Kao, G. P. van Galen and R. Hoosain. eds., *Graphonomics: Contemporary Research in Handwriting*.

Plamondon, R., O'Reilly, C., Galbally, J., Almaksour, A. and Anquetil, E. (2014). Recent developments in the study of rapid human movements with the Kinematic Theory: Applications to handwriting and signature synthesis, *Pattern Recogn. Lett.* **35**.

POW Studios (2008). Mr. spiff's revenge, URL https://youtu.be/nqeIuNTbT-8, accessed on May 2019.

Quinn, P. and Zhai, S. (2018). Modeling gesture-typing movements, *Hum.-Comput. Interact.* **33**, 2.

Rekik, Y., Vatavu, R.-D. and Grisoni, L. (2014). Understanding users' perceived difficulty of multi-touch gesture articulation, in *Proceedings of Intl. Conf. on Multimodal Interaction (ICMI)*.

Reznakova, M., Tencer, L., Plamondon, R. and Cheriet, M. (2015). The generation of synthetic handwritten data for improving on-line learning, in *Proc. IGS*.

Rubine, D. (1991). Specifying gestures by example, in *Proceedings of annual Conf. on Computer graphics and interactive techniques (SIGGRAPH)*.

Spano, L. D., Cisternino, A., Paternò, F. and Fenu, G. (2013). GestIT: A declarative and compositional framework for multiplatform gesture definition, in *Proceedings of ACM SIGCHI Symp. on Engineering Interactive Computing Systems (EICS)*.

Sutherland, I. E. (1963). Sketchpad: A man-machine graphical communication system, Tech. Rep. 296, Lincoln Laboratory, MIT.

Thomassen, A. J., Keuss, P. J. and van Galen, G. P. (1983). Motor aspects of handwriting, *Acta Psychol.* **54**, 1–3.

Trommershäuser, J., Maloney, L. T. and Landy, M. S. (2003). Statistical decision theory and trade-offs in the control of motor response, *Spat. Vis.* **16**, 3–4.

Ungurean, O.-C., Vatavu, R.-D., Leiva, L. A. and Martín-Albo, D. (2018a). Predicting stroke gesture input performance for users with motor impairments, in *Proceedings of ACM Conf. on Human-computer interaction with mobile devices and services (MobileHCI)*.

Ungurean, O.-C., Vatavu, R.-D., Leiva, L. A. and Plamondon, R. (2018b). Gesture input for users with motor impairments on touchscreens: Empirical results based on the Kinematic Theory, in *Proceedings of Extended Abstracts on Human Factors in Computing Systems (CHI EA)*.

van der Gon, J. J. D. and Thuring, J. P. (1965). The guiding of human movements, *Kybernetik* **14**, 1.

Vatavu, R.-D., Anthony, L. and Wobbrock, J. O. (2012). Gestures as point clouds: a $P recognizer for user interface prototypes, in *Proceedings of Intl. Conf. on Multimodal Interaction (ICMI)*.

Vatavu, R.-D., Anthony, L. and Wobbrock, J. O. (2013). Relative accuracy measures for stroke gestures, in *Proceedings of Intl. Conf. on Multimodal Interaction (ICMI)*.

Vatavu, R.-D., Anthony, L. and Wobbrock, J. O. (2018). $Q: A super-quick, articulation-invariant stroke-gesture recognizer for low-resource devices, in *Proceedings of ACM Conf. on Human-computer interaction with mobile devices and services (MobileHCI)*.

Vatavu, R.-D., Vogel, D., Casiez, G. and Grisoni, L. (2011). Estimating the perceived difficulty of pen gestures, in *Proceedings of IFIP Intl. Conf. on Human-computer Interaction (INTERACT)*.

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

Viviani, P. and McCollum, G. (1983). The relation between linear extent and velocity in drawing movements, *Neuroscience* **10**.

WB Games and Niantic (2019). Harry potter: Wizards unite, URL https://wizardsunitehub.info/spells/, accessed on Jun 2019.

Wobbrock, J. O., Wilson, A. D. and Li, Y. (2007). Gestures without libraries, toolkits or training: A $1 recognizer for user interface prototypes, in *Proceedings of Annual ACM Symp. on User Interface Software and Technology (UIST)*.

Zhai, S., Kristensson, P. O., Appert, C., Anderson, T. H. and Cao, X. (2012). Foundational issues in touch-surface stroke gesture design — an integrative review, in *Foundations and Trends in Human-Computer Interaction*, Vol. 5.