

iDeLog: Iterative Dual Spatial and Kinematic Extraction of Sigma-Lognormal Parameters

Miguel A. Ferrer, Moises Diaz, Cristina Carmona-Duarte, Réjean Plamondon

Abstract—The Kinematic Theory of rapid movements and its associated Sigma-Lognormal model have been extensively used in a large variety of applications. While the physical and biological meaning of the model have been widely tested and validated for rapid movements, some shortcomings have been detected when it is used with continuous long and complex movements. To alleviate such drawbacks, and inspired by the motor equivalence theory and a conceivable visual feedback, this paper proposes a novel framework to extract the Sigma-Lognormal parameters, namely iDeLog. Specifically, iDeLog consists of two steps. The first one, influenced by the motor equivalence model, separately derives an initial action plan defined by a set of virtual points and angles from the trajectory and a sequence of lognormals from the velocity. In the second step, based on a hypothetical visual feedback compatible with an open-loop motor control, the virtual target points of the action plan are iteratively moved to improve the matching between the observed and reconstructed trajectory and velocity. During experiments conducted with handwritten signatures, iDeLog obtained promising results as compared to the previous development of the Sigma-Lognormal.

Index Terms—Biometrics, Kinematic theory of rapid movements, Motor equivalent model, Sigma-Lognormal model, Signature

1 INTRODUCTION

Human movement modelling is of great interest for designing intelligent systems inspired by the understanding of fine motor control.

The velocity profile of human movement in general, and handwriting in particular, has been the subject of many different theories. Specifically, [1] discusses theories which rely on neural networks, behavioral models, coupled oscillator models, differential equation models, kinematic models, models exploiting minimization principles, etc. Moreover, many models exploit the properties of various mathematical functions, such as exponentials, Gaussians, beta functions, splines, etc., to reproduce human movements.

Among the models which provide analytical representations, the Kinematic theory of rapid human movements and its Delta and Sigma-Lognormal models have been extensively used to explain most of the basic phenomena reported in classical studies on human motor control of rapid and automated movements [2][3][4][5].

The Sigma-Lognormal model is based on the hypothesis that the velocity of a neuromuscular system can be modelled by a vector summation of a certain number of lognormal functions, each of them defined by six parameters [6].

Over the last 10 years of uninterrupted use, the model

has been challenged by many researchers, applying it to a wide variety of practical applications such as the development of Automatic Signature Verification systems [7], the generation of duplicated signatures to improve training [8] or the improvement of forgery detection [9]. It has also been used to synthesize Western [10][11][12][13] and Indian [14] signatures. Beyond the handwritten signature field, this model has been successfully applied to the following: the study and understanding of the evolution and acquisition of handwriting skills [15], the development of tools to help children learn handwriting [16], neuromuscular health care [17], the examination of biomedical systems for detecting fine motor control problems associated with brain strokes [18] and Parkinson's disease [19], as well as to the characterization of the cranio-caudal signature of a turn [20]. Recently, the Kinematic theory of rapid movements has been applied to many different areas, such as handwriting generation [21], CAPTCHA [22], graffiti design [23], mouse movement analysis [24], gesture generation [25] and the study of articulation in voice processing [26], among others.

Most of the above research calculate the Sigma-Lognormal parameters through the Robust XZERO algorithm implemented by the ScriptStudio application, which was proposed in 2007 [27][6].

1.1 Our proposal

Without questioning the efficacy of Sigma-Lognormal model, which works accurately with automated and rapid movements, some shortcomings have been seen when extracting the parameters of continuous long and complex movements. Incidentally, the main drawback observed has been a certain degree of drift in the trajectory reconstruction. In previous works, these issues are partially addressed by cutting the long pieces of handwriting into smaller pieces

This study was funded by the Spanish government's MINECO TEC2016-77791-C4-1-R research project and European Union FEDER program/funds and by the NSERC-Canada Grant RGPIN-2015-06409 to R. Plamondon. C. Carmona-Duarte is supported by a Juan de la Cierva (IJC-2016-27682) from the Spanish MINECO.

Miguel A. Ferrer and Cristina Carmona-Duarte are with the Instituto Universitario para el Desarrollo Tecnológico y la Innovación en Comunicaciones, Universidad de Las Palmas de Gran Canaria, Campus de Tafira, Las Palmas de Gran Canaria, Spain. Emails: {mferrer, ccarmona}@idetec.eu.

Moises Diaz is with Universidad del Atlántico Medio, Las Palmas de Gran Canaria, Spain. Email: mdiazc@unidam.es.

Réjean Plamondon is with École Polytechnique, Université de Montréal, Montréal, P.Q., Canada. Email: rejean.plamondon@polymtl.ca.

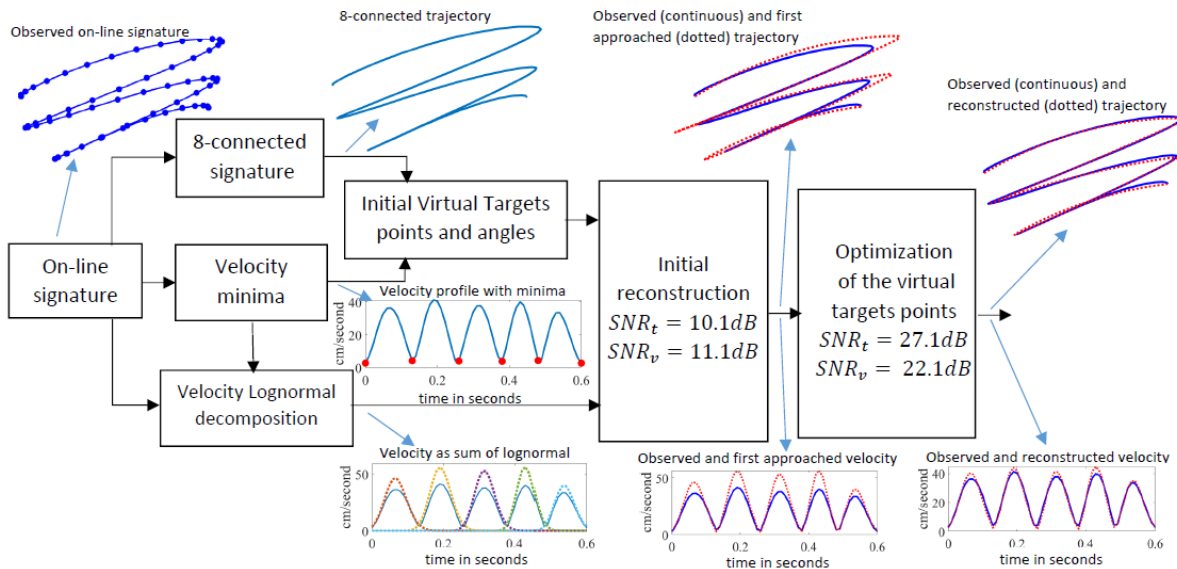


Fig. 1. Block diagram of iDeLog framework to calculate the Sigma-Lognormal parameters of an on-line signature. Solid blue lines represent observed movement, while dotted red lines represent reconstructed movement.

and reconstructing each of them independently to ensure better reconstruction rates [7][8] [10][11][28]. Following a new avenue, this paper aimed to establish a new framework to extract the Sigma-Lognormal parameters relying on the motor equivalence theory and a *conceivable* visual feedback in long and complex movements.

It should be noted that it is not the goal of this work to confirm or deny the validity of the motor equivalence theory or to look into the subtle integration of a possible visual feedback in automated human movements. Rather, our research aims to provide new insights into a more robust parameter extraction of the Sigma-Lognormal model from a novel point of view in the hopes of advancing the frontiers of the Sigma-Lognormal model inspired by such hypotheses.

The motor equivalence model, also known as the Degrees of Freedom (DoF) problem [29][30], suggests that the brain stores the movements aimed at performing a single task in two steps. The first step is *effector-independent*, and stores the movement in an abstract form as a spatial sequence of points representing the action plan. The second step is *effector-dependent*, and consists in a sequence of motor commands directed at obtaining particular muscular contractions and articulatory movements in order to execute a given action plan [31].

Additionally, in a rapid and short movement, it is conceivable that the planned and executed movements are matched. However, in long movements, there will be a likely drift between the planned movement and its execution, which would require some corrections. [32] looks at the interaction between visual feedback and automated handwriting movements, and asserts that the visual feedback does not abort a movement when a drift exists in the trajectory. Rather, a new stroke is added to correct the trajectory towards the new target. Thus, the visual feedback does not slow down open-loop movements to allow control of the motor output in a closed-loop mode.

The motor equivalence hypothesis, which divides human physical action into the cognitive plan and motor control, can be integrated into the Sigma-Lognormal model by working out its parameter on the basis of a dual spatial and kinematic calculation. As such, on the one hand, the action plan defined by the virtual points and its angle would be calculated from the observed trajectory. On the other hand, the remaining parameters would be deduced from the observed velocity.

The idea that visual feedback can modify a future stroke when the trajectory drifts without slowing down the open-loop motor control is useful for correcting the drift between the reconstructed and observed movements. We consider such a hypothetical visual feedback in the parameter extraction procedure of the Sigma-Lognormal model, which relies on well-learned automatic human movements governed by open-loop motor control without a significant visual feedback influence [33]; to that end, we propose an optimization stage of the initial set of parameters extracted by moving the virtual target points. As such, there is a significant improvement in the trajectory reconstruction, which is at the expense of a slight degradation of the velocity reconstruction.

On the bases of the above, the procedure for working out the Sigma-Lognormal parameters is completely reformulated. The novel procedure, which we call iDeLog in this work, calculates the 8-connected trajectory and speed profile of a given long and complex movement. Influenced by the motor equivalence model, the action plan, the total number of virtual target points, and the angles of their circular trajectories are dually derived from the trajectory, and the velocity decomposed as a sum of weighed lognormals. At this point, an initial reconstruction is obtained. Applying visual feedback principles, the reconstructed movement is iteratively optimized by moving the virtual target points with the ensuing changes of the angles and lognormal parameters. A block diagram of the procedure is shown in Figure 1.

The remainder of the paper is organized as follows: Section II briefly reviews the Sigma-Lognormal model and parameter extraction. Section III is devoted to iDeLog, while its evaluation and the related discussion are covered in Section IV. Section V concludes the article.

2 SIGMA-LOGNORMAL MODEL: A BRIEF REVIEW

2.1 The Sigma-Lognormal Model

The Kinematic Theory of rapid movements [2], from which the Sigma-Lognormal model was developed [6], considers that complex human movements consist of overlapping strokes. Each stroke has a lognormal-shaped velocity profile $v_j(t)$ defined as:

$$v_j(t; t_{0j}, \mu_j, \sigma_j^2) = D_j \Lambda(t; t_{0j}, \mu_j, \sigma_j^2) = \frac{D_j}{\sigma_j \sqrt{2\pi}(t - t_{0j})} \exp\left(-\frac{[\ln(t - t_{0j}) - \mu_j]^2}{2\sigma_j^2}\right) \quad (1)$$

where t is the time, t_{0j} the time of stroke occurrence, D_j the amplitude of the stroke, μ_j the stroke time delay and σ_j the stroke response time, both on a logarithmic time scale.

The overlapping of these lognormals can produce a complex trajectory from a hidden trajectory action plan. Such an action plan consists of a sequence of virtual target points linked together by circular arcs. Each arc is produced as a response of the motor system to a set of rhythmic commands from the cerebellum. The overlap in time of these movements results in:

$$\vec{v}(t) = \begin{bmatrix} v_x(t) \\ v_y(t) \end{bmatrix} = \sum_{j=1}^N D_j \begin{bmatrix} \cos(\phi_j(t)) \\ \sin(\phi_j(t)) \end{bmatrix} D_j \Lambda(t; t_{0j}, \mu_j, \sigma_j^2) \quad (2)$$

where N denotes the number of lognormals, or strokes, and $\phi_j(t)$ the angular position.

$$\phi_j(t) = \theta_{s_j} + \frac{\theta_{e_j} - \theta_{s_j}}{2} \left[1 + \operatorname{erf}\left(\frac{\ln(t - t_{0j}) - \mu_j}{\sigma_j \sqrt{2}}\right) \right] \quad (3)$$

where θ_{e_j} and θ_{s_j} are respectively the starting angle and the ending angle of the arc that links the two virtual target points corresponding to the j^{th} stroke. Note that this formula describes the sweep from θ_{s_j} to θ_{e_j} in a lognormal timing. Finally, the trajectory is calculated as:

$$\vec{s}(t) = \sum_{j=1}^N \frac{D_j}{\theta_{e_j} - \theta_{s_j}} \begin{bmatrix} \sin(\phi_j(t)) - \sin(\theta_{s_j}) \\ -\cos(\phi_j(t)) + \cos(\theta_{s_j}) \end{bmatrix} \quad (4)$$

This formula converts angles into circular arcs and overlaps them. Specifically, the j^{th} term of the summation represents the arc of the circumference that links the virtual target points tp_{j-1} and tp_j . The radius of this circumference is $D_j / (\theta_{e_j} - \theta_{s_j})$, and D_j coincides with the arc length. In this case, the virtual target points are defined by:

$$tp_j = tp_{j-1} + \frac{D_j}{\theta_{e_j} - \theta_{s_j}} \begin{bmatrix} \sin(\phi_j(T)) - \sin(\theta_{s_j}) \\ -\cos(\phi_j(T)) + \cos(\theta_{s_j}) \end{bmatrix} \quad (5)$$

where T is the duration of the signature.

2.2 Stroke Extraction

The procedure proposed in [27][6] starts by calculating

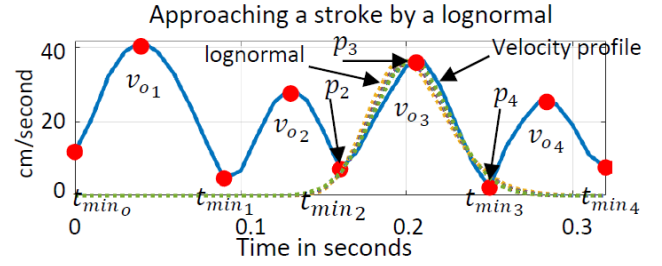


Fig. 2. Four lobes of an observed velocity profile with the points p_2 , p_3 and p_4 used to calculate the lognormal of the third lobe. Red circles: maximum and minimum from the observed velocity profile. Dotted line: different lognormals approaching the third velocity lobe calculated from different t_0 . All the lognormals are so similar that the figure cannot tell them apart.

the magnitude of the velocity profile. To enhance the quality of the observed velocity signal, it is re-sampled at 200 Hz through cubic splines, and smoothed by a Chebyshev filter [6][28].

The strokes are extracted according to the following iterative procedure:

1. The strokes are firstly identified in the velocity magnitude profile according to the lobes. In short, a lognormal is defined by its maximum p_3 and the minimum of the velocity lobes p_2 and p_4 . Figure 2 exemplifies this step for the third lognormal.
2. The values of t_{0j} , μ_j and σ_j are obtained from p_2 , p_3 and p_4 . Because any velocity lobe can be approached similarly by many lognormals, as it is shown in Figure 2, if the values of μ_j and σ_j are out of the expected range, the values p_2 and p_4 are recalculated. Noise can generate false positive strokes, and therefore, if the area and the maximum of the lognormal are lower than a specific threshold, the lognormal is discarded.
3. Once a stroke is identified, its velocity is reconstructed and removed from the observed velocity profile. This subtraction allows to uncover potential strokes whose local maxima are hidden by a faster lognormal neighbor.
4. The similarity between the preprocessed observed velocity $v_o(t)$ and the reconstructed velocity profile $v_r(t)$ is calculated by means of the signal-to-noise ratio, defined as:
$$SNR_v = 10 \log \left(\frac{\int_{t=0}^T v_o(t)^2 dt}{\int_{t=0}^T (v_o(t) - v_r(t))^2 dt} \right) \quad (6)$$
5. The procedure then returns to point 1 and, considering the subtracted velocity as the input velocity, the process is repeated, until the SNR_v is larger than a given threshold or the current number of strokes is larger than a given maximum, or there are no sets of p_2 , p_3 and p_4 points left.

2.3 Stroke Parameter Estimation

The Robust XZERO algorithm [34] is used to estimate the lognormal parameters that describe the velocity profile of each stroke. Let s_j be a stroke and p_2 , p_3 and p_4 its characteristic points, which occur at times $\{t_\alpha\}_{\alpha=2}^4$ and

$\{v_o(t_\alpha)\}_{\alpha=2}^4$, their corresponding velocity values. The lognormal parameters τ_j , μ_j , σ_j and D_j that fit $\{v_o(t_\alpha)\}_{\alpha=2}^4$ can be derived using different two-element combinations of the lognormal characteristic points such as:

$$\sigma^2 = \begin{cases} -2 - 2\log(r_{\alpha\beta}) - \frac{1}{2\log(r_{\alpha\beta})} & \text{if } \alpha = 2, \beta = 3 \\ -2 + 2\sqrt{1 + \log^2(r_{\alpha\beta})} & \text{if } \alpha = 2, \beta = 4 \\ -2 - 2\log(r_{\beta\alpha}) - \frac{1}{2\log(r_{\beta\alpha})} & \text{if } \alpha = 3, \beta = 4 \end{cases} \quad (7)$$

where \log is the natural logarithm and $r_{\alpha\beta} = \|\vec{v}_o(t_\alpha)\| / \|\vec{v}_o(t_\beta)\|$.

$$\mu = \log\left(\frac{t_\alpha - t_\beta}{e^{-a_\alpha} - e^{-a_\beta}}\right) \quad (8)$$

$$t_0 = t_\alpha - e^{\mu - a_\alpha} \quad (9)$$

$$D = \|\vec{v}_o(t_\alpha)\| \sigma \sqrt{2\pi} \exp\left(\mu + \frac{a_\alpha^2}{2\sigma^2}\right) - a_\alpha \quad (10)$$

where $\alpha, \beta \in \{2, 3, 4\}$, $\alpha < \beta$ and

$$a_i = \begin{cases} \frac{3}{2}\sigma^2 + \sigma\sqrt{\frac{\sigma^2}{4} + 1} & \text{if } i = 2 \\ \sigma^2 & \text{if } i = 3 \\ \frac{3}{2}\sigma^2 - \sigma\sqrt{\frac{\sigma^2}{4} + 1} & \text{if } i = 4 \end{cases} \quad (11)$$

The parameters are computed using all possible combinations. The set of parameters which minimizes the least-square error is kept as the solution.

2.4 Angle Estimation

The XZERO calculates the start θ_s and end θ_e angles that link two virtual target points as follows:

$$\theta_s = \phi(t_3) - \Delta\phi(d(t_3) - d(t_1)) \quad (12)$$

$$\theta_e = \phi(t_3) - \Delta\phi(d(t_5) - d(t_3)) \quad (13)$$

where

$$\Delta\phi = \frac{\phi(t_4) - \phi(t_2)}{d(t_4) - d(t_2)} \quad (14)$$

$$d(t_i) = \begin{cases} 0 & \text{if } i = 1 \\ \frac{D}{2} \left[1 + \operatorname{erf}\left(-a_i / \sigma\sqrt{2}\right) \right] & \text{if } i = 2, 3, 4 \\ D & \text{if } i = 5 \end{cases} \quad (15)$$

3 iDeLog: NOVEL SIGMA-LOGNORMAL EXTRACTOR

The Robust XZERO is a velocity-based procedure, which is prone to spatial deviations as the observed movement gets longer. This is due to the fact that small deviations in the velocity estimation are propagated over the entire movement when integrating the velocity to obtain the trajectory, thus resulting in an increased spatial deviation [28].

To alleviate this drawback, we rely on the motor equivalence model and a *conceivable* visual feedback paradigm. At a first step, a procedure to separately calculate the spatial and kinematics Sigma-Lognormal parameters is devised from the motor equivalence model hypothesis. Thus, the deviations of the modeled velocity are compensated by

TABLE 1. Mathematical notations

Symbol	Meaning
N	Number of virtual target points
T	Temporal length of the signature
$\{D_j, t_{0j}, \mu_j, \sigma_j^2\}_{j=1}^N$	Sigma-Lognormal parameters of stroke j
$\{\theta_s, \theta_e\}_{j=1}^N$	Starting and ending angles of the link between tp_{j-1} and tp_j
$\{tp_j\}_{j=0}^N$	Virtual Target point of stroke j
$\{sp_j\}_{j=0}^N$	Salient points (velocity minima) of original signature
$\{spr_j\}_{j=0}^N$	Salient points of reconstructed trajectory
$\{t_{min_j}\}_{j=0}^N$	Time of the salient points $\{sp_j\}_{j=0}^N$ and $\{spr_j\}_{j=0}^N$
$\{x_o(t), y_o(t)\}_{t=0}^T$	Samples of the on-line original trajectory
$\{x_r(t), y_r(t)\}_{t=0}^T$	Samples of the on-line reconstructed trajectory
$v_o(t)$	Velocity profile of the original movement
$v_r(t)$	Velocity profile of the reconstructed movement
$\{v_j(t)\}_{j=1}^N$	Velocity profile of stroke j , $0 < t < T$
SNR_v, SNR_t	Signal-to-Noise Ratio between original and reconstructed velocity and trajectory, respectively

the spatial parameters, avoiding trajectory drift. At a second step, from the visual feedback compatible with the open-loop motor control hypothesis, the spatial parameters of the Sigma-Lognormal model are optimized to improve the fitting between the observed and reconstructed movements. Thus, a dual trajectory and velocity optimization are carried out.

This section describes these new procedures based on the above underlying ideas. Specifically, we describe the stroke segmentation, the extraction of the spatial and kinematic Sigma-Lognormal parameters and their posterior optimization. We call this new framework 'iDeLog', which stands for *iterative Decomposition in Lognormals*. The mathematical notations used are presented in Table 1.

3.1 Stroke Segmentation

The strokes are estimated from the magnitude of the velocity. Contrary to [27][6], iDeLog does not need to resample and smooth the input signal. However, for the sake of comparison with [27][6], an option to smooth the input signal has been added to iDeLog.

Once the velocity magnitude is, the strokes are identified as the velocity peaks between velocity minima. Specifically, let the time of the velocity minima be $\{t_{min_j}\}_{j=0}^N$, where N is the number of strokes, $t_{min_0} = 0$, and $t_{min_N} = T$ the duration of the observed rapid movement, as can be seen in Figure 2. In this case, the velocity peak which corresponds to stroke j is defined as:

$$v_{oj}(t) = \begin{cases} v_o(t) & t_{min_{j-1}} \leq t \leq t_{min_j}, 1 < j < N \\ 0 & \text{otherwise,} \end{cases} \quad (16)$$

where $v_o(t)$ is the observed velocity signal. In the trajectory, the velocity minima correspond to the salient points $sp_j, j = 0, \dots, N$, where sp_0 is the first sample, sp_N is the last sample and sp_j is the sample $t_{min_j} \cdot f_m$, where f_m is the sampling frequency.

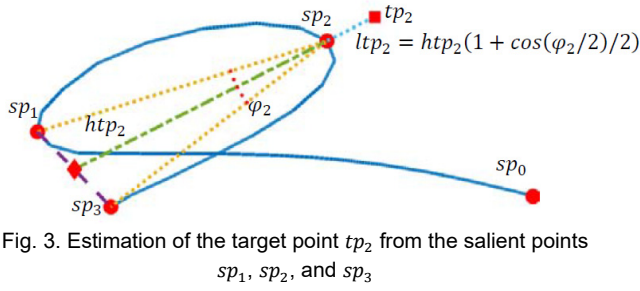


Fig. 3. Estimation of the target point tp_2 from the salient points sp_1 , sp_2 , and sp_3

3.2 Kinematic Parameter Extraction

This section approaches each velocity peak as a lognormal function with parameters t_{0j} , μ_j and σ_j . As shown in Figure 2, there are many lognormals that reasonably fit a given peak unless we fix the initial time t_{0j} beforehand. Therefore, we first calculate t_{0j} based on a biological argument, and later, μ_j and σ_j .

Estimation of t_{0j}

It is well known that the movement action originates in the cortex, and then passes through the Basal Ganglia, which decodes the message in order to activate the different pools of neurons which innervate the muscles to carry out a particular movement. In a well-learned movement, it makes sense to assume that the times between the movement actions are originated in the cortex. Thus, the movement action carried out by the motor system should be similar for each stroke. In our estimation procedure, this transmission time is represented by $t_{min,j-1} - t_{0j}$, which is approximately 0.5 seconds [35]. Specifically, in our case, $t_{0j} = t_{min,j-1} - 0.5$.

Initial Estimation of μ_j and σ_j^2

iDeLog tries to fit the velocity peak $v_{oj}(t)$ with the lognormal $v_j(t; t_{0j}, \mu_j, \sigma_j^2)$ through a nonlinear least-squares minimization. The function to minimize is:

$$\hat{\mu}_j, \hat{\sigma}_j^2 = \underset{\mu_j, \sigma_j^2}{\operatorname{argmin}} \int_{t=t_{min,j-1}}^{t_{min,j}} |v_j(t; t_{0j}, \mu_j, \sigma_j^2) - v_{nj}(t)|^2 dt \quad (17)$$

where

$$v_{nj}(t) = v_{oj}(t) / \int_{t=0}^T v_{oj}(t) dt \quad (18)$$

since the area of the lognormal function is equal to 1.

The minimization is performed by means of a Levenberg-Marquardt Algorithm (LMA), which is used in many applications in solving generic curve-fitting problems [36].

3.3 Spatial Parameters Extraction

The spatial parameters of the Sigma-Lognormal model refer to the virtual target points tp_j and their starting and ending angles, θ_{sj} and θ_{ej} , respectively. From these parameters, the trajectory is built up by overlapping the arc of the circumferences linking consecutive virtual target points.

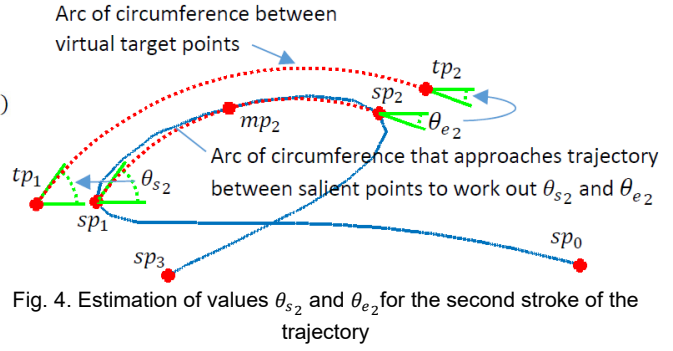


Fig. 4. Estimation of values θ_{s2} and θ_{e2} for the second stroke of the trajectory

Thus, a virtual target point involves two arcs of circumference: one that goes from the target point tp_{j-1} to tp_j and another that goes from the target point tp_j to tp_{j+1} . The first one ends at tp_j with ending angle θ_{ej} while the second one starts at tp_j with starting angle θ_{sj} . The overlap between the circumference arcs generates a trajectory with a higher curvature, or salient point, around the virtual target point. The value of D_j is determined by the length of the circumference arcs between virtual target points.

Note that virtual target points are also related to velocity minima, which generally coincide with salient points. A virtual target point marks the end of a stroke and the start of the next one: the overlap of the speed down to the end of a stroke (tail of the lognormal) with the speed up to the start of the next stroke (initial rise of the lognormal) generates a velocity minimum.

Location of the Initial Virtual Target Points tp_j

iDeLog estimates the initial virtual target points directly from the observed trajectory. Each salient point sp_j is associated with the virtual target points tp_j , $j = 0, \dots, N$. In iDeLog, $tp_0 = sp_0$ and $tp_N = sp_N$ which are the first and last points of the trajectory, respectively. The virtual target point tp_j is calculated using sp_{j-1} , sp_j and sp_{j+1} , $j = 1, \dots, N-1$, which form a triangle. The initial virtual target point is located on the median of the vertex sp_j , which is a straight line through the vertex sp_j and the midpoint $(sp_{j-1} + sp_{j+1})/2$ of the opposite side, at a distance ltp_j from the vertex sp_j defined as:

$$ltp_j = htp_j(1 + \cos(\varphi_j/2)/2) \quad (19)$$

where htp_j is the distance between the vertex sp_j and the midpoint of the opposite side, and φ_j is the angle of the vertex sp_j . Thus, when using the cosine function, the sharper the angle of the vertex sp_j , the farther tp_j is from sp_j . An example of this procedure is shown in Figure 3.

Estimation of Starting θ_{sj} and Ending θ_{ej} Angles

The angles of the circumferences that link virtual target points are defined by their starting θ_{sj} and ending θ_{ej} angles $\forall j = 1, \dots, N$, where N is the number of strokes. As the starting and ending angles are constitute a spatial characteristic of the signature, iDeLog estimates these parameters from the observed spatial trajectory.

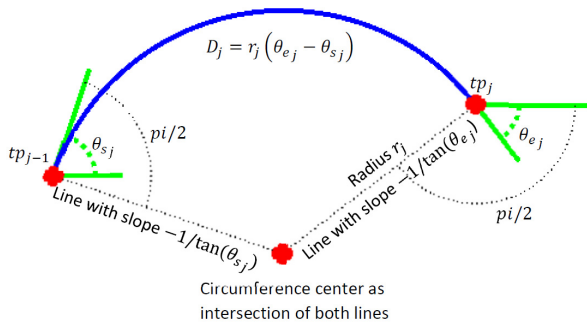


Fig. 5. Procedure to calculate D_j , given the target points tp_{j-1} , tp_j , and their start and end angles θ_{s_j} and θ_{e_j}

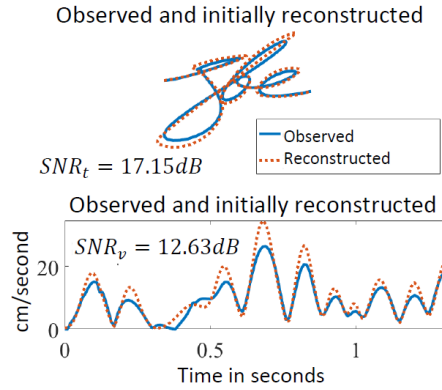


Fig. 6. Observed and reconstructed trajectory and velocity of a signature with initial parameters

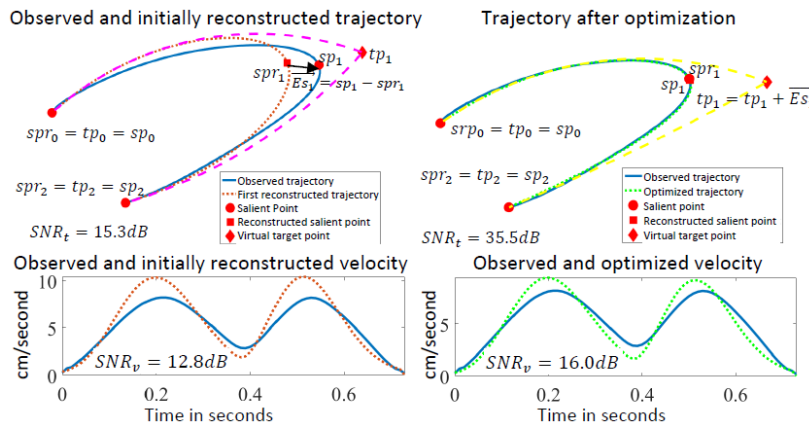


Fig. 7. Left: Observed and reconstructed trajectory, salient points, virtual target points and arcs of circumference obtained with the initial parameters. Right: Observed and reconstructed signature after virtual target point displacement.

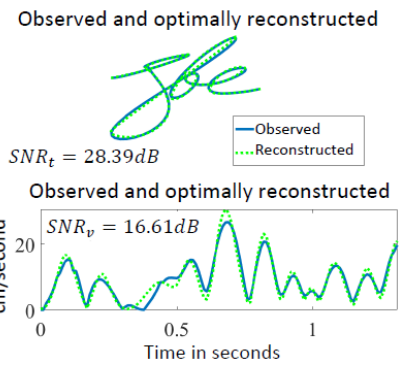


Fig. 8. Observed and reconstructed trajectory and velocity after optimization

They are calculated as follows:

1. The middle point mp_j of the trajectory of the j^{th} stroke between the salient points sp_{j-1} and sp_j is calculated. The term *middle point* refers to the fact the distance on the trajectory from mp_j to sp_{j-1} and to sp_j is equal.
2. A circumference that passes by these three points is obtained.
3. The angle θ_{s_j} is computed as the angle of the tangent to the circumference at sp_{j-1} . Then, the angle θ_{e_j} is obtained as the angle of the tangent to the circumference at sp_j . This procedure is illustrated in Figure 4.

3.4 Reconstruction of the Trajectory

From the above procedure, we have an initial gross estimation of the parameters of the Sigma-Lognormal model $t_{0j}, \mu_j, \sigma_j^2, tp_j, \theta_{s_j}$ and $\theta_{e_j}, j = 1, \dots, N$ from the observed samples $\{x_o(t), y_o(t)\}, 0 < t < T$. The parameters allow us to calculate the parameter D_j and obtain the reconstructed spatial trajectory $T_r(t) = \{x_r(t), y_r(t)\}$ following equations (3) and (4). The salient points of the reconstructed spatial trajectory $spr_j, j = 0, \dots, N$ are obtained through the minima of the reconstructed velocity $v_r(t)$, which is the derivative of the reconstructed trajectory.

Estimation of D_j

The value of the lognormal amplitude D_j describes the amplitude of the movement, and is defined without ambiguity by the position of the virtual target points $tp_{j-1}, tp_j, \theta_{s_j}$ and θ_{e_j} . It is calculated as:

$$D_j = r_j (\theta_{e_j} - \theta_{s_j}), \quad j \in 1, \dots, N \quad (20)$$

where r_j is the radius of the circumference that goes from tp_{j-1} to tp_j . To calculate r_j , we first calculate the center of the circumference as the intersection of the line that crosses tp_{j-1} with slope $-1/\tan(\theta_{s_j})$ and the line that crosses tp_j with slope $-1/\tan(\theta_{e_j})$. Then, r_j is the distance from the center of the circumference to either tp_{j-1} or tp_j . This procedure is illustrated in Figure 5.

It should be pointed out that the Sigma-Lognormal model overdefines the circumference with five parameters: $D_j, tp_{j-1}, tp_j, \theta_{s_j}$ and θ_{e_j} , whereas a circumference only requires three parameters.

An example of a recovered spatial trajectory with these initial gross parameters is shown in Figure 6.

3.5 Step 2: Optimization

The basic idea underlying the optimization is to update the position of the virtual target points $tp_j, j = 1, \dots, N - 1$ to improve both the reconstructed trajectory and velocity

profile simultaneously. Thus, a sort of visual feedback is included in the procedure without slowing down the open-loop motor control.

The improvement is done by means of an iterative Least Mean Squares (LMS) algorithm applied to each stroke. Because a movement at a virtual target point modifies the entire trajectory, adapting all of them at the same time would mean that the iterative algorithm would not converge. For the sake of convergence, the optimization has to be carried out stroke by stroke in the same order as a natural human movement.

The optimization is conducted as follows:

1. Set $j = 1$.
2. Refine the j^{th} virtual target point.
3. Calculate the error or difference \vec{ES}_j between the observed and reconstructed trajectory salient points $\vec{ES}_j = sp_j - spr_j$.
4. Update the virtual target point as $tp_j = tp_j + \eta \vec{ES}_j$, where $0 < \eta < 1$. The value $\eta = 1$ represents the fastest fitting.
5. With this new virtual target point, reconstruct the entire trajectory $T_r(t) = \{x_r(t), y_r(t)\}$ and calculate the new salient points $spr_j, j = 1, \dots, N - 1$.
6. Set $j = j + 1$ and go to 2 if $j < N$.

An illustration of this procedure can be found in Figure 7. As can be seen, there is a significant improvement of the reconstructed trajectory and velocity. The improvement of the optimization over a signature is shown in Figure 8.

This optimization can be repeated as many times as necessary. In our experiments, the optimization barely improves the SNR of the reconstructed trajectory and velocity after two iterations when $\eta = 1$.

The iDeLog is summarized in algorithm 1.

4. EXPERIMENTS

Our evaluation here aimed to measure the ability of iDeLog to extract the Sigma-Lognormal parameters of continuous long and complex human movements such as a handwritten signature. These are supposed to be executed rapidly, and involve a large number of muscles, which is consistent with the underlying hypotheses of the Kinematic Theory of rapid human movements. The experiments were performed with genuine signatures from existing databases, as these are well-learned movements written naturally. This avoided biased measures due to constrained and sluggish movements of imitated signatures.

The ability of iDeLog to reconstruct a handwritten signature will be compared with [27][6] through ScriptStudio. It is an application used to automatically extract the Sigma-Lognormal parameters of rapid and complex human movements. Its robustness and flexibility have been demonstrated in numerous studies, as stated in Section 1. The comparison is performed in terms of Signal-to-Noise Ratio and Equal Error Rate over several publicly available databases and Automatic Signature Verifiers.

Algorithm 1. iDeLog for Sigma-Lognormal parameter extraction

```

% Input: On-line samples of an observed signature: (xo, yo)
% Output: Sigma-Lognormal parameters (t0j, μj, σj, tpj, θs,j, θe,j)
[xc, yc] = 8 - connected_trajectory(xo, yo)
vo = Velocity_profile(xo, yo)
N = Number_of_velocity_minima(vo)
tminj = Time_of_velocity_minima(vo)
% Location of velocity minima on 8-connected trajectory
spj = SampleLocation_on_8c_trajectory(tminj, xo, yo, xc, yc)
% Velocity profiles between velocity minima
voj = Velocity_lobe(vo, tminj)
% First step: initial set of Sigma-Lognormal parameters.
For j = 1: N % For each stroke j
    % Lognormal parameters from velocity
    t0j = Temporal_offset(tminj)
    [μj, σj] = Levenberg-Marquardt_Algorithm(voj)
    % Lognormal parameters from trajectory
    % Virtual points tpj, angles θs,j, θe,j and areas Dj
    tpj = Triangularization(spj, xc, yc)
    [θs,j, θe,j] = Trajectory_slope(tpj, spj, xc, yc)
    Dj = Lognormal_area(tpj, θs,j, θe,j)
    [xr, yr, sprj] = Reconstruct_trajectory(t0j, μj, σj, tpj, θs,j, θe,j)
end
% Second step: parameter refinement by moving virtual target points
For j = 1: N
    ESj = spj - sprj % Error at velocity minimum
    tpj = tpj + η ESj % Update virtual target point
    [θs,j, θe,j] = Trajectory_slope(tpj, spj, xc, yc)
    Dj = Lognormal_amplitudes(tpj, θs,j, θe,j)
    [xr, yr, sprj] = Reconstruct_trajectory(t0j, μj, σj, tpj, θs,j, θe,j)
end

```

4.1 Databases

Three on-line signature databases were selected. The first were real on-line genuine signatures from BiosecureID-SONOFF [37], consisting of 16 specimens from 132 users. The signatures were acquired by a WACOM tablet. The files include the x - and y -coordinates of the trajectory and the pressure. The sampling frequency is 100 samples per second.

The second database was the MCYT100 sub-corpus [38]. This comprises the 100 first users of the full MCYT-330 database, whose signatures were also captured by a WACOM tablet. It contains 25 genuine signatures per user. The database provides the x - and y -coordinates of the trajectory and the pressure. The sampling frequency is 100 Hz.

The BiosecureID-SONOFF and MCYT databases were selected since they also include pen-ups. Thus, we could test the performance of iDeLog in reconstructing continuous long and complex movements.

The third database was the SUSIGVisual sub-corpus [39]. This database contains 94 users, with 20 genuine signatures per user. The information provided for each signature are the x - and y -coordinates of the trajectory and the pressure. The sampling frequency is 100 Hz. This sub-corpus was collected with an LCD touch device which does not capture pen-up movements. As a result, there was some jitter in the sampling period, leading to noisy lognormals.

The SUSIGVisual database does not contain pen-ups, but rather, a concatenated sequence of pen-downs. In such

TABLE 2
RESULTS OF SCRIPTSTUDIO AND IDELOG

Database	Procedure	SNR_t	σ_{SNR_t}	$SNR_t / NbLog$	SNR_v	σ_{SNR_v}	$SNR_v / NbLog$
BiosecureID	ScriptStudio	6.47dB	5.48	0.283	16.35dB	4.88	0.740
	iDeLog	23.28dB	5.67	0.880	15.22dB	1.19	0.569
MCYT100	ScriptStudio	6.57dB	5.50	0.298	16.02dB	4.94	0.698
	iDeLog	21.33dB	5.30	0.905	15.22dB	1.21	0.630
SUSIG Visual	ScriptStudio	16.39dB	7.02	0.808	17.78dB	4.27	0.881
	iDeLog	24.11dB	5.49	1.338	15.19dB	1.25	0.841

a situation, when we compare iDeLog and ScriptStudio, the ScriptStudio performance appears to be better [28].

4.2 Quantitative Measures

The reconstruction results are given in the following terms: (1) Signal-to-Noise Ratio (SNR), and (2) the SNR and the number of lognormal detected ratio ($SNR/NbLog$) for both trajectory and velocity.

The Signal-to-Noise Ratio for trajectory, SNR_t , measures the difference between the observed and reconstructed trajectories. If the observed trajectory is preprocessed, the measure compares the observed preprocessed trajectory with the reconstructed one. Thus, we have:

$$SNR_t = 10 \log \left(\frac{\int_{t=0}^T (x_{op}(t)^2 + y_{op}(t)^2) dt}{\int_{t=0}^T (x_{or}(t)^2 + y_{or}(t)^2) dt} \right) \quad (21)$$

where the numerator $x_{op}(t) = x_o(t) - \bar{x}_o(t)$, $y_{op}(t) = y_o(t) - \bar{y}_r(t)$ and the denominator $x_{or}(t) = x_o(t) - x_r(t)$, $y_{or}(t) = y_o(t) - y_r(t)$. The subindexes o and r denote respectively the observed and the reconstructed signal.

The Signal-to-Noise Ratio for velocity SNR_v is the one used by ScriptStudio. This measure is considered to index the lognormality of the movements. As these experiments dealt with genuine signatures which are supposed to be lognormal, given that they are handwritten fluently, the higher the SNR obtained, the better the reconstruction. The SNR_v is defined as:

$$SNR_v = 10 \log \left(\frac{\int_{t=0}^T v_o(t)^2 dt}{\int_{t=0}^T (v_o(t) - v_r(t))^2 dt} \right) \quad (22)$$

The ratio of the SNR_v and the number of lognormals detected $SNR_v / NbLog$ has been proposed as a variable that reflects the writer's ability to make regular movements [40][41]. In evaluating a procedure, such a ratio takes into account the number of lognormals required by the procedure to reach the given SNR_v . Similarly, this is defined by $SNR_t / NbLog$.

The quantitative measures described in this section were obtained for each single specimen. The averaged results and their standard deviation are provided for each experiment. For each result, a balanced two-way ANOVA analysis was run to check for the significance of the observed differences, based on $p < 0.05$ values.

4.3 Experiment 1: ScriptStudio and iDeLog

For a fair comparison between ScriptStudio and iDeLog, the observed signal used as input to both procedures

TABLE 3
MEAN AND STANDARD DEVIATION OF THE NUMBER OF LOGNORMALS DETECTED BY SCRIPTSTUDIO AND IDELOG

Database	ScriptStudio		iDeLog with/without smoothing			
	$NbLog$	σ_{NbLog}	with		without	
			$NbLog$	σ_{NbLog}	$NbLog$	σ_{NbLog}
BiosecureID	26.34	11.37	36.20	18.58	37.84	19.60
MCYT100	28.24	12.18	36.50	19.41	44.31	25.04
SUSIGVisual	22.82	8.26	21.77	10.10	39.61	21.24

should be the same. Consequently, the input to iDeLog and ScriptStudio are interpolated and smoothed. Furthermore, to measure the ability to decompose continuous long complex movements, both iDeLog and ScriptStudio were fed with the whole signature, including pen-ups.

The results are shown in Table 2. It is worth pointing out that the $SNRs$ in Table 2 are calculated in the same way as ScriptStudio, i.e., as the ratio between the reconstructed signal and the preprocessed observed one. The ANOVA analysis comparing ScriptStudio and iDeLog resulted in p -values lower than 0.05 in all cases shown in Table 2.

As can be seen, the performance of ScriptStudio in fitting the velocity profile is significantly better than that of iDeLog, but the iDeLog does a significantly better job adjusting the trajectory. Furthermore, the standard deviation of SNR_v is meaningfully lower in iDeLog than ScriptStudio, which means iDeLog is a more stable procedure.

The slight decrease in SNR_v of iDeLog is attributed to the harder constraint of fitting the trajectory and velocity concurrently. Conversely, this condition is believed to keep the standard deviation lower.

It is worth mentioning that, as expected, the performance of ScriptStudio on the SUSIGVisual database was significantly better than with the MCYT100 or BiosecureID-SONOFF. This was due to the fact that the SUSIGVisual database does not contain pen-ups. This helps ScriptStudio to segment the signature in shorter pieces, therefore reducing the cumulative error in the velocity.

The question which arises is, since the trajectory is the integral of the velocity, how could a similar SNR_v therefore result in such a different SNR_t ? The answer seems to be related to the distribution of the error in the reconstructed velocity.

Indeed, when adjusting the virtual target points in iDeLog, an error in the reconstruction of the velocity produces a new velocity deviation to compensate for the drift of the trajectory and to keep both trajectory and velocity errors as low as possible. It can therefore be said that the error is distributed along the whole signature. In the case of ScriptStudio, an error in the velocity is not compensated since the trajectory reconstruction is not checked. Thus, a missing or an extra lognormal or a deficient fit to a lognormal results in an unrectified trajectory drift.

Regarding the number of lognormals detected by ScriptStudio and iDeLog, Table 3 reports the results. Here

TABLE 4
RESULTS OF IDELOG WITH AND WITHOUT SMOOTHING

Database	Smoothing	SNR_t	σ_{SNR_t}	$SNR_t/NbLog$	SNR_v	σ_{SNR_t}	$SNR_v/NbLog$
BiosecureID	Yes	23.28dB	5.67	0.880	15.22dB	1.19	0.569
	No	22.63dB	5.16	0.811	16.20dB	1.91	0.579
MCYT100	Yes	21.33dB	5.30	0.905	15.22dB	1.21	0.630
	No	20.81dB	4.65	0.748	15.31dB	1.87	0.537
SUSIG Visual	Yes	24.11dB	5.49	1.338	15.19dB	1.25	0.841
	No	21.63dB	6.07	0.709	10.25dB	3.68	0.358

again, the statistical analysis confirms that all the results were significantly different. As can be seen, the number of lognormals detected by ScriptStudio is more stable. This is due to the stop criteria in the stroke extraction procedure. On the one hand, ScriptStudio stops the stroke extraction when the current reconstruction is considered good enough or the current number of strokes is larger than a maximum value, or when there is no set of characteristic points left. On the other hand, iDeLog extracts as many strokes as there are minima found in the velocity profile. Obviously, if there is no smoothing, the number of strokes is larger, as can be seen in Table 3. Thus, the smoothing reduces the number of lognormals in this case.

4.4 Experiment 2: Smoothing vs. Observed Signal

For a deeper examination of the effect of smoothing in iDeLog, we analyzed the SNR results in these two conditions. ScriptStudio was not part of this analysis since the SNR_v of decomposing an observed signal without smoothing is not available as a feature in the licensed version of ScriptStudio. The results obtained with and without smoothing are summarized in Table 4. Note that in the case of no smoothing, the SNRs compare the reconstructed signal with the observed one. Here, the ANOVA analysis detected only significant differences in the case of SNR_v for the SUSIGVisual database.

As can be seen in Table 4, there is no clear tendency in favor of using or not using smoothing, except in the case of SNR_v , because of the noisy velocity profiles present, possibly caused by a jitter in the sampling frequency.

4.5 Experiment 3: Automatic Signature Verification with the Reconstructed Databases

The purpose of this experiment was not to propose iDeLog as an Automatic Signature Verifier (ASV). Instead, we were keen on identifying the kind of reconstructed signatures that has more similarities to real signatures performance-wise, in a signature verification test scenario. On-line signature verification requires a precise reconstruction not only in the velocity domain, but also in the trajectory domain. Here, it is expected that iDeLog offers many advantages.

Automatic Signature Verification

An automatic signature verifier compares a questioned signature with a set of undoubted signatures from a signer.

TABLE 5
AREA BETWEEN THE DET CURVES OF THE ORIGINAL AND RE-CONSTRUCTED SIGNATURES DATABASES.

Verifier	Database	Reconstructed with		
		ScriptStudio	iDeLog with smoothing	iDeLog without smoothing
DTW	BiosecureID	0.0138	0.0120	0.0111
	MCYT100	0.1230	0.0086	0.0083
	SUSIGVisual	0.0185	0.0077	0.0085
MD	BiosecureID	0.0333	0.0155	0.0158
	MCYT100	0.0338	0.0095	0.0105
	SUSIGVisual	0.0282	0.0236	0.0359

The result of such a comparison is usually quantified as a score. If the score is greater than a given threshold, the questioned signature is accepted as genuine, otherwise it is rejected.

The performance of an automatic signature verifier is measured in terms of False Acceptance Rate (FAR) and False Rejection Rate (FRR). The Equal Error Rate (EER) is the verification error when FAR is equal to FRR. The Detection Error Tradeoff (DET) curve plots the FAR against the FRR.

Two different on-line Automatic Signature Verifiers (ASVs) were chosen to cover a wide range of signature properties in the test:

1. A Dynamic Time Warping (DTW)-based verifier that compares the dynamics of the questioned and reference signature using a DTW algorithm with Euclidean Distance [42].
2. A Manhattan distance (MD)-based verifier that compares histograms of absolute and relative frequencies of the dynamics. The distance between histograms of reference and questioned signatures is calculated by means of the Manhattan distance [43].

These experiments were performed using the trajectory, velocity, acceleration and pressure as input parameters. In the case of the original databases, the velocity and acceleration were computed as the first and second derivatives of the trajectory; for the reconstructed databases, the trajectory was obtained as the integral of the velocity and the acceleration as the first derivative of the velocity. The pressures in the reconstructed database were 1 for pen-downs and 0 for pen-ups.

All the verifiers were trained with the 5 first signatures of each signer, while the remaining genuine signatures were used for testing the false rejection rate. In all cases, the false acceptance rate was obtained with the genuine test samples from all the remaining users, considered as potential random forgeries.

EER and DET comparison

This test consists in decomposing the entire signature database as a sum of lognormals and reconstructing them from the Sigma-Lognormal parameters. The Automatic Signature Verifiers were run with the original and reconstructed databases. The more the results were similar, the more accurate the reconstruction was. Specifically, we

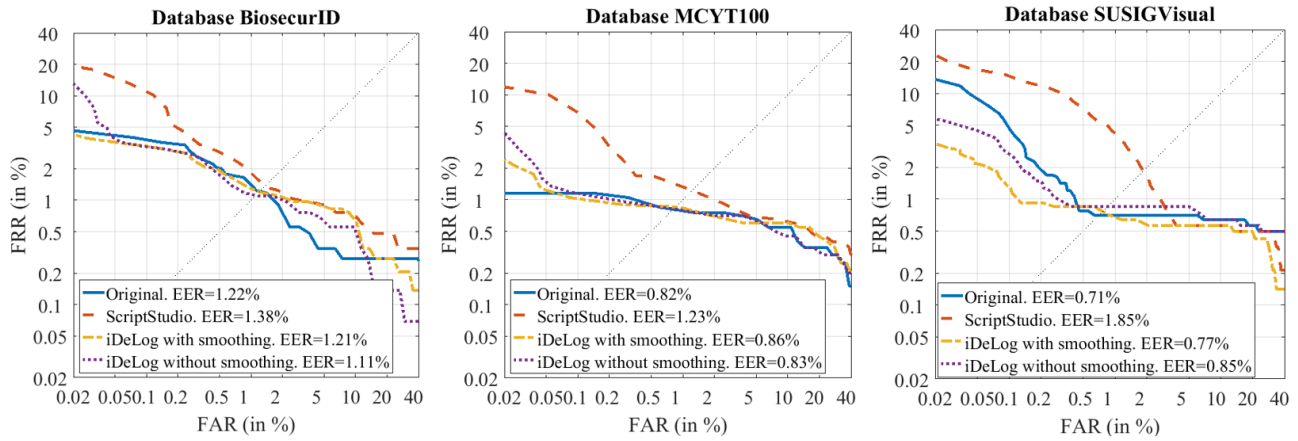


Fig. 9. DET curves of the DTW automatic signature verifier with the original databases and the reconstructed databases with ScripStudio, iDeLog with smoothing, and with iDeLog without smoothing. The databases are BiosecureID, MCYT100 and SUSIGVisual.

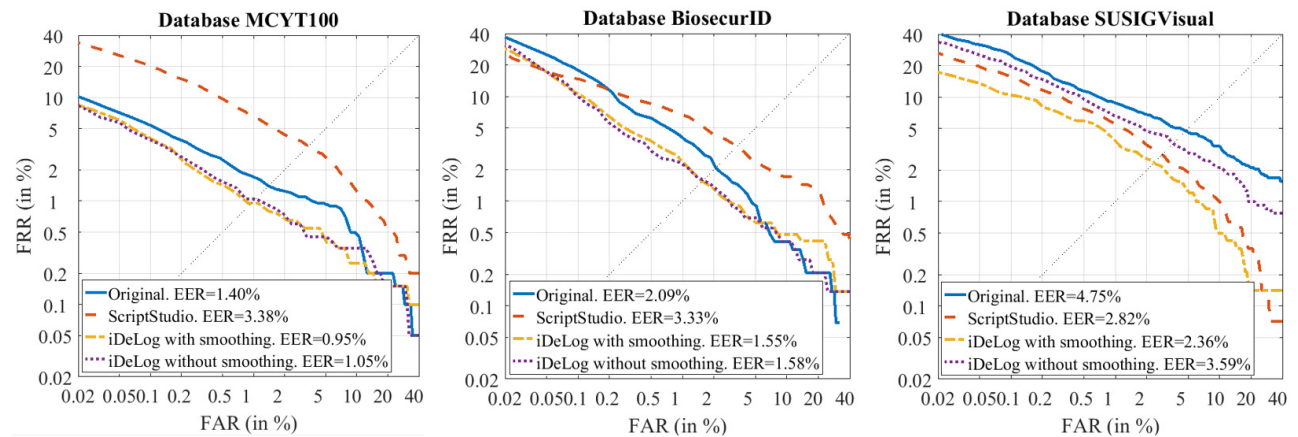


Fig. 10. DET curves of the Manhattan distance automatic signature verifier with the original and the reconstructed databases with ScripStudio, iDeLog with smoothing, and with iDeLog without smoothing. The databases are BiosecureID, MCYT100 and SUSIGVisual.

compared the original database with the databases reconstructed by ScriptStudio and iDeLog with smoothing, and by iDeLog without smoothing.

The DET curves, along with the EERs, obtained with the different databases are shown in Figures 9 and 10 for DTW and MD, respectively. To compare the DET curves, Table 5 shows the area between the DET curves of the original database and those of the reconstructed databases. The lower this area, the better the reconstruction.

As can be seen, the iDeLog reconstructed database displays similar DET curves and EERs as the original one in the three databases using the DTW verifier. However, in accordance with our results above, there is no clear indication for using or not using smoothing.

In the case of the MD verifier, we had different findings. The EERs of the iDeLog reconstructed databases were similar, but lower than the EERs obtained with the original database. Furthermore, the EERs obtained with smoothing outperformed the EERs without smoothing. This could be due to the fact the MD verifier uses features based on histograms of dynamic characteristics, such as velocity and acceleration. As a result, it seems that smoothed velocity profiles result in more discriminative histograms.

A second test was conducted in both verifiers using only

the velocity values as input parameters. This is because ScriptStudio focuses only on velocity. The EERs were higher in nearly all the cases since the verifiers were fed with less information in these cases. The same ranking tendency with regard to the DET curves was observed in the results.

5. CONCLUSIONS

This paper presented iDeLog, a novel framework for calculating the parameters of the Sigma-Lognormal model. The framework is designed to handle continuous, long and complex movements, where the trajectory and velocity reconstructions are required simultaneously.

iDeLog depends on motor equivalence theory and the hypothesis of a visual feedback compatible with open-loop motor control. Thus, it extracts the spatial and kinematic parameters of the Sigma-Lognormal model separately, and then performs a joint parameter optimization, moving the virtual target points.

The experiments conducted show that statistically, iDeLog is able to reconstruct the trajectory significantly better than ScriptStudio when movement is continuous, long and complex, as is the case of handwritten signatures.

However, iDeLog provides poorer results than ScripStudio in reconstructing velocity, which is probably due to the trade-off between trajectory and velocity in iDeLog.

iDeLog also provides the possibility of avoiding the initial preprocessing, which thus allows it to retain the statistical performance, except in the case of the velocity, where the input is noisy. This could be an advantage for some verifiers, as seen in the MD results.

The new procedure is expected to represent a further step forward for applications based on the Sigma-Lognormal model and on the Kinematic Theory, in which both the velocity and trajectory reconstructions are important. Moreover, iDeLog can be especially useful in those cases where there is some information in high frequency components, and therefore, where it would be unnecessary to complete a smoothing preprocessing stage.

REFERENCES

- [1] R. Plamondon, C. O'Reilly, J. Galbally, A. Almaksour, and É. Anquetil, "Recent Developments in the Study of Rapid Human Movements with the Kinematic Theory: Applications to Handwriting and signature synthesis," *Pattern Recognit. Lett.*, vol. 35, no. 1, pp. 225–235, 2014.
- [2] R. Plamondon, "A kinematic Theory of Rapid Human Movements. Part I: Movement Representation and Generation," *Biol. Cybern.*, vol. 72, no. 4, pp. 309–320, 1995.
- [3] R. Plamondon, "A Kinematic Theory of Rapid Human Movements. Part II: Movement Time and Control," *Biol. Cybern.*, vol. 72, no. 2, pp. 309–320, 1995.
- [4] R. Plamondon, "A Kinematic Theory of Rapid Human Movements. Part III: Kinematic Outcomes," *Biol. Cybern.*, vol. 78, no. 2, pp. 133–145, 1998.
- [5] R. Plamondon, "A Kinematic Theory of Rapid Human Movements. Part IV: A Formal Mathematical Proof and New Insights," *Biol. Cybern.*, vol. 89, no. 2, pp. 126–138, 2003.
- [6] C. O'Reilly and R. Plamondon, "Development of a Sigma-Lognormal representation for on-line signatures," *Pattern Recognit.*, vol. 42, no. 12, pp. 3324–3337, 2009.
- [7] A. Fischer and R. Plamondon, "Signature Verification Based on the Kinematic Theory of Rapid Human Movements," *IEEE Trans. Human-Machine Syst.*, vol. 47, no. 2, pp. 169–180, Apr. 2017.
- [8] M. Diaz, A. Fischer, M. A. Ferrer, and R. Plamondon, "Dynamic Signature Verification System Based on One Real Signature," *IEEE Trans. Cybern.*, vol. 48, no. 1, pp. 228–239, Jan. 2018.
- [9] M. Gomez-Barrero, J. Galbally, J. Fierrez, J. Ortega-Garcia, and R. Plamondon, "Enhanced on-line Signature Verification Based on Skilled Forgery Detection using Sigma-Lognormal Features," in *2015 International Conference on Biometrics (ICB)*, 2015, pp. 501–506.
- [10] J. Galbally, R. Plamondon, J. Fierrez, and J. Ortega-Garcia, "Synthetic On-line Signature Generation. Part I: Methodology and Algorithms," *Pattern Recognit.*, vol. 45, no. 7, pp. 2610–2621, Jul. 2012.
- [11] J. Galbally, J. Fierrez, J. Ortega-Garcia, and R. Plamondon, "Synthetic On-line Signature Generation. Part II: Experimental Validation," *Pattern Recognit.*, vol. 45, no. 7, pp. 2622–2632, Jul. 2012.
- [12] M. A. Ferrer, M. Diaz-Cabrera, and A. Morales, "Static Signature Synthesis: A Neuromotor Inspired Approach for Biometrics," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 667–680, Mar. 2015.
- [13] M. A. Ferrer, M. Diaz, C. Carmona-Duarte, A. Morales, C. Carmona, and A. Morales, "A Behavioral Handwriting Model for Static and Dynamic Signature Synthesis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1041–1053, Jun. 2017.
- [14] M. Ferrer et al., "Static and Dynamic Synthesis of Bengali and Devanagari Signatures," *IEEE Trans. Cybern.*, pp. 1–14, 2018.
- [15] C. Carmona-Duarte, M. A. Ferrer, A. Parziale, and A. Marcelli, "Temporal Evolution in Synthetic Handwriting," *Pattern Recognit.*, vol. 68, pp. 233–244, Aug. 2017.
- [16] S. Djeziri, W. Guerfali, R. Plamondon, J.M. Robert, "Learning Handwriting with Pen-based Systems: Computational Issues," *Pattern Recognit.*, vol. 35, no. 5, pp. 1049–1057, 2002.
- [17] D. Impedovo et al., "Writing Generation Model for Health Care Neuromuscular System Investigation," in *Tenth International Meeting on Computational Intelligence Methods for Bioinformatics and Biostatistics*, 2013, pp. 1–16.
- [18] C. O'Reilly, R. Plamondon, "Impact of the Principal Stroke Risk Factors on Human Movements," *Hum. Mov. Sci.*, vol. 30, no. 4, pp. 792–806, 2011.
- [19] A. Van Gemmert, R. Plamondon, and C. O'Reilly, "Using the Sigma-lognormal Model to Investigate Handwriting of Individuals with Parkinson's Disease," in *16th Biennial Conf. on the International Graphonomics Society*, 2013, pp. 119–122.
- [20] K. Lebel, H. Nguyen, C. Duval, R. Plamondon, and P. Boissy, "Capturing the Cranio-Caudal Signature of a Turn with Inertial Measurement Systems: Methods, Parameters Robustness and Reliability," *Front. Bioeng. Biotechnol.*, vol. 5, Aug. 2017.
- [21] M. Režnáková, L. Tencer, R. Plamondon, and M. Cheriet, "Forgetting of Unused Classes in Missing Data Environment using Automatically Generated Data: Application to On-line Handwritten Gesture Command Recognition," *Pattern Recognit.*, vol. 72, pp. 355–367, Dec. 2017.
- [22] C. Ramaiah, R. Plamondon, and V. Govindaraju, "A Sigma-lognormal Model for Character Level CAPTCHA Generation," in *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, 2015, pp. 966–970.
- [23] D. Berio, Frederic Fol Leymarie, Rejean Plamondon, "Computer Aided Design of Handwriting Trajectories with the Kinematic Theory of Rapid Human Movements," in *18th Biennial Conference of the International Graphonomics Society (IGS)*, 2017.
- [24] D. Martín-Albo, L. A. Leiva, J. Huang, and R. Plamondon, "Strokes of Insight: User Intent Detection and Kinematic Compression of Mouse Cursor Trails," *Inf. Process. Manag.*, vol. 52, no. 6, pp. 989–1003, Nov. 2016.
- [25] L. A. Leiva, D. Martín-Albo, and R. Plamondon, "The Kinematic Theory Produces Human-Like Stroke Gestures," *Interact. Comput.*, vol. 29, pp. 552–565, Jan. 2017.
- [26] C. Carmona-Duarte et al., "Phonatory and Articulatory Correlates in Kinematic Neuromotor Degeneration," 2017, pp. 203–208.
- [27] C. O'Reilly, R. Plamondon, "A Software Assistant for the Design and Analysis of Neuromuscular Tests," in *IEEE Biomedical Circuits and Systems Conferenc*, 2007, pp. 107–110.
- [28] D. Martín-Albo, Réjean Plamondon, Enrique Vidal, "Improving Sigma-lognormal Parameter Extraction," in *13th International Conference on Document Analysis and Recognition*, 2015.
- [29] K. S. Lashley, "Basic Neural Mechanisms in Behavior," *Psychol. Rev.*, vol. 37, no. 1, pp. 1–24, 1930.
- [30] N. A. Bernstein, "The Co-ordination and Regulation of Movements: Conclusions towards the Study of Motor Co-ordination," *J. Neurol. Exp. Neurol.*, vol. 27, no. 2, pp. 104–113, 1968.
- [31] A. M. Wing, "Motor Control: Mechanisms of Motor Equivalence in Handwriting," *Curr. Biol.*, vol. 10, no. 6, pp. R245–R248, Mar. 2000.
- [32] C. Marquardt, W. Gentz, and N. Mai, "Visual Control of Automated Handwriting Movements," *Exp. Brain Res.*, vol. 128, no. 1–2, pp. 224–228, Sep. 1999.
- [33] M. Pertsinakis and H.-L. Teulings, "Effect of Visual Feedback on

the Static and Kinematic of Signatures," in *Proceedings of International Graphonomics Society*, 2017, pp. 83–87.

- [34] M. Djioua and R. Plamondon, "A new Algorithm and System for the Characterization of Handwriting Strokes with Delta-Lognormal Parameters," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 11, pp. 2060–2072, 2009.
- [35] E. R. Kandel, J. H. Schwartz, and T. M. Jessell, *Principles of Neural Science*, vol. 4, 2013.
- [36] D. W. Marquardt, "An Algorithm for Least-Squares Estimation of Nonlinear Parameters," *J. Soc. Ind. Appl. Math.*, vol. 11, no. 2, pp. 431–441, Jun. 1963.
- [37] J. Galbally, M. Diaz-Cabrera, M. A. Ferrer, M. Gomez-Barrero, A. Morales, and J. Fierrez, "On-line Signature Recognition through the Combination of Real Dynamic Data and Synthetically Generated Static Data," *Pattern Recognit.*, vol. 48, no. 9, pp. 2921–2934, Sep. 2015.
- [38] J. Ortega-Garcia et al., "MCYT Baseline Corpus: A Bimodal Biometric Database," *IEE Proc. - Vision, Image, Signal Process.*, vol. 150, no. 6, p. 395, 2003.
- [39] A. Kholmatov and B. Yanikoglu, "SUSIG: An On-line Signature Database, Associated Protocols and Benchmark Results," *Pattern Anal. Appl.*, vol. 12, no. 3, pp. 227–236, Sep. 2009.
- [40] T. Duval, C. Rémi, R. Plamondon, J. Vaillant, and C. O'Reilly, "Combining Sigma-lognormal Modeling and Classical Features for Analyzing Graphomotor Performances in Kindergarten Children," *Hum. Mov. Sci.*, vol. 43, pp. 183–200, Oct. 2015.
- [41] R. Plamondon, C. O'Reilly, C. Rémi, and T. Duval, "The Lognormal Handwriter: Learning, Performing, and Declining," *Front. Psychol.*, vol. 4, 2013.
- [42] A. Fischer, M. Diaz, R. Plamondon, and M. A. Ferrer, "Robust Score Normalization for DTW-Based On-line Signature Verification," in *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, 2015, vol. 2015–Novem, pp. 241–245.
- [43] Napa Sae-Bae, N. Memon, N. Sae-Bae, and N. Memon, "Online Signature Verification on Mobile Devices," *IEEE Trans. Inf. Forensics Secur.*, vol. 9, no. 6, pp. 933–947, Jun. 2014.



Miguel A. Ferrer received an M.Sc. and a Ph.D. from the Universidad Politécnica de Madrid, Madrid, Spain, in 1988 and 1994, respectively. He joined the University of Las Palmas de Gran Canaria, Las Palmas, Spain, in 1989, where he is currently a Full Professor. He established the Digital Signal Processing Research Group in 1990. His current research interests include pattern recognition, biometrics, audio quality, and computer vision applications to fisheries and aquaculture.



Moises Diaz received an M.Tech., an M.Sc., and a Ph.D. in Engineering, and an M.Ed. degree in secondary education from La Universidad de Las Palmas de Gran Canaria, Las Palmas, Spain, in 2010, 2011, 2016, and 2013, respectively. He is currently an associate professor at Universidad del Atlántico Medio, Spain. His current research interests include pattern recognition, document analysis, handwriting recognition, biometrics, computer vision, and intelligent transportation systems.



Cristina Carmona-Duarte received a Telecommunication Engineering degree in 2002 and a Ph.D. in 2012 from La Universidad de Las Palmas de Gran Canaria. She has been assistant Professor at Universidad de Las Palmas, where she is currently a researcher. Her research areas include high resolution radar, pattern recognition and biometrics.

Réjean Plamondon received a B.Sc. in Physics, and an M.Sc.A. and a Ph.D. in Electrical Engineering from Université Laval, Quebec, P.Q., Canada in 1973, 1975 and 1978, respectively. In 1978, he joined the faculty of the École Polytechnique, Université de Montréal, Montreal, P.Q., Canada, where he is currently a Full Professor. He was the Head of the Department of Electrical and Computer Engineering from 1996 to 1998 and the President of École Polytechnique from 1998 to 2002. He is now the Head of the Laboratoire

Scribens at this institution. In 2013, he received the IAPR/ICDAR outstanding achievement award "for theoretical contributions to the understanding of human movement and its applications to signature verification, handwriting recognition, instruction, and health assessment, and for promoting on-line document processing in numerous multidisciplinary fields."