# Kinematic Reconstruction of Calligraphic Traces from Shape Features

1st Daniel Berio
*Department of Computing*
*Goldsmiths, University of London*
London, United Kingdom
d.berio@gold.ac.uk

2nd Frédéric Fol Leymarie
*Department of Computing*
*Goldsmiths, University of London*
London, United Kingdom
ffl@gold.ac.uk

3rd Réjean Plamondon
*Department of Electrical Engineering*
*Polytechnique Montréal*
Montréal, Québec, Canada
rejean.plamondon@polymtl.ca

*Abstract*—**Our goal is to be able to reproduce computationally calligraphic traces, e.g. as found in the art practices of graffiti and various forms of more traditional calligraphy, while mimicking the production process of such art forms. We design our user interfaces in a procedural generation and computer aided design (CAD) setting. As a result, we seek to seamlessly work between data used in design packages (without kinematics) and data easily digitised by users (e.g. online, with kinematics). To achieve these goals, we propose a method that allows to reconstruct kinematics from solely the geometric trace of handwritten trace in the form of parameters of the Sigma-Lognormal model. We purposely ignore the kinematics possibly embedded in the data in order to treat online data and vector patterns with the same procedure.**

*Index Terms*—**graphonomics, graffiti, calligraphy, curvature, symmetry, kinematic theory, sigma-lognormal**

## I. INTRODUCTION

Many handwriting analysis methods rely on a prior segmentation of the handwriting trace into constituent primitives or strokes. Some methods exploit the kinematics of the movement and segment the trajectory in correspondence with minima of velocity. Accordingly with the stereotypical inverse relation between speed and curvature of handwriting traces, other methods rely on the identification of curvature extrema along the pen-trace. This fits with a modelization of handwriting in the form of ballistic stroke primitives, in which curvature extrema will typically correspond with velocity minima and are indicative of the initiation of a new stroke.

A robust identification of curvature extrema can be difficult, since curvature is a second order differential quantity which tends to amplify the effects of noise in the input as an outcome of the digitisation process. One popular method to overcome this problem is to first smooth the digitised signal using a filter (e.g convolving with a Gaussian) or interpolating with some analytic function (e.g. smoothing splines). However, smoothing risks to remove (perceptually) important features of an outline and choosing reasonable parameters remains a difficult task. To overcome this fundamental issue, one possible avenue is to generate an intermediate *scale-space* in which features are identified and tracked at different scales [1]. Such a scale-space is very often produced by iterative

Gaussian filtering in the spatial domain, or via the frequency domain using wavelets [2]. An alternative to such traditional filtering (which tends to blur away details) is to use a structural notion of scale, e.g. by associating a support metric along a contour with each curvature feature being tracked, such as when performing morphological operations on the curvature function [3]. Nevertheless, such methods operating directly on the curvature function (along a contour) suffer from poor localisation, and do not capture well singularities which can be perceptually significant, such as curvature breaks (typical of corner features). Also, such methods are usually developed and implemented for closed contours, which is not typical of handwritten traces, and further do not behave well with loops or other singular behaviours which are common in calligraphy.

An alternative to working directly with the curvature function is to exploit the correspondence existing between symmetry axes and the curvature behavior of a contour [4]. Originally pioneered by Harry Blum in the 1960's for the study of biological shape [5], the Symmetry Axis Transform (SAT) – also known as Medial Axis or simply skeleton for closed contours – is a shape representation that provides a bridge between geometry and topology. The SAT is commonly viewed as the set of centers of "maximally inscribed" disks, or with the "prairie grassfire" or wave-front analogy, in which the symmetry axes are given by the "quench" points at which fire fronts or waves propagating from the object boundary meet and stop expanding. Contrary to a common misinterpretation, the SAT is not only defined for closed shapes, but is valid also for open contours or even point samplings [6] — in the latter case becoming similar to the Voronoi graph very popular in computational geometry and computer aided design (CAD).

In this paper, as a starting point, we make use of the duality between curvature and symmetry axes [4] in order to extract more robustly curvature features, such as extrema along a handwriting or drawing trace. The method is also directly adaptable to open contours, to contours with breaks in curvature, and can further be used to identify loops. Each feature is also explicitly paired with corresponding contact circles and a support region — where curvature is approximately monotonic. Given such a robust and rich feature description of a handwritten trace, we show how to exploit this spatial and structural geometric representation to *infer the kinematics*

of a likely generative movement. To do so, we rely on the *Kinematic Theory of Rapid Human Movements* [7], [8], a family of models of reaching and handwriting motions, in which a movement is described as the result of the parallel and hierarchical interaction of a large number of coupled neuromuscular components.

The resulting method allows the reconstruction of physiologically plausible velocity profiles for the geometric trace of an input movement given as an ordered sequence of points. While state of the art methods exist for the parameter reconstruction of Kinematic Theory based models from digitised traces of handwriting [8]–[10], we design our method with the goal of targeting applications in both graphonomics and CAD/computer graphics. As a result, we purposely ignore the kinematics of the input in order to seamlessly handle online handwriting with arbitrary sampling quality as well as vector art in which only the sequential ordering of points may be available. We also choose this approach with the future aim of combining our method with one that recovers temporal information from bitmap images such as the one presented by Plamondon and Privitera [11].

In the following sections we first summarise the Sigma Lognormal ($\Sigma\Lambda$) model (Section II), which we use to describe the motions of a pen-tip. We then describe the trajectory segmentation method (Section III), which is used as a preprocess for the reconstruction of $\Sigma\Lambda$ parameters from geometry only (Section IV).

## II. The Sigma Lognormal ($\Sigma\Lambda$) Model

On the basis of the Kinematic Theory [7], we describe the kinematics of a handwriting/drawing movement via the the Sigma Lognormal ($\Sigma\Lambda$) model [8]. The $\Sigma\Lambda$ model describes complex handwriting trajectories via the vectorial superimposition of $N$ time shifted stroke primitives. The speed of each stroke is given by a lognormal

$$\Lambda_i(t) = \frac{1}{\sigma_i\sqrt{2\pi}(t - t_{0i})} \exp\left(-\frac{(ln(t - t_{0i}) - \mu_i)^2}{2\sigma_i{}^2}\right) \quad (1)$$

which describes impulse response of each stroke to a centrally generated command occurring at time $t_{0i}$. The parameters $\mu_i$ and $\sigma_i$ respectively describe the stroke delay and response time in a logarithmic time scale, and determine the shape and asymmetry of the lognormal. With the assumption that handwriting movements are made with rotations of the elbow or wrist, the curvilinear evolution of a stroke can be described by a circular arc. The angular evolution of a stroke is described by using the time integral of eq. 1:

$$\phi_i(t) = \theta_i - \frac{\delta_i}{2} + \delta_i \int_0^t \Lambda_i(u)du \quad (2)$$

$$= \theta_i - \frac{\delta_i}{2} + \frac{\delta_i}{2}\left[1 + \mathrm{erf}\left(\frac{\log(t - t_{0i}) - \mu_i}{\sigma_i\sqrt{2}}\right)\right], \quad (3)$$

where $\theta_i$ is the direction of the stroke and $\delta_i$ is the *stroke curvature* parameter which determines the internal angle of

the circular arc. The planar pen-tip velocity is then calculated with:

$$\dot{x} = \sum_{i=1}^{N} \hat{D}_i\Lambda(t)_i cos(\phi_i(t)) \quad \text{and} \quad \dot{y} = \sum_{i=1}^{N} \hat{D}_i\Lambda(t)_i sin(\phi_i(t)) \quad (4)$$

where $\hat{D}_i = D_i h(\theta_i)$ is the length $D_i$ of the stroke scaled by

$$h(\theta_i) = \begin{cases} \frac{2\theta_i}{2\sin\theta_i} & \text{if } |\sin\theta_i| > 0, \\ 1 & \text{otherwise}, \end{cases} \quad (5)$$

which compensates for the stroke curvature based on the ratio between the perimeter and the chord length of a circular arc. The acceleration components of the lognormal trajectory are then given by [12]:

$$\ddot{x} = \sum_{i=1}^{N} \hat{D}_i\dot{\Lambda}_i(t)\cos(\phi_i(t)) - \hat{D}_i\delta_i\Lambda_i^2(t)\sin(\phi_i(t)) \quad (6)$$

$$\ddot{y} = \sum_{i=1}^{N} \hat{D}_i\dot{\Lambda}_i(t)\sin(\phi_i(t)) + \hat{D}_i\delta_i\Lambda_i^2(t)\cos(\phi_i(t)). \quad (7)$$

with

$$\dot{\Lambda}_i(t) = \Lambda_i(t)\frac{\mu_i - \sigma_i^2 - \log(t - t_{0i})}{\sigma^2(t - t_{0i})} \quad (8)$$

which allows us to compute the curvature at time $t$ with the well known formula:

$$\kappa(t) = (\dot{x}\ddot{y} - \dot{y}\ddot{x})/(\dot{x}^2 + \dot{y}^2)^{3/2}. \quad (9)$$

The sequence of curvilinear strokes describes an *action plan* consisting of an initial position $\boldsymbol{p}_0$ followed by a sequence of $N$ *virtual targets* $\boldsymbol{p}_1, \ldots, \boldsymbol{p}_N$ each corresponding to an imaginary aiming locus per stroke (Fig. 5). The degree of time overlap between lognormal components then defines the degree of smoothness of the trajectory in correspondence with each virtual target, where a greater time overlap results in a smoother trajectory. In order to facilitate interactive applications [13] and to simplify the subsequently described parameter reconstruction method, we compute the directions $\theta_i$ and length $D_i$ for each pair of consecutive virtual targets $(\boldsymbol{p}_{i-1}, \boldsymbol{p}_i)$. Furthermore, we explicitly define the time overlap of each lognormal through an intermediate parameter $\Delta t_i \in [0, 1]$ where $t_{0i} = t_{0i-1} + \Delta t_i \sinh(3\sigma_i)$ if $i > 1$ and $t_{01} = 0$.

## III. Trajectory segmentation and analysis

The proposed trajectory reconstruction method exploits a prior feature analysis of the input which is based on the SAT together with the *Symmetry Curvature Duality* (SCD) theorem [4], a result presented by Leyton which links the symmetry axes of a shape to its curvature extrema and states [4]:

> *Any segment of a smooth planar curve, bounded by two consecutive curvature extrema of the same type, has a unique symmetry axis, and the axis terminates at the curvature extremum of the opposite type.*

Following the SCD theorem, given a symmetry axis, it is then possible to identify a curvature extrema at one end (tip).

In a traditional setting, the SAT is computed at once for a given final contour or written trace (in our case). This however does not allow to identify all perceptually significant curvature extrema, as part of a contour may forbid the existence of a symmetry axis that would otherwise end near a curvature extremum.[1] Berio and Fol Leymarie have recently developed an alternative way to compute symmetry axes which avoids this masking effect[2] In summary, a symmetry axis is evaluated as one travels along a written trace. Once a significant axis is found, its existence ends once another significant axis emerge, and the previous written trace already traversed is "forgotten" (so as to not mask other potential extrema of curvature associated to later parts of the trace).

### A. Feature analysis

The input to our method is a curve $z(s)$ parameterised by arc length $s$ and with total arc length $l$. For each curvature extrema, the feature segmentation outputs a sequence of disks $\{\Omega_i\}$ centered at $c(\Omega_i)$, with radius $r(\Omega_i)$ and corresponding with the circle of curvature osculating the extrema. Each disk has contact with the input curve along a region $P(\Omega_i)$ at which curvature is approximately constant, which we refer to as the disk's *projection* and is delimited between two *anchor points* defined by their respective arc lengths $s_0(\Omega_i)$ and $s_1(\Omega_i)$. The location of the curvature curvature extrema $s(\Omega_i)$ is then given by the midpoint of the projection $(s_0(\Omega_i) + s_1(\Omega_i))/2$. The curvature of the extrema is then simply $\kappa(\Omega_i) = \pm 1/r(\Omega_i)$, where the sign is computed using the signed area of the triangle $[z(s_0(\Omega_i)), c(\Omega_i), z(s_1(\Omega_i))]$.

### B. Trajectory segmentation

The symmetry axis extraction process results in the identification of a set of curvature extrema, where each extrema is paired with the corresponding osculating circle of curvature. The projection at which the disk has contact with the curve describes a trajectory segment where curvature is approximately constant. Given the estimate of curvature extrema we proceed with the evaluation of a curvilinear profile for the trajectory by fitting Euler spirals to contour segments bounded by consecutive curvature extrema (Section III-B1). Each Euler spiral segment is then decomposed into one or two circular arc segments, depending on the presence or not of an inflection (Section III-B2). This results in a sequence of circular arc segments that are then used in the subsequent estimation of $\Sigma\Lambda$ curvature parameters $\{\delta_i\}$ (Section IV).

*1) Euler spiral fitting:* Euler spirals (also known as Cornu spirals, or clothoids) [15] are an interesting type of curve in which curvature varies linearly with arc length, permitting the description of variably curved segments which may contain

---

[1]One way to prove this masking effect, is by considering the behavior of cusps of evolutes in relation to a symmetry axis. Belyaev and Yoshizawa [14] prove that an evolute cusp correspond with a symmetry axis branch only when the segment going from the cusp to the associated curvature extrema, does not intersect the remaining skeleton of the shape.

[2]This work is being detailed and submitted as a separate publication as it is more generic in application than solely to written traces, and as it requires more space to present it in sufficient detail.
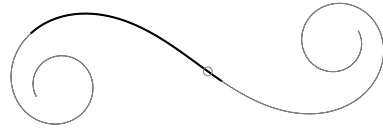


Fig. 1. An Euler spiral, its inflection point (circle) and a Euler spiral segment (black).

one inflection (Fig. 1). An Euler spiral is commonly parameterised by arc length $s$ using the cosine ($C(s)$) and sine ($S(s)$) Fresnel integrals:

$$C(s) + iS(s) = \int_0^s e^{i\pi t^2/2} du. \tag{10}$$

The curve is then defined between an initial ($s_{s0}$) and final ($s_{s1}$) parameter and can be conveniently computed in an efficient manner using an approximation method developed by Heald in [16].

In order to fit an Euler spiral, we first compute approximate tangent directions along the trajectory in correspondence with the initial and final points of the segment under examination. This allows to compute a first estimate of the spiral's initial and final parameters rapidly using a secant method described in [15]. On the other hand, the tangent estimate is likely not to be precise due to noise in the input so we then proceed to refine the fit with a least squares optimisation using the Gauss-Newton method. In order to do so we parameterise a spiral segment scaled by a factor $\alpha$ and rotated by an amount $\omega$ with:

$$\boldsymbol{p}_s(s) = \begin{bmatrix} \alpha \cos(\omega) C_0(s) - \alpha \sin(\omega) S_0(s) \\ \alpha \sin(\omega) C_0(s) + \alpha \cos(\omega) S_0(s) \end{bmatrix}, \quad \text{where} \tag{11}$$

$$C_0(s) = C(s) - C(s_{s0}) \quad \text{and} \quad S_0(s) = S(s) - S(s_{s0}). \tag{12}$$

We then transform the input curve segment so that its end points match the end points of the spiral in its canonical form with $\alpha = 1$ and $\omega = 0$, and then proceed with the minimisation

$$\min_{s_{s0}, s_{s1}, \alpha, \omega} \frac{1}{2} \sum_{i=1}^n \| \boldsymbol{p}_s(s_s[i]) - \boldsymbol{z}(\hat{s}[i]) \|^2 \tag{13}$$

where $s_s[i]$ and $\hat{s}[i]$ respectively give $n$ equally spaced points (i) sampled along the spiral $\boldsymbol{p}_s(s)$ between $s_{s0}$ and $s_{s1}$ and (ii) along the input curve $\boldsymbol{z}(s)$.

*2) Inflections and Circular Arc Decomposition:* Inflections are directly found by checking the signs of the two spiral parameters $s_0$ and $s_1$. If the parameters have different sign the position along the spiral at which the arc length parameter $s = 0$ determines the location of the inflection. For each inflection, we check if the ratio $\min(|s_0|, |s_1|)/|s_1 - s_0|$ is less than a user defined threshold $\epsilon_{infl}$ (which we empirically set to $0.2$ in the accompanying examples), in which case we discard it as a degenerate case (being too close to one spiral end point).

Depending on the presence of an inflection we fit either one or two circular arcs to each Euler spiral segment. The internal angle of the circular arcs is easily estimated by integrating the curvature of the spiral and distinguishing between 3 cases.

*(a)* For the case of two arcs the internal angles are given by $s|s|$ for each parameter $s_0$ and $s_1$ (Fig. 2a). *(b)* In the case of a degenerate inflection, we use the same method to fit a single arc and choose only the parameter with the greatest absolute value and consequently higher curvature. *(c)* When no inflection is present the internal angle is given by $|(s_1|s_1| - s_0|s_0|)| \operatorname{sgn}(s_0)$. For each arc, we then check if the absolute internal angle is greater than $(3/4)\pi$, in which case we subdivide the arc in two (Fig. 2b).



Fig. 2. Decomposing Euler spirals (stippled grey) into arcs. **(a)** two arcs delimiting an inflection. **(b)** one segment with angle $> (3/4)\pi$ divided into two arcs.

These final steps produce an ordered sequence of $N$ circular arcs with internal angles $\hat{\theta}_{i\,i=1}^{N}$, center $\boldsymbol{c}(\hat{\theta}_i)$ and radius $r(\hat{\theta}_i)$. The circular arcs are delimited by $N+1$ feature points $\{\hat{s}_i\}_{i=0}^{N}$, each defining a distance along the contour and with $\{\hat{s}_0, \hat{s}_N\}$ indicating the initial and final points. Each feature point corresponding with a curvature extrema is also paired with its corresponding osculating circle with radius $r(\hat{s}_i)$ and curvature $\kappa(\hat{s}_i)$. Thus the projection $p(\hat{s}_i)$ for each extremum also defines a circular arc, at which the curvature $\kappa(\hat{s}_i)$ is constant (Fig. 3). Extrema at which the trajectory is smoother will result in a larger radius of curvature and a larger projection. As a result, it is then possible to produce an approximate reconstruction of the original trajectory in the form of circular arc segments, similarly to the method originally proposed in [17], but with the difference that each curvature extremum is also paired with a supporting arc, which results in a more accurate reconstruction of the original trace.
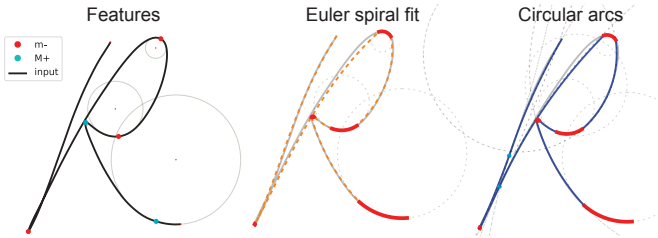


Fig. 3. Feature extraction (left) followed by Euler spiral fitting (middle) and arc decomposition (right) of a sample from the UJI handwritten character dataset. The arcs in red indicate regions with approximately constant curvature corresponding with curvature extrema.

## IV. ITERATIVE RECONSTRUCTION OF $\Sigma\Lambda$ PARAMETERS

Given the previous trajectory segmentation step and a number of simplifying assumptions, we have the information necessary to reconstruct the input trajectory given its geometry only, in terms of the shape features previously indentified (i.e. curvature extrema, inflections, interpolating arcs of spirals). The method is a development and improvement over our prior efforts [18], [19] in a similar direction. While a number
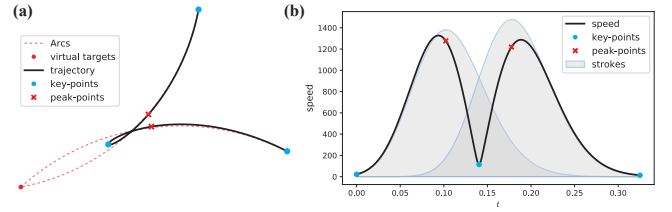


Fig. 4. Key-points (cyan circles) and peak-points (red crosses) overlaid on the trace (a) and speed profile (b) of a trajectory with two strokes.

of state of the art methods exist [8]–[10] for the accurate reconstruction of $\Sigma\Lambda$ parameters from digitised traces, these methods require the kinematics of the original trajectory. In our method, we sacrifice to some extent the accuracy of the reconstruction in order to seamlessly deal with online handwriting data as well as vector art in which only the time/sequential ordering of points may be available.

An initial estimation of the trajectory parameters is given by a virtual target sequence $\boldsymbol{p}_i = \boldsymbol{z}(\hat{s}_i)$, stroke curvature parameters $\delta_i = \theta(\hat{s}_i)$ and time overlap parameters $\Delta t_i = 0.5$. For the sake of simplicity, we consider the remaining parameters $\sigma_i$ and $\mu_i$ as typical properties of the neuromuscular system of a writer and keep these set to a user configurable value. The initial trajectory estimate is likely to differ from the original and to possess a reduced scale due to the smoothing effect of the lognormal stroke overlap (Fig. 5a).

To improve the reconstruction, we adopt an iterative refinement scheme (Fig. 5b) in which we adjust the curvature and time overlap parameters together with the virtual target positions in order to minimise the difference between the reconstructed and original trajectories. At each iteration, we rely on the estimation of a series of feature points along the generated lognormal trajectory. We compute $N + 1$ *key-points* $\{\tau_i\}_{i=0}^{N}$ along the trajectory (Fig. 4) where $\tau_1, \ldots, \tau_{N-1}$ indicates the time occurrence at which the influence of one stroke exceeds the previous one and curvature is maximal, while $\tau_0$ and $\tau_N$ are respectively the starting and ending time of the trajectory. In addition we compute $N$ *peak-points* $\{\gamma_i\}_{i=1}^{N}$, which indicate the approximate time occurrence of the maximum speed for each stroke (Fig. 4), which for each stroke is given by the mode of the corresponding lognormal $t_{0i} + \exp(\mu_i - \sigma_i^2)$.

The iterative refinement scheme is based on 3 observations: **Observation 1.** The time parameter $\Delta t_i$ is proportional to the curvature $\kappa(\tau_i)$ at the time of the corresponding key-point. Thus, a higher value of $\Delta t_i$ will decrease the amount of overlap of successive lognormals. This will result in a lower speed and higher curvature $\kappa(\tau_i)$ at the time occurrence of the key-point. Since we have a good approximation of the curvature $\kappa(\hat{s}_i)$ in the original trajectory, the relation between the two can be exploited in order to adjust $\Delta t_i$ proportionally at each iteration. We observe that changes in $\Delta t_i$ are not linearly related to changes in the curvature $\kappa(\tau_i)$ at the corresponding key-point. In order to compensate for this, we assume a $1/3$ power relation [20] which has been

often observed in human movement and particularly holds for elliptical portions of the trajectory [12], which is often the case near key-points. The reasoning is that given the relations

$$\Delta t \propto \kappa \quad \text{and} \quad \Delta t \propto 1/v$$

where $v$ denotes speed, we have the proportions relating desired and generated curvature/velocity:

$$\rho_\kappa = \hat{\kappa}/\kappa \quad \text{and} \quad \rho_v = \hat{v}/v.$$

As a result, given the power law $v = \kappa^{1/3}$ [20] and because velocity and $\Delta t$ are inversely proportional, we finally get the relation:

$$\rho_{\kappa v} = v/\hat{v} = (\kappa/\hat{\kappa})^{-1/3} = (\hat{\kappa}/\kappa)^{1/3}.$$

**Observation 2.** Shifting a virtual target $\boldsymbol{p}_i$ in a given direction, will cause the generated trajectory point $\boldsymbol{p}(\tau_i)$ to move in a similar direction. As a result, shifting the virtual target $\boldsymbol{p}_i$ along the vector $\boldsymbol{z}(\hat{s}_i) - \boldsymbol{p}(\tau_i)$ will decrease the distance between curvature extrema in the generated and original trajectories.

**Observation 3.** The distance $D_i$ between successive virtual targets $\boldsymbol{p}_i$ and $\boldsymbol{p}_{i-1}$ will influence the curvature of the stroke. In fact, augmenting this distance will increase the radius of curvature of the circular arc defined by the parameter $\delta_i$ and will result in a decrease of curvature for the stroke. While the trajectory tends to depart from the circular arc near the key-points at $t = \tau_i$ due to the smoothing effect of the lognormal time overlap, it tends to pass closer to the circular arc at $t = \gamma_i$ where the amplitude of the lognormal is maximal. As a result, we utilise this locus to evaluate the deviation from the desired arc $\hat{\theta}_i$ and correct the parameter $\delta_i$ accordingly.

As a result of these observations, we define each iteration of our method that consists of the following steps:

$$\Delta t_i \leftarrow \begin{cases} \Delta t_i(\kappa(\hat{s}_i)/\kappa(\tau_i))^{1/3} & \text{if} \quad r(\hat{s}_i) \neq \infty \\ \Delta t_{min} & \text{otherwise} \end{cases}, \quad (14)$$

$$\delta_i \leftarrow \delta_i + \lambda_\delta(\hat{\delta}_i - \delta_i) \quad \text{and} \quad (15)$$

$$\boldsymbol{p}_i \leftarrow \boldsymbol{p}_i + \lambda_p(\boldsymbol{c}(\hat{s}_i) - \boldsymbol{p}(\tau_i)), \quad (16)$$

where $\lambda_\delta$ and $\lambda_p$ are damping parameters that we experimentally tune to 0.1 and 0.5 to avoid excessive adjustments at each iteration, and

$$\hat{\delta}_i = 4\tan^{-1}\left[\frac{h}{a}\tan\left(\frac{\delta_i}{4}\right)\right] \quad \text{with} \quad (17)$$

$$a = \| \boldsymbol{p}_i - \boldsymbol{p}_{i-1} \| \quad \text{and} \quad (18)$$

$$h = \left(r(\hat{\theta}_i) - \| \boldsymbol{p}(\gamma_i) - \boldsymbol{c}(\hat{\theta}_i) \|\right)\text{sgn}(\hat{\theta}_i). \quad (19)$$

The term $h$ determines the amount to shift the curvature parameter $\delta_i$ by comparing the radius of the circular arc $\hat{\theta}_i$ initially fitted to the input to the distance between its center and the lognormal peak point $\boldsymbol{p}(\gamma_i)$. Note that for the case of inflection points ($r(\hat{s}_i) = \infty$), where the curvature is 0, we force $\Delta t_i$ to a user-defined minimum value. This results in a maximal overlap between lognormal components and gives a smooth transition between strokes in the generated trajectory.
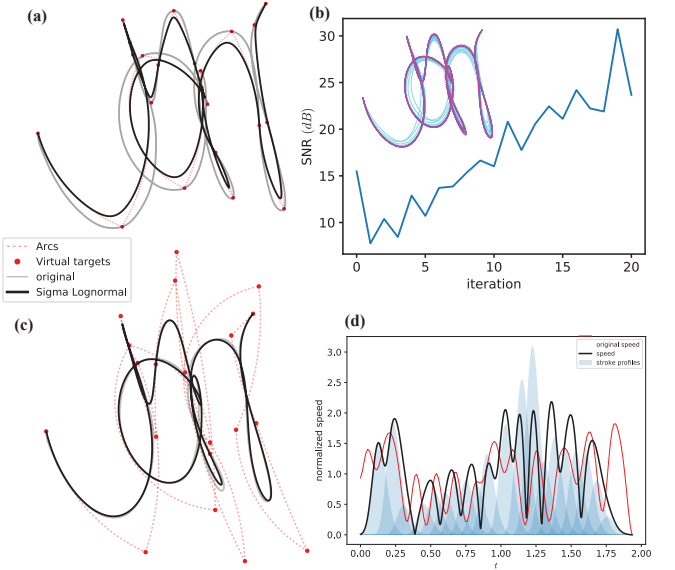


Fig. 5. $\Sigma\Lambda$ parameter reconstruction from shape features. **(a)** First guess of the parameters from features. **(b)** Iterative refinement steps and the SNR for each iteration, computed from 300 position samples along the generated and original trajectories. **(c)** The reconstructed trajectory. **(d)** Kinematics and strokes profiles of the generated trajectory, normalised and overlaid over the (smoothed) kinematics of the input.
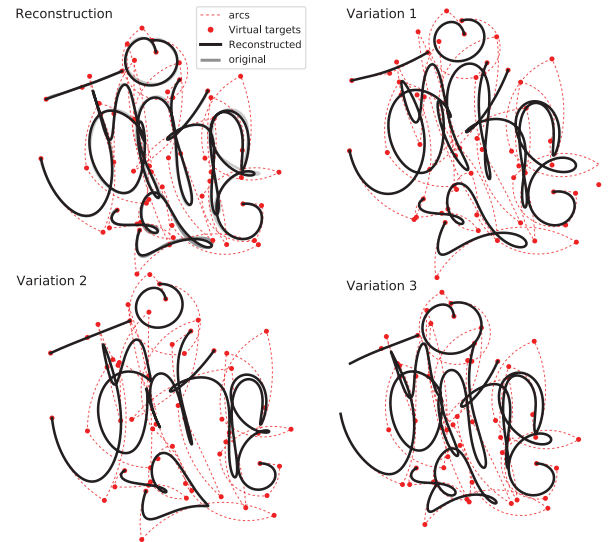


Fig. 6. Reconstruction and parametric variations of a graffiti instance from the Graffiti Analysis database.

## V. DISCUSSION AND APPLICATIONS

We tested the iterative refinement on different inputs ranging from online data (e.g. Graffiti Analysis database [21] and UJI handwritten character dataset [22]) to vector traces with no kinematic information, and it consistently produces visually accurate reconstructions of the input. We observe that, while fluctuations may appear during the iteration, the refinement scheme consistently and rapidly converges towards a reduction of the error between the input and the generated trajectories.

The reconstructed $\Sigma\Lambda$ parameters provide a concise and easily manipulable representation of a geometric trace. This can be exploited in a number of applications that are relevant to our desired use cases in CAD [13] and procedural content generation [19]. For example, new instances of a given trace can be generated by randomly perturbing the virtual target positions and scaling the curvature and time overlap parameters, which results in variations that evoke multiple instances of writing by the same or multiple authors (Fig. 6). Furthermore, the $\Sigma\Lambda$ parameterisation is well suited for interactive editing applications [13]. As a result a user can easily adjust the output of the reconstruction in real time using a point and click procedure that is similar to the ones traditionally used in CAD software packages.

## VI. CONCLUSION

We demonstrated a simple method to reconstruct $\Sigma\Lambda$ parameters from solely the geometric trace (left by handwriting), which relies on a prior segmentation at perceptually salient points. The proposed method produces an accurate kinematic and geometric reconstruction of an input given just an ordered sequence of points. On the other hand, we consider this still as a stepping stone towards the development of more accurate and physiologically plausible methods, which exploit the same feature extraction and preprocessing steps. In future work, we intend to develop more sophisticated methods of choosing the $\Sigma\Lambda$ parameters $\mu_i$ and $\sigma_i$, which are currently experimentally set, and plan to explore in depth how the inferred kinematics relate to human data.

## REFERENCES

[1] A. Witkin, "Scale-space filtering," in *Proc. 8th Int. Joint Conf. Art. Intell.*, Karlsruhe, Germany, 1983, pp. 1019—22.

[2] C. De Stefano *et al.*, "A wavelet based curve decomposition method for on-line handwriting," *Advances in Graphonomics: Proc. of IGS*, 2005.

[3] F. Leymarie and M. Levine, *Curvature morphology*. McGill University Montreal, 1988.

[4] M. Leyton, "Symmetry-curvature duality," *Computer vision, graphics, and image processing*, vol. 38, no. 3, pp. 327–341, 1987.

[5] H. Blum *et al.*, "A transformation for extracting new descriptors of shape," *Models for the perception of speech and visual form*, vol. 19, no. 5, pp. 362–380, 1967.

[6] H. Blum, "Biological shape and visual science (part i)," *Journal of theoretical Biology*, vol. 38, no. 2, pp. 205–287, 1973.

[7] R. Plamondon, "A Kinematic Theory of Rapid Human Movements. Part I. Movement Representation and Generation," *Biological Cybernetics*, vol. 72, no. 4, pp. 295–307, March 1995.

[8] R. Plamondon *et al.*, "Recent developments in the study of rapid human movements with the kinematic theory: Applications to handwriting and signature synthesis," *Pattern Recognition Letters*, vol. 35, no. 1, pp. 225–235, January 2014.

[9] C. O'Reilly and R. Plamondon, "Automatic extraction of Sigma-Lognormal parameters on signatures," in *Proc. of 11th International Conference on Frontier in Handwriting Recognition (ICFHR)*, 2008.

[10] A. Fischer, R. Plamondon, C. O'Reilly, and Y. Savaria, "Neuromuscular representation and synthetic generation of handwritten whiteboard notes," in *Proc. of Frontiers in Handwriting Recognition (ICFHR)*, 2014, pp. 222–7.

[11] R. Plamondon and C. M. Privitera, "The segmentation of cursive handwriting: an approach based on off-line recovery of the motor-temporal information," *IEEE Transactions on Image Processing*, vol. 8, no. 1, pp. 80–91, 1999.

[12] R. Plamondon and W. Guerfali, "The 2/3 power law: When and why?" *Acta psychologica*, vol. 100, no. 1, pp. 85–96, 1998.

[13] D. Berio, F. Fol Leymarie, and R. Plamondon, "Computer aided design of handwriting trajectories with the kinematic theory of rapid human movements," in *18th Biennial Conference of the International Graphonomics Society*, 2017.

[14] A. Belyaev and S. Yoshizawa, "On evolute cusps and skeleton bifurcations," in *Shape Modeling and Applications, SMI 2001 International Conference on.* IEEE, 2001, pp. 134–140.

[15] R. Levien, "From spiral to spline: Optimal techniques in interactive curve design," Ph.D. dissertation, EECS Department, University of California, Berkeley, December 2009.

[16] M. Heald, "Rational approximations for the Fresnel integrals," *Mathematics of Computation*, vol. 44, no. 170, pp. 459–459, 1985.

[17] X. Li, M. Parizeau, and R. Plamondon, "Segmentation and reconstruction of on-line handwritten scripts," *Pattern recognition*, vol. 31, no. 6, pp. 675–684, 1998.

[18] D. Berio and F. Fol Leymarie, "Computational Models for the Analysis and Synthesis of Graffiti Tag Strokes," in *Computational Aesthetics*, P. Rosin, Ed. Eurographics Association, 2015, pp. 35–47.

[19] D. Berio, M. Akten, F. Fol Leymarie, M. Grierson, and R. Plamondon, "Calligraphic stylisation learning with a physiologically plausible model of movement and recurrent neural networks," in *Proc. of 4th Int'l Conf. on Movement Computing (MOCO)*, London, UK, 2017.

[20] P. Viviani and R. Schneider, "A developmental study of the relationship between geometry and kinematics in drawing movements," *Journal of Experimental Psychology: Human Perception and Performance*, vol. 17, no. 1, pp. 198–218, 1991.

[21] G. R. Lab, "Graffiti analysis database (000000book.com)," Web, 2009. [Online]. Available: http://000000book.com

[22] D. Llorens *et al.*, "The ujipenchars database: a pen-based database of isolated handwritten characters." in *Proc. of the 6th International Conference on Language Resources and Evaluation*, 2008.