

Chapter 11

Kinematics Reconstruction of Static Calligraphic Traces from Curvilinear Shape Features

Daniel Berio^{*}, Frederic Fol Leymarie^{*} and Réjean Plamondon[†]

Our goal is to be able to reproduce computationally calligraphic traces, such as found in the art practices of graffiti and various forms of more traditional calligraphy, while mimicking their production process. To this end, we propose a method that allows to reconstruct kinematics solely from the geometric samples of handwritten traces in the form of parameters of the *Sigma-Lognormal model*. We ignore the kinematics possibly embedded in the data in order to treat online data and vector patterns with the same procedure.

At the heart of our method, we develop a robust procedure to identify curvilinear shape features based on an analysis of local symmetry axes. These features determine the segmentation of a trace into circular arcs and guide an iterative reconstruction of the input kinematics and geometry in the form of Sigma-Lognormal parameters. We demonstrate how this parametrisation can be used to generate plausible kinematics for a static input trace, and how parameter variations can be exploited to generate traces that resemble the ones seen in real instances of human made calligraphy and graffiti.

1. Introduction

Many handwriting analysis methods rely on a prior segmentation of the written trace into constituent *primitives* or *strokes*. Some methods exploit the kinematics of the movement and segment the trajectory in correspondence with minima or other features of velocity [O'Reilly and Plamondon, 2008; Plamondon *et al.*, 2014]. Using the known inverse relation between speed and curvature [Viviani and Schneider, 1991], other methods rely on the identification of curvature extrema along the pen-trace [De Stefano *et al.*, 2005; Ferrer *et al.*, 2018], where movement slows down or has to stop momentarily to allow for a discrete change in orientation. 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.

^{*}Goldsmiths, University of London, U.K., Computing Dept.

[†]Polytechnique Montréal, Canada, Electrical Engineering Dept.

Most of the existing works are aimed either at a precise analysis of the kinematics of a digitised input or at the segmentation of a handwriting trace into components for biometric or pattern recognition purposes. On the other hand, our specific aim is perceptually and artistically driven, and we seek to *infer a physiologically plausible motion* from an input trace, the kinematics of which may be unavailable, such as when using vector graphics inputs, or may be degraded or unreliable due to the poor quality of a digitisation device, such as when using low cost tablets or trackpads. The motivation for this approach is grounded on the hypothesis that the visual perception of marks made by a drawing hand triggers activity in the motor areas of the brain [Freedberg and Gallese, 2007; Longcamp *et al.*, 2003], and further induces an approximate mental recovery of the (likely) movements and gestures underlying the artistic production [Freyd, 1983; Pignocchi, 2010]. We argue that this is particularly true for certain art forms such as expressed in calligraphy [Fong, 2003] and graffiti art [Berio *et al.*, 2017a; Mediavilla, 1996], in which the mastery of a skilful movement in large part determines the aesthetic quality of the resulting artefact.

Symbol	Meaning
$\mathbf{z}(s)$	Input trace (parametrised)
$\mathbf{z}_i(s)$	Trace segment
\mathbf{z}_i^{kmax}	Curvature maximum locus (on trace)
$\mathbf{z}_i^{lhs}(s)$	Left segment (for a given \mathbf{z}_i^{kmax})
$\mathbf{z}_i^{rhs}(s)$	Right segment (for a given \mathbf{z}_i^{kmax})
CC_i	Contact circle
\widehat{CC}_i	Contact circular arc
N_{CSF}	Number of Curvilinear Shape Features
N_S	Number of strokes (drawn gestures)
$\Sigma\Lambda$	Sigma Lognormal (SL) function
$\mathbf{slt}(t)$	SL generated trace
\mathbf{p}_i	Virtual or Aiming target per stroke

Table 1.: Main symbols.

In this chapter we take advantage of the duality between curvature and symmetry axes [Leyton, 1987] in order to extract more robustly curvilinear

shape features, such as those based upon extrema (of some curvature measure or approximation) along a handwriting or drawing trace. The method is also directly adaptable to open contours (with ends), to contours with breaks in curvature (non differentiable at some finite set of loci), and can further be used to identify loops (such as where a trace overlaps itself). Each feature is also explicitly paired with corresponding contact gauges (in the present case: circles) and a pair of curvilinear support regions: contour traces on each side of an identified extremum, where curvature is approximately monotonic. Given such a robust and rich feature description of an handwritten trace, we show how to exploit this spatial and structural geometric representation to infer the kinematics of a likely generative movement — as would be performed by a skilled human expert or artist, as predicted by the lognormality principle. To do so, we rely on the *Kinematic Theory of Rapid Human Movements* [Plamondon, 1995; Plamondon *et al.*, 2014], 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 [O'Reilly and Plamondon, 2008; Plamondon *et al.*, 2014; Fischer *et al.*, 2014; Ferrer *et al.*, 2018], we design our method with the goal of targeting applications in graphonomics, computer aided design (CAD) and 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 [1999]. We also argue that – from a computer graphics perspective– the ability to represent curves and traces through parameters that directly reflect the kinematics of a physiologically plausible movement opens up many possibilities, such as (i) more expressive rendering techniques, (ii) the generation of artificial variations that reflect the ones that would be made by a human, (iii) artistically driven fairing or beautification [Thiel *et al.*, 2011; McCrae and Singh, 2009; Zitnick, 2013], as well as (iv) potential applications in procedural content generation for movies and games.

In the following sections, we summarise first our approach to identify curvilinear shape features or CSF (Section 2). We then describe the trajectory seg-

mentation method (Section 3) based on CSF and Euler spiral fitting. In Section 4 we summarise the Sigma Lognormal ($\Sigma\Lambda$) model used to describe the motions of a pen-tip. In Section 5 we consider the recovery of $\Sigma\Lambda$ parameters *from geometry only*. In Section 6 we discuss some applications before concluding.

2. Curvilinear Shape Features

2.1. Background

The study of the curvature along a contour — being the trace of an object's boundary — has been the focus of decades of research in various fields, including that of visual perception and cognitive science, as well as computer vision and pattern recognition. Significant extrema of curvature (i.e. with associated contour segments having a role of support) have been shown empirically to be the most salient loci along piecewise smooth contours [Attneave, 1954; Feldman and Singh, 2005; De Winter and Wagemans, 2008] and to play an important role in the perceptual decomposition of objects into parts [Richards and Hoffman, 1985; Brault and Plamondon, 1993b,a; De Winter and Wagemans, 2006]. Curvature also plays an important role in the study of human movement and handwriting by relating the kinematics of a movement to its trace. It is well known that tangential speed and curvature of human hand movements are inversely proportional [Viviani and Schneider, 1991] and this relation can take the form of a power law [Viviani and Schneider, 1991; Plamondon and Guerfali, 1998]. Furthermore, curvature extrema are indicative of speed minima and therefore useful for the segmentation of complex movements into simpler primitives, for the study of traces in both the structural and kinematic domains.

A robust identification of curvature extrema from the curvature function alone can be difficult as it requires the evaluation of 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 removing 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 [Witkin, 1983]. Such a scale-space is very often produced by iterative Gaussian filtering in the spatial domain, or via the frequency domain using wavelets [De Stefano *et al.*, 2005]. An alternative to such traditional filtering — which tends to

blur away details, especially in the vicinity of corners with sharp changes in orientation — 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 [Leymarie and Levine, 1989]. Nevertheless, such methods operating directly on the curvature function (along a contour) still suffer from poor localisation, and do not capture well singularities which can be perceptually significant, such as curvature discontinuities which are typical of corner features.

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 [Leyton, 1987]. Originally pioneered by Harry Blum in the 1960's for the study of biological shape [Blum, 1967], 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 [Leymarie and Levine, 1992]. Variants of the SAT have also been widely used in pattern recognition applications, especially to extract topological and structural knowledge from the outline of handwritten or printed characters [Li and Plamondon, 2006]. 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 [Blum, 1973] — in the latter case becoming similar to the Voronoi graph.

2.2. *Our solution*

In order to robustly identify useful curvilinear shape features, we exploit the duality [Leyton, 1987] between the two representations of (2D) contour curvature and SAT, which allows us to identify *significant* curvature extrema and discontinuous breaks along a handwriting or drawing trace through the analysis of its local symmetries. Each feature is paired with a corresponding contact circle (near the extrema) and its support region — where curvature is approximately monotonic and in correspondence with a local symmetry axis. We then exploit this intermediate representation to infer the kinematics of a movement that reconstructs a likely trace.

We recall a result presented by Leyton which links the symmetry axes of an object having a smooth bounding contour to its curvature extrema [Leyton, 1987]:

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.

Motivated by this concept,^a given a symmetry axis, it is then possible to identify and locate a curvature extrema near one axis end. In a traditional setting, the SAT is computed at once for a given final contour or, as in our case, a written trace. This however does not allow to identify all perceptually significant curvature extrema, as part of a contour may forbid or *mask* the existence of a symmetry axis that would otherwise end at a curvature break or corner, or end at the center of curvature of the circular arc associated to a curvature extremum.^b Furthermore, data from contours or handwritten traces typically are not smooth everywhere, and thus we need a solution which is adapted to discrete samplings, possibly with sharp changes in orientation (corresponding to a break in curvature for the equivalent continuous trace), as well as for open contours (with ends).

Inspired by Leyton's idea, we have explored the following discrete scheme, as illustrated step by step, in Fig. 1: (i) we select a starting point either at one end of an open contour or randomly along a closed contour; (ii) we traverse the contour in one direction (until another end or until we come back to our initial position); (iii) we initiate a local symmetry axis computation from the sequential set of encountered trace samples (e.g. using a well established local Voronoi method [Ogniewicz, 1992]) and identify a first local axis end; (iv) we pursue the local axis reconstruction until we detect a new potential axis end — this indicates the end of one curvilinear side of the associated region of support — (v) we repeat the computation of a local axis for the new axis end, ignoring old samples which are past the previously identified curvature extremum (i.e. associated to the previously identified local axis end) — for this new axis end we already know its first curvilinear region of support, and thus we keep refining the local axis computation by visiting more contour samples, until we hit yet another candidate axis end or run out of samples; (vi) the procedure repeats until no more samples are left to consider.

In summary, a local symmetry axis is evaluated as one travels along a planar contour, such as obtained from a written trace, denoted $z(s)$, which is assumed

^aLeyton's result [Leyton, 1987] is restricted to smooth contours and to the description of so-called "codons": contour segments delimited by a triplet of curvature extrema such that a pair of concavities (convexities) delimit an intermediate convexity (concavity).

^bOne way to prove this masking effect, is by considering the behavior of cusps of evolutes in relation to a symmetry axis. Belyaev and Yoshizawa [2001] proved 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.

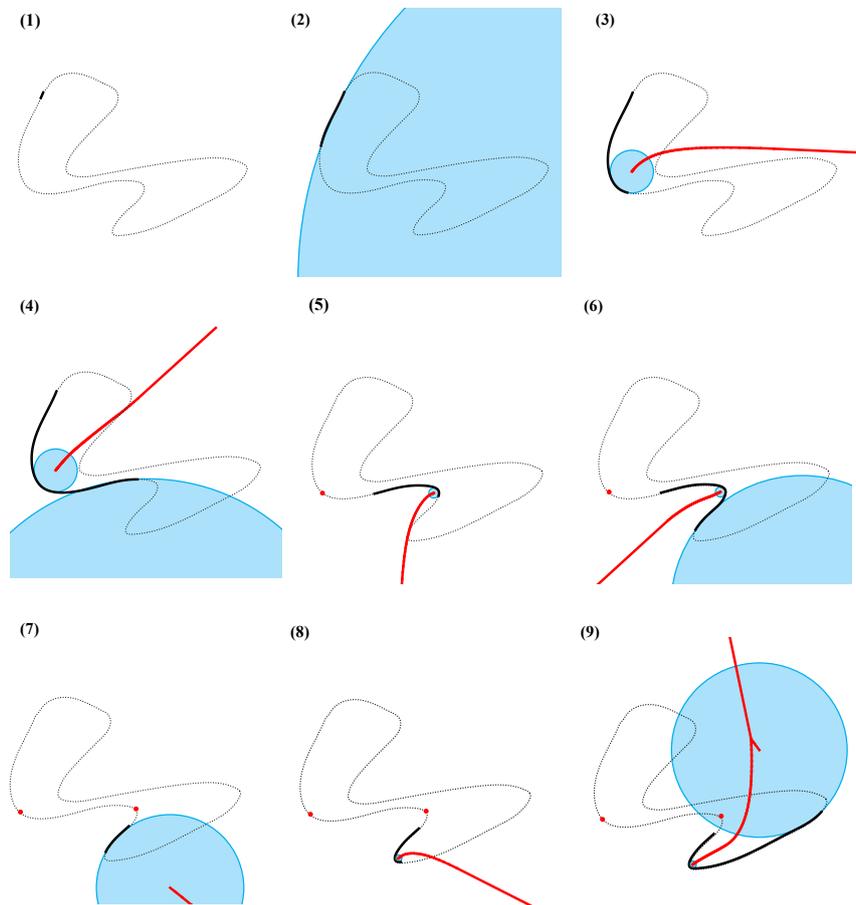


Fig. 1.: Significant steps of computation of the swept symmetry axis. In cyan, the disks for the extremities of the axes (red); in black the current contour segment. The small red circles are the extrema. **(1)**, the first step of the process. The contour segment is too short so there are no branches. **(2)**, a first branch appears (out of the figure) with the corresponding disk. The disk is unstable, i.e. it tends to shrink and moves towards a curvature extrema. **(3)**, the disk stabilises in proximity of a curvature extrema. **(4)**, a new axis appears; this triggers an event in which **(5)** the extrema corresponding to the disk in (3) is marked and a new start point is set for the contour segment. The new start point corresponds with the first anchor point of the newly appeared disk. The remaining images show steps identical to (3, 4, 5) which are repeated iteratively until all significant symmetry axes have been found. Note that in **(9)** a new symmetry branch appears, which will lead to the identification of the subsequent axis and associated curvature extrema along the outline.

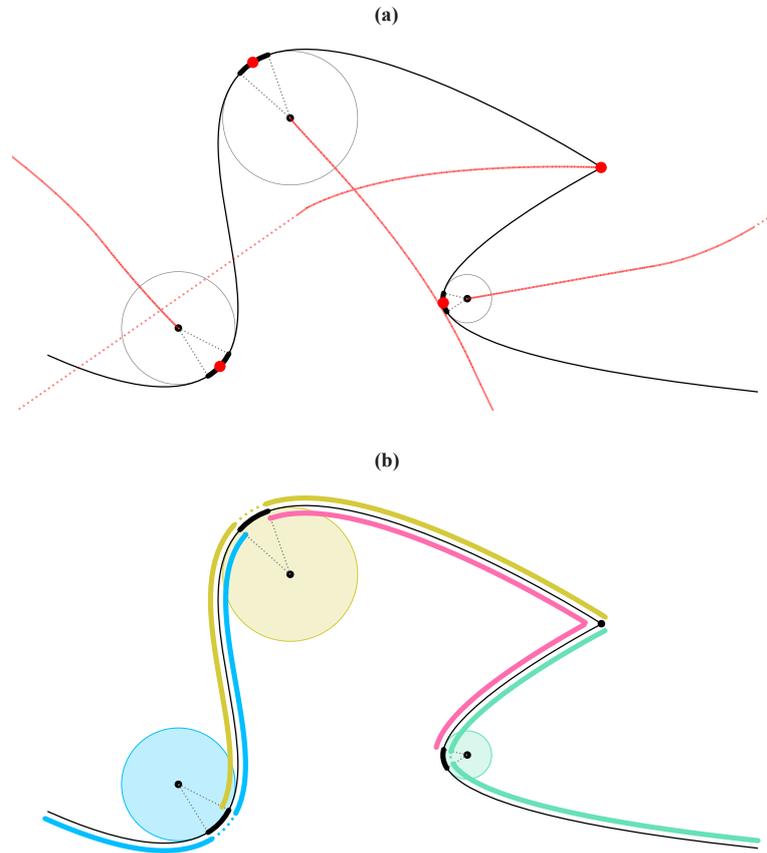


Fig. 2.: (a) Four successive Curvilinear Shape Features (CSFs): Each CSF is defined by (i) a symmetry axis, SA_i , (ii) a contact circle, CC_i , (iii) its corresponding circular arc, \widehat{CC}_i , part of the local trace, (iv) a representative curvature extremum locus, z_i^{kmax} , taken as the mid-point of \widehat{CC}_i (indicated as red dots), (v) a pair of contour segments, $[z_i^{lhs}(s), z_i^{rhs}(s)]$ delimiting the region of influence of a CSF. Note that the third CSF (from the left) corresponds to a typical discontinuity in curvature, a corner where there is a sudden change in orientation, and thus with a contact circle reduced to a point which coincides with the curvature extremum locus. This is also indicated by having the local symmetry axis reaching the input trace $(z(s))$. (b) Each pair of contour segments for each CSF is emphasised: colored and slightly translated away from the original outline.

parametrised by arc length s .^c Each *significant* axis is found,^d such that its ex-

^cIn practice, we sample, for example uniformly, the trace in a set of discrete points at approximately equal distance from each other. Other (adaptive) sampling strategies are possible, e.g. in the presence of corners.

^dSignificance is linked to the particular computation of symmetry axial structure; for the results reported here, we rely on the well established parametric method of Ogniewicz [1992].

istence ends once another significant axis emerges, and the previous written trace already traversed is then "forgotten", so as to not mask other potential extrema of curvature associated to later parts of the trace.

Definition 1 (Curvilinear Shape Feature (CSF)). Each CSF is composed of five main elements (Fig. 2): (i) SA_i , a computed significant symmetry axis, (ii) CC_i , a contact circle centered at the initial tip of a given SA_i , (iii) \widehat{CC}_i , the associated circular arc overlapping with the contour trace, (iv) z_i^{kmax} , the associated curvature maximum locus, which we take as the mid-point of \widehat{CC}_i ,^e (v) $[z_i^{lhs}(s), z_i^{rhs}(s)]$, a pair of contour segments on each side of \widehat{CC}_i . The index i indicates one of N_{CSF} computed CSF: $1 \leq i \leq N_{CSF}$.

We note that each SA_i is not necessarily part of the classic Blum medial axis set, nor the Voronoi graph; rather, it is always part of the Full Symmetry Set as defined in singularity theory [Bruce and Giblin, 1992]. Furthermore, a given arc, \widehat{CC}_i , may vanish in size when coinciding with a break of curvature or sharp corner tip, becoming identically z_i^{kmax} . We also emphasise that this definition of a CSF is more general than the older concept of a (contour-based) "codon": a triplet of alternate (in sign) curvature maxima (e.g. concave, convex, concave) [Richards and Hoffman, 1985].

3. Trajectory segmentation into circular arcs

The proposed trajectory reconstruction method exploits the prior feature analysis of the input $z(s)$ (Section 2), and thus takes as its input the set of computed Curvilinear Shape Features (CSF) which identifies a set of contact circles, CC_i , circular arcs, \widehat{CC}_i , curvature extremum loci, z_i^{kmax} , and pairs of contour segments, $[z_i^{lhs}(s), z_i^{rhs}(s)]$, while the corresponding axial branches, SA_i , are ignored — their essential role in our method being in identifying the other items.

On the basis of this information, our goal is then to segment the entire trace, $z(s)$, with a series of best fitting circular arcs. The reason being that circular arcs can be directly exploited by the kinematic analysis based on the Sigma lognormal modelling (to be described in the following sections 4 and 5). To obtain a trajectory segmentation in terms of a series of circular arcs, we start from the set \widehat{CC}_i and then propose in what follows a systematic method to approximate the remaining parts of the trace (given by the set of pairs $[z_i^{lhs}(s), z_i^{rhs}(s)]$) in two steps: (i) first by tightly fitting Euler spiral segments, and then (ii) mapping

^eAlternatively, z_i^{kmax} can be taken as the projection of SA_i intersecting the contour trace, a definition closer to the spirit of the work of Leyton [1987].

each Euler spiral segment to up to four best fitting circular arc(s), as a function of the presence of an inflection along the spiral segment and the angular extent of each arc.^f

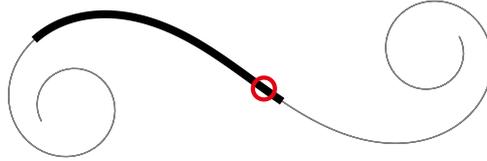


Fig. 3.: An Euler spiral, its inflection point (circle) and a Euler spiral segment (black).

3.1. Euler spiral fitting

Euler spirals (also known as Cornu spirals, or clothoids) [Levien, 2009] are a useful type of curves in which curvature varies linearly with arc length, permitting the description of variably curved segments which may contain an inflection (Fig. 3). An Euler spiral is commonly parameterised by arc length using the cosine ($C(t)$) and sine ($S(t)$) Fresnel integrals:

$$C(t) = \int_0^t \cos(u^2) du, \quad S(t) = \int_0^t \sin(u^2) du. \quad (1)$$

Such that the Euler spiral is obtained as:

$$\mathbf{q}(t) = (x(t), y(t)) = (C(t), S(t)), \quad (2)$$

where t can vary from minus infinity to plus infinity, and where the origin $(0, 0)$ corresponds to $t = 0$ which is the inflection point for the spiral; note that t is then identically the (signed) arc length parameter for the spiral curve. In our application, an Euler spiral segment to be fitted to the data is defined between an initial ($t = t_1$) and final ($t = t_2$) parameter values. If the values alternate in sign, then we have a segment with an inflection (at $t = 0$). Such a segment can be conveniently computed and sampled in an efficient manner using an approximation method developed by Heald [1985], which results in n samples along the segment from $\mathbf{q}(t_1)$ and $\mathbf{q}(t_2)$.

In order to fit an Euler spiral segment to one of the segments from our input trace, we first compute approximate tangent directions along the trajectory, for a given segment, $\mathbf{z}_i(s)$, *i.e.*, in correspondence with the initial and final points of the segment under examination. This allows to rapidly compute a first estimate

^fOur use of Euler spirals is also inspired by the work of Leyton who studied contour regions between curvature extrema of opposite sign by using "bi-spiral" segments [Leyton, 1987].

of the spiral segment's initial and final parameter values using a secant method described by Levien [2009].

However, the tangent estimates are likely to be unreliable in the presence of noisy input data, and thus we proceed to refine this initial fit with a least squares optimisation based on the classic Gauss-Newton method. Our method consists then in three additional steps. First, we linearly transform the given trace segment, $\mathbf{z}_i(s)$, such that its end points match those of the computed spiral segment in its canonical form. Second, we modify the canonical form of the Euler spiral, by introducing a scaling factor α and a rotation by an angle ω (in radians) with:

$$\mathbf{q}(t) = \begin{bmatrix} \alpha \cos(\omega)C_1(t) - \alpha \sin(\omega)S_1(t) \\ \alpha \sin(\omega)C_1(t) + \alpha \cos(\omega)S_1(t) \end{bmatrix}, \quad \text{where} \quad (3)$$

$$C_1(t) = C(t) - C(t_1) \quad \text{and} \quad S_1(t) = S(t) - S(t_1). \quad (4)$$

Note that the (initial) canonical form is for $\alpha = 1$ and $\omega = 0$. Third, and finally, we proceed with the minimisation:

$$\min_{t_1, t_2, \alpha, \omega} \frac{1}{2} \sum_{j=1}^n \|\mathbf{q}[j] - \mathbf{z}_i[j]\|^2, \quad (5)$$

where $\mathbf{q}[j]$ and $\mathbf{z}_i[j]$ both denote n equally spaced samples with a sampling index $[j]$, in the former case for the spiral segment $\mathbf{q}(t)$ between t_1 and t_2 , and in the later case along the input trace segment $\mathbf{z}_i(s)$.

3.2. Inflections and final circular arc decomposition

The presence of inflections can easily and robustly be identified by checking the signs of the two spiral parameters t_1 and t_2 . With alternate signs, $t = 0$ gives the inflection position. For each potential inflection, we then check if the ratio $[\min(|t_1|, |t_2|)]/|t_1 - t_2|$ is less than a user defined threshold ϵ^{flex} (which we empirically set to 0.2 in the accompanying examples), in which case we discard it as a near degenerate case, the inflection being very close to one spiral's end point.

Depending on the presence of an inflection, we fit either one or two circular arcs to each Euler spiral segment. A final simple refinement step is then executed by considering the angular extent of each computed arc. For relatively wide arcs, we split these in two halves. The heuristic used here being that arcs should not be too wide such that the later computation of the kinematics converges more easily.

The internal angle of any circular arc derived from Euler spiral segments is easily estimated by integrating the curvature of the spiral and distinguishing

between three cases: (i) For the case of two arcs separated by an inflection, the pair of internal angles are given by $t_1 \times |t_1|$ and $t_2 \times |t_2|$ (Fig. 4.(a)). (ii) 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. (iii) When no inflection is present the internal angle is given by: $|(t_2 \times |t_2| - t_1 \times |t_1|) \times \text{sgn}(t_1)|$. In each of these three cases, and for each arc, we then check if the absolute internal angle is greater than $(3/4)\pi$ — heuristically set, so that no derived circular arc is close to half a circle, similarly to Li *et al.* [1998] — in which case we further subdivide such a wide arc in two halves (Fig. 4.(b)).

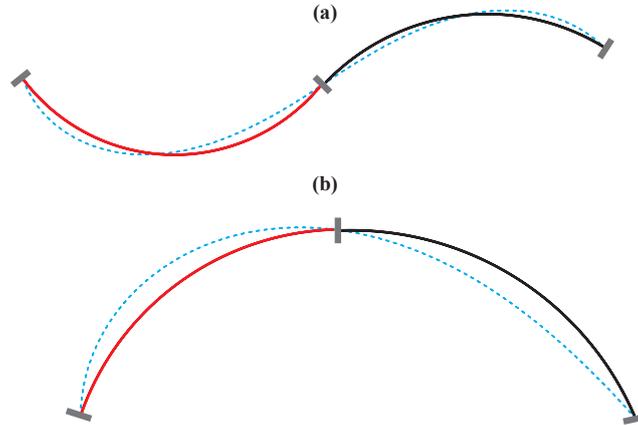


Fig. 4.: Decomposing Euler spirals (stippled cyan) into arcs. (a) two arcs delimiting an inflection. (b) one segment with internal angle $> (3/4)\pi$ divided into two equal arcs.

In summary, we have that the trace is now represented by a sequence of N_{CSF} contact circular arcs, \widehat{CC}_i , with intermediate Euler spiral segments each with or without an inflection. Each spiral segment is then mapped to either a pair of arcs (with a separating inflection) or a single arc. Each such circular arc derived from a spiral may then be further split in two if wide enough. A little inspection shows that, in the case of a closed trace (no endpoints), we have a maximum of $5 \times N_{CSF}$ circular arcs fitted, while for an open trace (with a pair of endpoints) the maximum is $N_{CSF} + 4 \times (N_{CSF} + 1)$. Fig. 5.(a) shows the results of (i) identifying (here, five) curvature maxima, z_i^{kmax} , followed by (ii) fitting (five) Euler spirals, and (iii) finding (two) inflections and corresponding circular arcs. This approximate reconstruction of the original trajectory in the form of circular arc segments, is related to the method originally proposed by Li *et al.* [1998],

but with the following main three differences: (i) We have found experimentally that our method to identify curvature maxima is more robust — in particular as it does not rely on an explicit *a priori* estimation of the curvature signal. (ii) We use Euler spirals to fit intermediate data which gives a simpler and more robust method to identify inflections. (iii) We explicitly obtain contact circle arcs, \widehat{CC}_i , which results in a more accurate reconstruction of the original trace (Fig. 5).

This representation of the input trace, $z(s)$, as a series of circular arcs, is now ready to be exploited (in §5) together with the Signal Lognormal model which we first summarise in the next section.

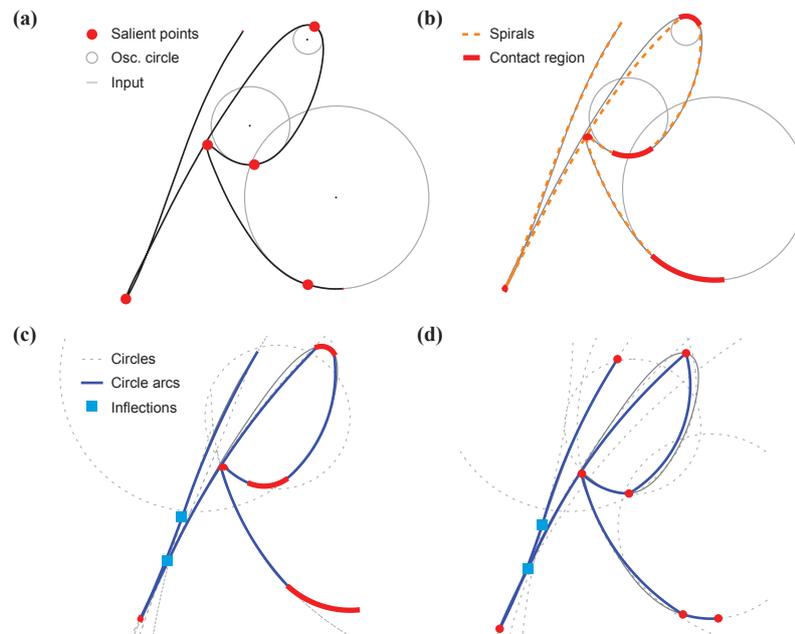


Fig. 5.: Sample “R” character from the UJI handwritten character dataset [Llorens *et al.*, 2008]. (a) Significant curvature maxima, z_i^{Kmax} . (b) Euler spiral fitting; and (c) circular arcs decomposition, where the arcs in red indicate the contact circular arcs, \widehat{CC}_i . (d) Demonstrative example of least-squares fitting of circular arcs to the segments defined between (the same) consecutive curvature maxima, z_i^{Kmax} . Conclusion: Not considering the contact region results in a much less precise reconstruction of the input.

4. The Sigma Lognormal ($\Sigma\Lambda$) Model

On the basis of the Kinematic Theory [Plamondon, 1995], we study handwriting and drawing movements via the the Sigma Lognormal ($\Sigma\Lambda$) model [Plamondon *et al.*, 2014], which describes complex trajectories on the plane via the vectorial superimposition of N_S time shifted stroke primitives.

Each stroke is described as an aiming drawing gesture, and the sequence of strokes is modelled by a scaffold-like action plan, starting from an initial position, and defining a curvilinear graph made of vertices as aiming positions with connecting circular arcs. The integration of the $\Sigma\Lambda$ along this scaffold provides for a drawing engine where variations of parameters give different possible (but related) approximating traces, more or less smooth, more or less reaching the various aiming vertices (aka “virtual targets”, Fig. 6). We summarise the main algorithmic elements of the $\Sigma\Lambda$ model in the remaining of this section.

The velocity function of each stroke is given by a classic lognormal function:

$$\Lambda_i(t) = \frac{1}{\sigma_i \sqrt{2\pi}(t - t_{0i})} \exp\left(-\frac{(\ln(t - t_{0i}) - \mu_i)^2}{2\sigma_i^2}\right) , \quad (6)$$

which characterises the impulse response of each stroke to a centrally generated command occurring at time t_{0i} . The parameters μ_i and σ_i represent the stroke delay and response time in a logarithmic scale; they also determine the shape and asymmetry of the lognormal profile.

4.1. Circular arc assumption

An important practical assumption is typically made when initiating the $\Sigma\Lambda$ model: *handwriting* movements are mostly made with rotations of the elbow or wrist. The corollary is then that the curvilinear evolution of a drawing stroke can be approximated by a *circular arc*. This has for consequence to simplify the computation of the *angular evolution* of a stroke by using the time integral of equation (6):

$$w_i(t) = \int_0^t \Lambda_i(u) du = \frac{1}{2} \left[1 + \operatorname{erf}\left(\frac{\log(t - t_{0i}) - \mu_i}{\sigma_i \sqrt{2}}\right) \right] , \quad (7)$$

which results in

$$\phi_i(t) = \theta_i - \frac{\delta_i}{2} + \delta_i w(t) , \quad (8)$$

such that θ_i is the *direction* of the stroke and δ_i is the *stroke curvature* parameter which determines the internal angle of the assumed *circular arc*.

The planar pen-tip velocity is then calculated with:

$$\dot{x} = \sum_{i=1}^N \hat{D}_i \Lambda_i(t) \cos(\phi_i(t)) \quad \text{and} \quad \dot{y} = \sum_{i=1}^N \hat{D}_i \Lambda_i(t) \sin(\phi_i(t)) \quad , \quad (9)$$

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

$$h(\theta_i) = \begin{cases} \frac{\delta_i}{2\sin(\delta_i/2)} & \text{if } |\sin\theta_i| > 0 \quad , \\ 1 & \text{otherwise} \quad , \end{cases} \quad (10)$$

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 [Plamondon and Guerfali, 1998]:

$$\ddot{x} = \sum_{i=1}^N \hat{D}_i \dot{\Lambda}_i(t) \cos(\phi_i(t)) - \hat{D}_i \delta_i \Lambda_i^2(t) \sin(\phi_i(t)) \quad , \quad (11)$$

$$\ddot{y} = \sum_{i=1}^N \hat{D}_i \dot{\Lambda}_i(t) \sin(\phi_i(t)) + \hat{D}_i \delta_i \Lambda_i^2(t) \cos(\phi_i(t)) \quad , \quad (12)$$

with

$$\dot{\Lambda}_i(t) = \Lambda_i(t) \frac{\mu_i - \sigma_i^2 - \log(t - t_{0i})}{\sigma^2(t - t_{0i})} \quad , \quad (13)$$

which allows us to compute the curvature function at time t with the classic formula from differential geometry:

$$\kappa(t) = (\dot{x}\ddot{y} - \dot{y}\ddot{x}) / (\dot{x}^2 + \dot{y}^2)^{3/2} \quad . \quad (14)$$

Given equation (7) the planar displacements along the $\Sigma\Lambda$ trajectory can then be computed using the error function and thus not requiring the numerical integration of equation (9), which simplifies to:

$$x = \sum_{i=1}^N \begin{cases} \frac{\hat{D}_i}{\delta_i} (\sin(\delta_i w(t) + \theta_i - \delta_i/2) - \sin(\theta_i - \delta_i/2)) & \text{if } \delta_i > 0 \quad , \\ \hat{D}_i w_i(t) \cos\theta_i & \text{otherwise} \quad , \end{cases} \quad (15)$$

$$y = \sum_{i=1}^N \begin{cases} \frac{\hat{D}_i}{\delta_i} (\cos(\delta_i w(t) + \theta_i - \delta_i/2) - \cos(\theta_i - \delta_i/2)) & \text{if } \delta_i > 0 \quad , \\ \hat{D}_i w_i(t) \sin\theta_i & \text{otherwise} \quad . \end{cases} \quad (16)$$

4.2. Targets

The previous pair of equations for displacement give us a way to efficiently compute a new position, $\mathbf{s}l\mathbf{t}(t)$, at a given time t from an initial position \mathbf{p}_0 :

$$\mathbf{s}l\mathbf{t}(t) = \mathbf{p}_0 + [x, y]^\top . \quad (17)$$

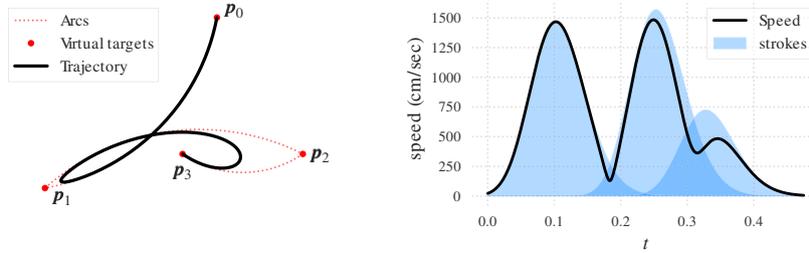


Fig. 6.: Left: Lognormal trajectory (in black) with the corresponding action plan (targets as red dots, and scaffold of circular arcs as dotted red curves). Right: The corresponding speed profile with different time overlaps between lognormals.

The *initial* sequence of curvilinear strokes describes an *action plan* consisting of the initial position \mathbf{p}_0 followed by a sequence of N_S targets $\mathbf{p}_1, \dots, \mathbf{p}_{N_S}$ each corresponding to an aiming locus per stroke (Fig. 6, Left).

The degree of *time overlap* between lognormal components is used to characterise the degree of smoothness of the trajectory in correspondence with each target, where a greater overlap results in a smoother trajectory (Fig. 6, Right). In order to facilitate interactive applications [Berio *et al.*, 2017b] and to simplify the subsequently described parameter reconstruction method, we compute the directions θ_i and length D_i for all N_S pairs of consecutive targets $(\mathbf{p}_{i-1}, \mathbf{p}_i)$. Furthermore, we explicitly define the time overlap of each lognormal through an intermediate parameter $\Delta t_i \in [0, 1]$ where $t_{0i} = t_{0i-1} + \Delta t_i \sinh(3\sigma_i)$ for $i > 1$ and $t_{01} = 0$.

5. Iterative Reconstruction of $\Sigma\Lambda$ parameters

Having introduced the essential elements of the $\Sigma\Lambda$ model in the previous section, and given the earlier trajectory segmentation derived from identified CSFs (§2) and circular arcs derived from Euler spiral fitting (§3), we have the necessary information to describe how we reconstruct the input trajectory with an approximate associated kinematics given only information about its (static) sampled geometry. Hence, we will be able to seamlessly process on-line handwriting data as well as vector art in which only the sequential ordering and coordinates of trace samples is required. The method is a development and improvement over our prior efforts [Berio and Fol Leymarie, 2015; Berio *et al.*, 2017a]. We re-emphasise that, although a number of methods exist for the accurate reconstruction of $\Sigma\Lambda$ parameters from digitised traces [O'Reilly and Pla-

mondon, 2008; Plamondon *et al.*, 2014; Fischer *et al.*, 2014], these require as input the kinematics of the original trajectory.

5.1. Initialisation: Features, Strokes, Initial Targets

The initial set of aiming *targets* (aka “virtual targets”) consists of three types of *feature points*, or *features* for short: from CSF analysis (i) recovered curvature maxima loci, \mathbf{z}_i^{Kmax} , and from Euler spiral analysis (ii) inflections, and (iii) splits (of wide angled circular arcs). We can either directly used these loci or find their nearest neighbors, $\mathbf{z}(\hat{s}_i)$, on the original input trace, $\mathbf{z}(s)$, which leads to slightly more accurate reconstructions. We follow the later approach in results reported thereafter.

The initial set of N_S *strokes* is defined from only those circular arcs derived from the Euler spiral segment fitting; i.e. we do not generate strokes for each contact circular arc, \widehat{CC}_i , associated to each \mathbf{z}_i^{Kmax} . We have found experimentally that if we include strokes for such arcs and compute trajectories via the $\Sigma\Lambda$ model we easily overshoot the original trace, $\mathbf{z}(s)$, in the vicinity of each \mathbf{z}_i^{Kmax} . Also, removing these contact circular arcs, \widehat{CC}_i , from the $\Sigma\Lambda$ modeling gives us more flexibility in aiming at curvature maxima loci when generating smooth variants (using equation (17)), and also it reduces the overall possible number of strokes considered.

However, the set \widehat{CC}_i can be used for defining stopping criteria for our iterative scheme, as defined below. Intuitively, as a generated trace, $\mathbf{slt}(t)$ aims at reaching towards a curvature maximum locus, \mathbf{z}_i^{Kmax} , we can stop locally pulling in that direction once the generated equivalent curvature maximum locus enters a zone such as the angular sector defined by a \widehat{CC}_i or even as it enters the interior of the contact circle itself, CC_i .

The used circular arcs — all derived from Euler spiral segments — give us a set of internal angles $\hat{\theta}_{i=1}^{N_S}$, centers $\mathbf{c}(\hat{\theta}_i)$ and radii $r(\hat{\theta}_i)$. The circular arcs are delimited by N_S feature points $\{\hat{s}_i\}_{i=1}^{N_S}$, plus additional starting and end loci if modeling an open trace (with endpoints, \mathbf{p}_0 and \mathbf{p}_{N_S+1}). If modeling a closed trace, we randomly pick one feature point as both the start and end positions.

An initial estimation of the trajectory parameters is given by a target sequence $\mathbf{p}_i = \mathbf{z}(\hat{s}_i)$, 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 σ_i and μ_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, $\mathbf{z}(s)$, and to be much smoother due to the initial lognormal stroke overlap (Fig. 7.(a)).

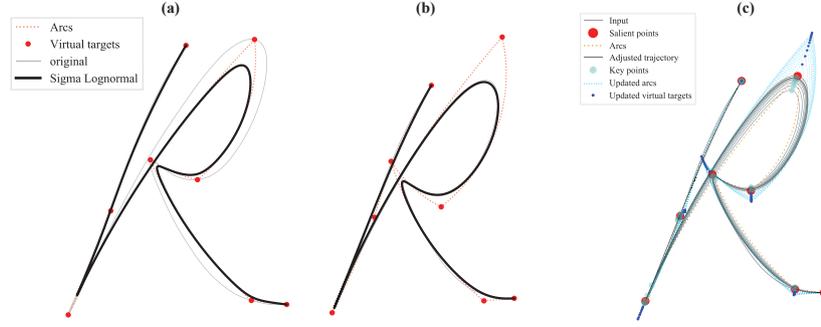


Fig. 7.: $\Sigma\Lambda$ parameter reconstruction using features from CFSs and Euler spiral derived arcs. **(a)** First guess (in black) of the stroke parameters and action plan from features. **(b)** Reconstruction of the input after iterative refinement steps. **(c)** Iterative refinement steps. The initial action plan has targets corresponding with the features along the input (large red circles). At every iteration, the targets are shifted (small blue circles) in order to reduce the distance between keys along the generated trajectory (cyan circles) and the features of the original trace.

5.2. Iterative scheme: Keys, Max speeds, Moving Targets

To improve the reconstruction, we adopt an iterative refinement scheme (Fig. 7) in which we adjust the curvature and time overlap parameters together with the target positions in order to minimise the difference between the reconstructed, $\mathbf{slt}(t)$, and original, $\mathbf{z}(s)$, trajectories. At each iteration, we rely on the estimation of a series of *key points*, or *keys* for short, which approximate the initial feature point loci. We compute N_S keys $\{\tau_i\}_{i=1}^{N_S}$ along the trajectory (Fig. 8) where $\tau_1, \dots, \tau_{N_S}$ indicates the time occurrence at which the influence of one stroke exceeds the previous one and the generated curvature is locally maximal, while τ_0 and τ_{N_S+1} are the starting and ending time of the trajectory. To compute the keys, we intersect the scaled profiles of consecutive stroke pairs, using an iterative or Newton scheme, by solving:

$$\hat{D}_i \Lambda_i(t) - \hat{D}_{i+1} \Lambda_{i+1}(t) = 0 \quad .$$

In addition, we compute N_S *maximum speed points*, or *max speeds* for short, $\{\gamma_i\}_{i=1}^{N_S}$, which indicate the time occurrence of the maximum speed for each stroke (Fig. 8); this is explicitly obtained by the mode of the corresponding lognormal: $t_{0i} + \exp(\mu_i - \sigma_i^2)$.

The iterative refinement scheme was designed based on three observations:

Observation 1. The time parameter Δt_i is proportional to the curvature $\kappa(\tau_i)$ at the time of the corresponding key point. Thus, a higher value of Δt_i will decrease the amount of overlap of successive lognormals. This will result in a

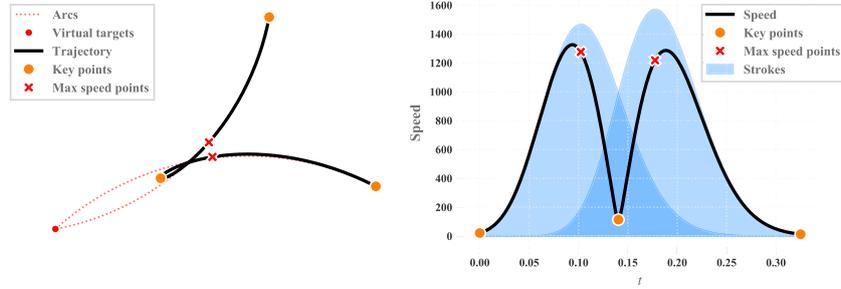


Fig. 8.: Keys (orange circles) and max speeds (red crosses) overlaid on the trace (a) and speed profile (b) of a trajectory with two strokes.

lower speed and higher curvature $\kappa(\tau_i)$ at the time occurrence of the key. 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 Δt_i proportionally at each iteration. We observe that changes in Δt_i are not linearly related to changes in the curvature $\kappa(\tau_i)$ at the corresponding key. In order to compensate for this, we assume a 1/3 power relation [Viviani and Schneider, 1991] which has often been observed in human movement and particularly holds for elliptical portions of the trajectory [Plamondon and Guerfali, 1998], which is often the case near keys. The reasoning is that given the relations

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

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

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

As a result, given the *power law* [Viviani and Schneider, 1991] $v = \kappa^{-1/3}$ and because velocity and Δt are inversely proportional, we finally get the relation:

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

Observation 2. Moving targets play a role similar to control points in spline analysis. Shifting a target \mathbf{p}_i in a given direction will cause the generated key $\mathbf{s}lt(\tau_i)$ to move in a similar direction. As a result, shifting the target \mathbf{p}_i along the vector $\mathbf{z}(\hat{s}_i) - \mathbf{s}lt(\tau_i)$ will decrease the distance between successive generated keys and original features (Fig. 7.(c)).

Observation 3. The distance D_i between successive targets \mathbf{p}_i and \mathbf{p}_{i-1} will influence the curvature of the resulting stroke. Augmenting this distance will increase the radius of curvature of the circular arc defined by the parameter δ_i and will result in a decrease of curvature for the stroke. While the trajectory tends to depart from the circular arc near the keys 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 use this locus to evaluate the deviation from the desired arc $\hat{\theta}_i$ and correct the parameter δ_i accordingly.

As a result of these observations, we define each iteration of our method to consist of the following ordered steps:

$$\Delta t_i \leftarrow \Delta t_i + \lambda_\Delta (\zeta(\hat{\Delta t}_i, \Delta t_{min}, \Delta t_{max}) - \Delta t_i) \quad , \quad (18)$$

$$\delta_i \leftarrow \delta_i + \lambda_\delta (\hat{\delta}_i - \delta_i) \quad \text{and} \quad (19)$$

$$\mathbf{p}_i \leftarrow \mathbf{p}_i + \lambda_p (\mathbf{z}(\hat{s}_i) - \mathbf{s}l\mathbf{t}(\tau_i)) \quad . \quad (20)$$

Here λ_Δ , λ_δ and λ_p are damping parameters that we experimentally tune to 0.1 and 0.1 and 0.5 to avoid excessive adjustments at each iteration. The target time offset parameter for each iteration is computed by assuming a 1/3 power relation to curvature and is given by

$$\hat{\Delta t}_i = \zeta \left((\kappa(\hat{s}_i) / \kappa(\tau_i))^{1/3}, \Delta t_-, \Delta t_+ \right) \quad , \quad (21)$$

which is restricted to a user specified range $[\Delta t_-, \Delta t_+]$ by using a sigmoid function ζ . We observe that this restricted range improves convergence of our method and permits to apply smoothing effects to the trajectory during the reconstruction step (examples in §6).

The desired internal angle of an arc is given by:

$$\hat{\delta}_i = 4 \tan^{-1} \left[\frac{h}{a} \tan \left(\frac{\delta_i}{4} \right) \right] \quad \text{with} \quad (22)$$

$$a = \|\mathbf{p}_i - \mathbf{p}_{i-1}\| \quad \text{and} \quad (23)$$

$$h = (r(\hat{\theta}_i) - \|\mathbf{p}(\gamma_i) - \mathbf{c}(\hat{\theta}_i)\|) \text{sgn}(\hat{\theta}_i) \quad , \quad (24)$$

where the term h determines the amount to shift the curvature parameter δ_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 max speed point $\mathbf{p}(\gamma_i)$.

5.3. Stopping Criteria, SNR

A few different stopping criteria for the iterative scheme are possible, depending on the user needs. The simplest — and most practical for experimenting

with the approach — is to let the user define an overall maximum iteration. Other more sophisticated criteria we have experimented with include: (i) let keys reach each associated CC_i or \widehat{CC}_i ; (ii) minimise the overall distance between the generated, $\mathbf{slt}(t)$, and the input, $\mathbf{z}(s)$, traces, by either selecting a threshold value or letting the algorithm reach a local minimum; (iii) optimise the quality of the reconstruction by maximising an error criterion such as the SNR (defined next). We have found in practice the latter SNR-based criterion gives a good compromise between reconstruction quality and computational complexity.

Because we do not take into consideration the kinematics of the input, we evaluate the quality of the reconstruction using the Signal to Noise Ratio (SNR) computed between the reconstructed and input trajectory. While this could be done by uniformly sampling the two trajectories at a constant distance step, this will result in a propagation of errors along the reconstructed trajectory, which leads to unreliable SNR measurements. To overcome this problem, we exploit our initial estimation of features $\mathbf{z}(\hat{s}_i)$ in the input and the segmentation given by the keypoints τ_i of the reconstructed trajectory, $\mathbf{slt}(t)$, and uniformly sample m segments for the original and generated trajectory, where the j th point for the i th segment are respectively denoted as $\mathbf{z}_{i,j}$ and $\mathbf{p}_{i,j}$ and the mean of an input segment is denoted by $\bar{\mathbf{z}}_i$. The trajectory SNR is then:

$$SNR = 10 \log_{10} \frac{\sum_i \sum_j (\mathbf{z}_{i,j} - \bar{\mathbf{z}}_i) \cdot (\mathbf{z}_{i,j} - \bar{\mathbf{z}}_i)}{\sum_i \sum_j (\mathbf{z}_{i,j} - \mathbf{p}_{i,j}) \cdot (\mathbf{z}_{i,j} - \mathbf{p}_{i,j})}, \quad (25)$$

which trivially generalises to the case of multiple disconnected trajectory segments, such as when the writer lifts-up their pen or brush.

We tested the iterative refinement on different inputs ranging from vector traces with no a priori kinematic information (Fig. 9), to online data — including the Graffiti Analysis database [F.A.T. Lab, 2009] (Fig. 10 and Fig. 11) and the UJI handwritten character dataset [Llorens *et al.*, 2008] (Fig. 7) — and it consistently produces visually accurate reconstructions of the input. We observe that, while fluctuations may appear during iterations, the refinement scheme consistently and rapidly converges towards a reduction of the error between the input and the generated trajectories and an increase in SNR (equation (25)).

The iterative scheme can be applied in a *batch* manner, in which all the $\Sigma\Lambda$ parameters for all strokes are updated at each iteration, or similarly to the iDe-Log framework [Ferrer *et al.*, 2018] by traversing the trajectory in an *incremental* manner and adjusting pairs of strokes ordered in time. In our experiments both approaches present similar convergence properties and produce reconstructions with similar SNR.

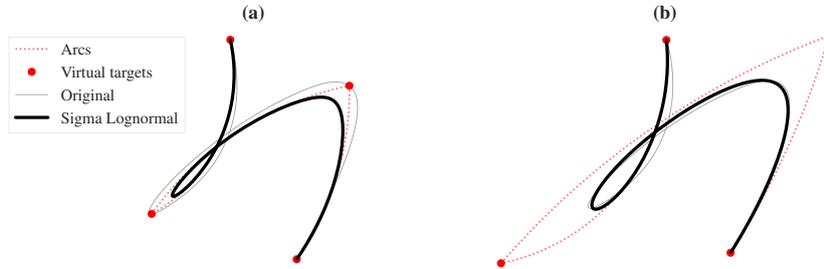


Fig. 9.: Reconstruction of vector input initially built with piecewise Bézier curves. Our method reconstructs the (originally only guaranteed to be C^0 continuous) input with smooth kinematics given by the $\Sigma\Lambda$ model. (a) First guess of the parameters from features. (b) Reconstruction of the input after iterative refinement steps.

6. Discussion and Applications

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 [Berio *et al.*, 2017b] and procedural content generation [Berio *et al.*, 2017a]. We list a few such applications in the remaining of this section.

6.1. Artificial Variability

The $\Sigma\Lambda$ model directly reflects the characteristics of a smooth human movement at the planning (targets and scaffold) and neuromotor level (the remaining parameters). We therefore expect and observe that parameter perturbations result in variations of a trace that are similar to the one that would be seen in multiple instances of handwriting or drawing made by one or more subjects (Fig. 12). This property has been exploited to create artificial data for handwriting recognizers [Fischer *et al.*, 2014], signature verifiers [Galbally *et al.*, 2012; Diaz-Cabrera *et al.*, 2018], gesture graphical input [Leiva *et al.*, 2017, 2016], and can be used for artistic oriented applications as well [Berio and Fol Leymarie, 2015; Berio *et al.*, 2017a]. For such a task, it is convenient to apply perturbations at the level of the intermediate parameters Δt_i and δi and (explicitly defined) target positions \mathbf{p}_i , which avoids the excessive distortions that we observe by directly perturbing relative angles and displacements as defined in the original formulation of the model [Plamondon *et al.*, 2014].

In our experiments, when perturbing targets, we have found that applying

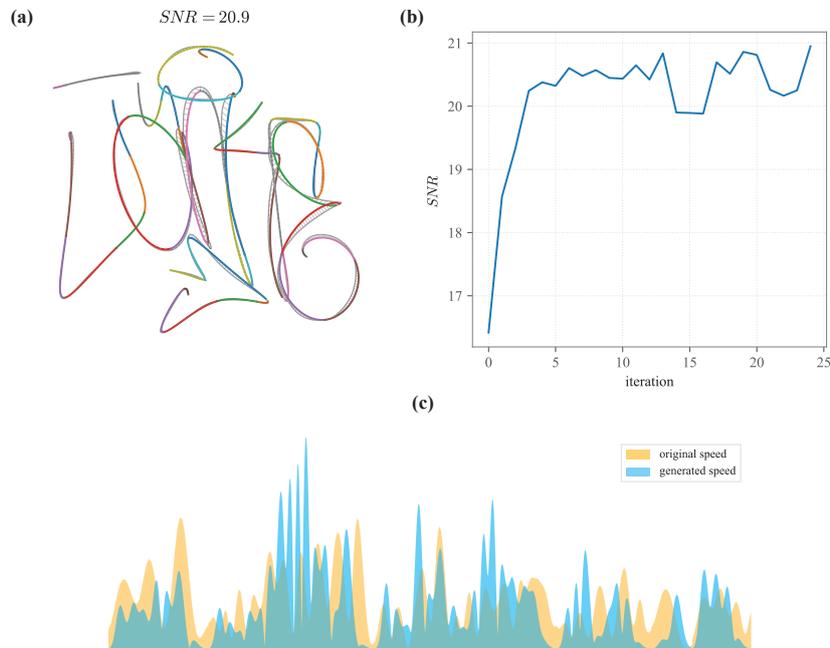


Fig. 10.: Reconstruction of a graffiti signature "JANKE" from the Graffiti Analysis database [F.A.T. Lab, 2009]. **(a)** The reconstructed trajectory, subdivided into segments for comparison (color coded), overlaid on the original trace (light grey). The short grey segments mark the errors and correspondences between uniformly distanced samples for each trajectory segment. **(b)** Plot of the SNR_t for each iteration of the iterative optimization scheme. **(c)** The speed profiles of the original (light grey) and reconstructed (dark grey) trajectories, scaled for comparison.

the perturbation with a variance inversely proportional to the temporal overlap parameters Δt_i (Fig. 12, top left) improves the legibility of the variations. This corresponds to imposing a higher precision requirement at trajectory locations with higher curvature, locations that are known to be the most informative of a trace/contour [Feldman and Singh, 2005]. From a motor control perspective, this effect is consistent with the "minimum intervention principle" [Todorov, 2004], suggesting that human movement variability is higher where it does not interfere with the performance required for a task.

6.2. Smoothing and Stylisation

Another application of the $\Sigma\Lambda$ parametrisation is to perform kinematic smoothing of a given trajectory. The result is a method that bears similarities to computer graphic approaches for curve fairing or neatening [Thiel *et al.*,

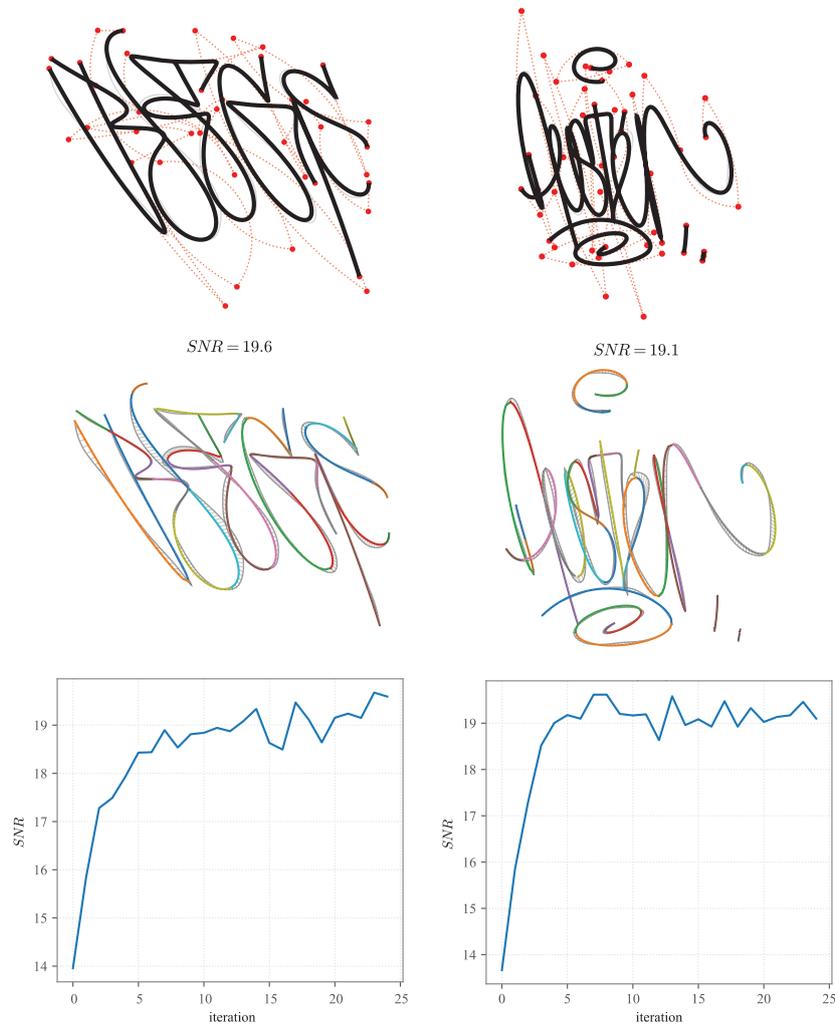


Fig. 11.: Additional examples of graffiti tag [FA.T. Lab, 2009] reconstruction together with the corresponding SNR plots.

2011; McCrae and Singh, 2009] as well as curve stylisation approaches [Lang and Alexa, 2015; Lu *et al.*, 2012]. In particular, the Euler spiral decomposition step of our reconstruction method is similar to a few previously proposed methods [Baran *et al.*, 2010; McCrae and Singh, 2009], which exploit the decomposition of an input curve into Euler spiral segments to remove discontinuities and

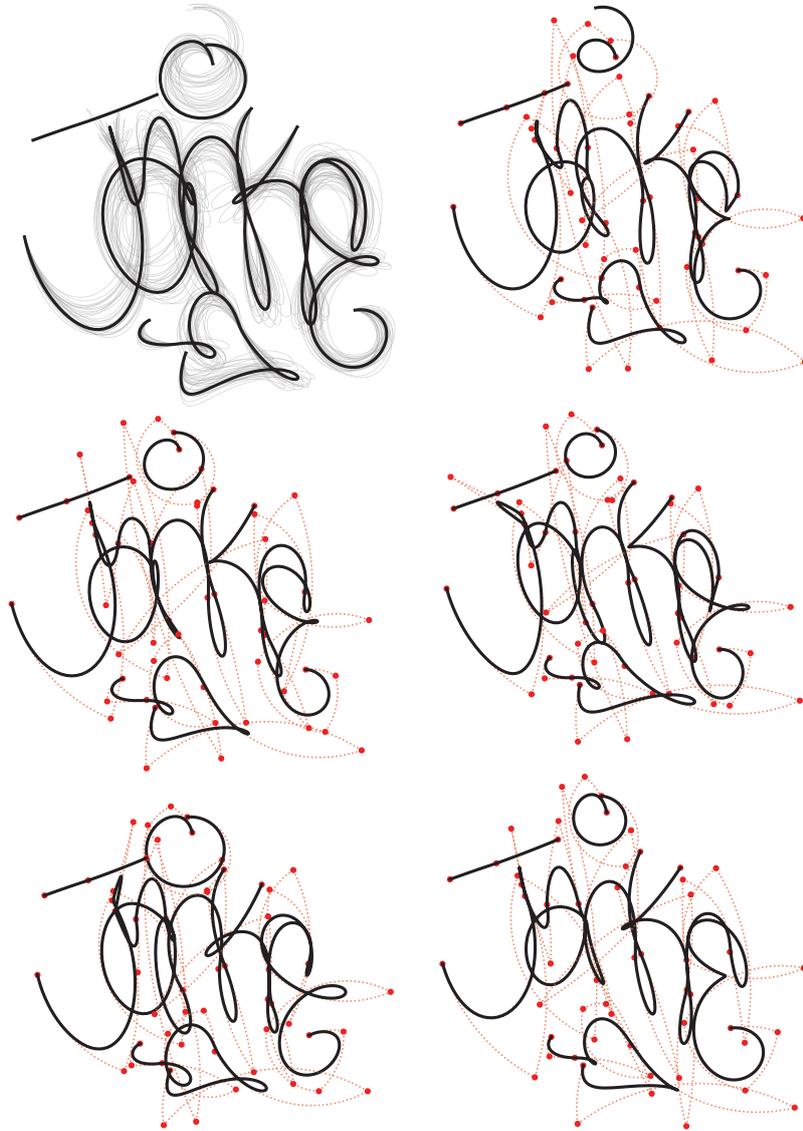


Fig. 12.: Parametric variations of a reconstructed graffiti instance from the Graffiti Analysis database. Top left, the original reconstruction (black trace) overlaid with 30 variations. Note that variability is higher in proximity of smooth segments of the trajectory. The remaining traces are randomly perturbed samples, with the corresponding (perturbed) action-plan in red.

guarantee second order (C^2) continuity in the output. In our case, we instead rely on the properties of the $\Sigma\Lambda$ model, which ensures the resulting reconstruction is smooth and infinitely differentiable (C^∞).

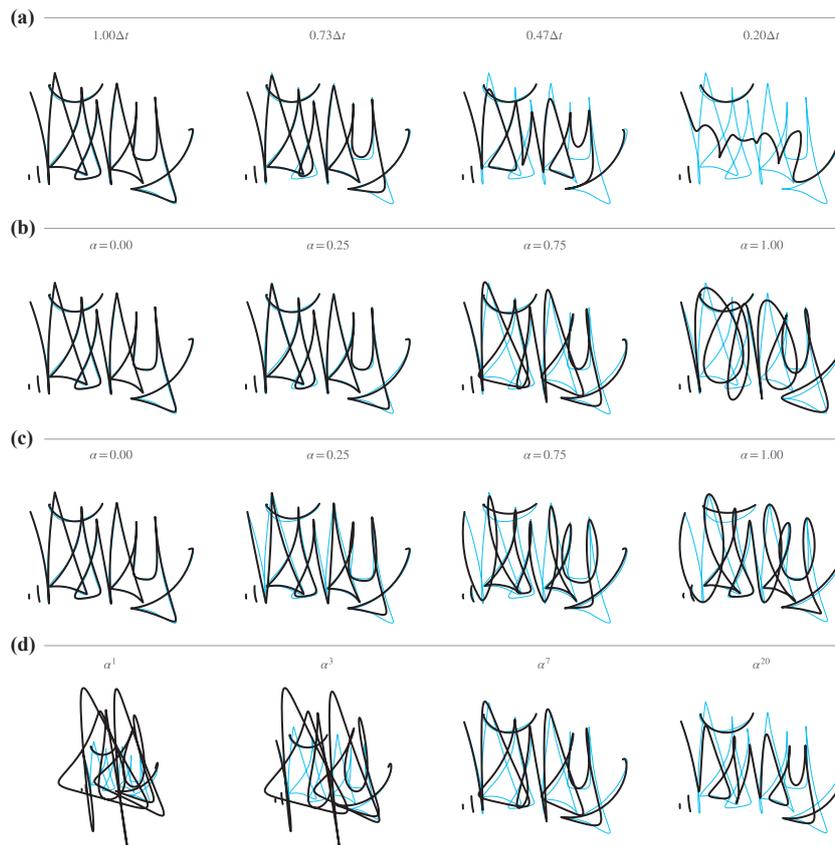


Fig. 13.: Comparison of smoothing and stylisation methods. Row (a), smoothing by global scaling of the time offset parameters Δt_i . Row (b), smoothing by using the parameter α to interpolate between the $\Sigma\Lambda$ parameters of two reconstructions with different values for Δt_+ . Row (c), stylisation effects achieved by interpolating between the $\Sigma\Lambda$ parameters of two reconstructions, where the second is performed with user specified values of δ_j . Row (d), effect of different powers of $\alpha = 0.75$ used to interpolate the virtual target positions between the two reconstructions used in row (b); The examples in row (b) and (c) use a power of 7.

Another method to smooth the generated trajectory is to simply scale the Δt_i parameters (Fig. 13.(a)). However, this quickly and coarsely simplifies the original trace — similarly to the smoothing effect of a convolution with a large

mask — and can even result in a loss of structure and legibility. One method to mitigate this undesirable effect is to run a second instance of the iterative refinement procedure with a lower value of $\Delta t_-, \Delta t_+$. As a result, we can achieve a smoothing effect while still preserving the structural similarity of the input, as provided by the original features. Variable degrees of smoothing can then be achieved by interpolating the $\Sigma\Lambda$ parameters between the original reconstruction and the smoothed one with a parameter $\alpha \in [0, 1]$ (Fig. 13.(b)).

More flexible stylisation effects can also be achieved with a similar approach, for example by constraining all stroke curvature parameters δ_i to a user-specified value (Fig. 13.(c)) and then running the iterative refinement with $\lambda_\delta = 0$, hence not further affecting the parameters. While we use linear interpolation for the parameters $\Delta t_i, \delta_i$, we observe that these are not linearly related to the target positions. While this relation deserves further analysis in future studies, we achieve satisfactory results by specifying a power of α for interpolating targets (Fig. 13.(d)) and observe experimentally that a power of 7 works particularly well for our use case (Fig. 13.(b), (c)).

6.3. Expressive Rendering and Interaction

The smooth kinematics produced by the $\Sigma\Lambda$ model can be exploited to generate expressive brush renderings of the trajectory. This has been explored in previous work to generate realistic renditions of signatures [Ferrer *et al.*, 2015] using a model of ink deposition model [Franke and Rose, 2004]. Here we demonstrate a brush model that builds upon the assumption that the amount of paint deposited is inversely proportional to the speed of the drawing tool. Furthermore, we can sweep a texture along the generated trajectory with width also inversely proportional to speed — refer to Berio *et al.* [2018] for mathematical and implementation details. While this is a simplistic model, it generates patterns that are highly evocative of some instances of calligraphy as well as graffiti made with markers or spray paint (Fig. 14), and accentuates the perceived dynamism of the trace.

The trajectory generated by the reconstruction, as well as the brush rendering parameters can be edited in real time with an intuitive user interface [Berio *et al.*, 2018]. This allows a user to fine tune the rendering results or to apply subsequent modifications to the trajectory by editing the target positions and the stroke parameters through a CAD-like interface. Furthermore, the resulting kinematics reproduce natural human-like movements that can be exploited to create stroke animations of the input as well as to generate smooth motion paths for virtual characters or even humanoid robots [Berio *et al.*, 2016].

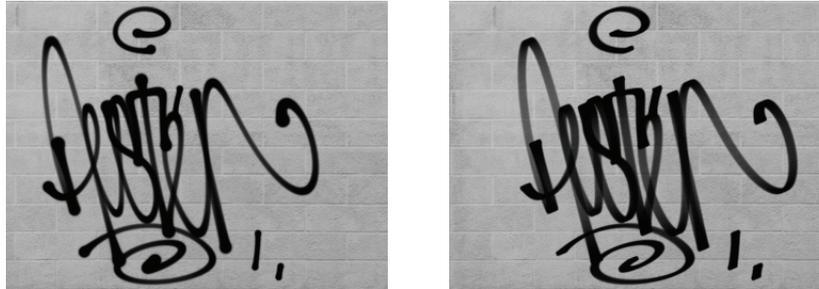


Fig. 14.: Examples of, on the left, a real graffiti textured over a wall and, on the right, one rendered with kinematics based brushes.

7. Conclusion

In this chapter, we developed and explained a systematic method to reconstruct $\Sigma\Lambda$ parameters from solely the static geometric trace (left by handwriting or drawing gestures), which relies on an initial segmentation at perceptually salient points. The method is capable of producing an accurate geometric reconstruction of the input, while inferring plausible kinematics underlying its generation on the basis of just an ordered sequence of points as an input.

We achieved our goal of a plausible reconstruction of the kinematics by designing a method exploiting a notion of Curvilinear Shape Features (or CSF) to incrementally adjust the temporal and spatial parameters of the $\Sigma\Lambda$ model. The method consistently produces accurate ($> 15\text{Db}$ SNR) reconstructions of the input, while providing flexibility for the use of additional constraints that can be exploited in order to generate interactive stylisations and variations.

One possible line of extension of our work is to exploit the availability of higher order derivatives of the $\Sigma\Lambda$ model to develop solutions using constrained optimisation methods with stronger convergence guarantees. Another interesting future line of work is to develop more sophisticated methods of fixing the $\Sigma\Lambda$ parameters μ_i and σ_i , which are currently experimentally set and kept constant. This shall require to explore in depth how the inferred kinematics relate to human data and are perceived by human observers.

References

- Attneave, F. (1954). Some informational aspects of visual perception. *Psychological review* **61**, 3, p. 183.
- Baran, I., Lehtinen, J. and Popović, J. (2010). Sketching clothoid splines using shortest paths, *Computer Graphics Forum* **29**, 2, pp. 655–664.

- Belyaev, A. and Yoshizawa, S. (2001). On evolute cusps and skeleton bifurcations, in *Proceedings of the International Conference on Shape Modeling and Applications (SMA)*, pp. 134–140, <https://doi.org/10.1109/SMA.2001.923384>.
- Berio, D., Akten, M., Fol Leymarie, F., Grierson, M. and Plamondon, R. (2017a). Calligraphic stylisation learning with a physiologically plausible model of movement and recurrent neural networks, in *ACM Proceedings of the 4th International Conference on Movement Computing (MOCO)*, London, UK, <https://doi.org/10.1145/3077981.3078049>.
- Berio, D., Calinon, S. and Fol Leymarie, F. (2016). Learning dynamic graffiti strokes with a compliant robot, in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Daejeon, South Korea, pp. 3981–86.
- Berio, D. and Fol Leymarie, F. (2015). Computational Models for the Analysis and Synthesis of Graffiti Tag Strokes, in P. Rosin, ed., *Computational Aesthetics*, Eurographics Association, pp. 35–47.
- Berio, D., Fol Leymarie, F. and Plamondon, R. (2017b). Computer aided design of handwriting trajectories with the kinematic theory of rapid human movements, in *Proceedings of the 18th International Graphonomics Society Conference (IGS)*, Gaeta, Italy, <https://graphonomics.net/igs2017/>.
- Berio, D., Leymarie, F. F. and Plamondon, R. (2018). Expressive Curve Editing with the Sigma Lognormal Model, in O. Diamanti and A. Vaxman, eds., *Proceedings of Eurographics — Short Papers*, Delft, The Netherlands, pp. 33–36.
- Blum, H. (1967). A transformation for extracting new descriptors of shape, in W. Wathen-Dunn, ed., *Models for the perception of speech and visual form*, pp. 362–380, MIT Press, proceedings of a symposium held in Boston, MA, November 1964.
- Blum, H. (1973). Biological shape and visual science (part I), *Journal of Theoretical Biology* **38**, 2, pp. 205–287.
- Brault, J.-J. and Plamondon, R. (1993a). A complexity measure of handwritten curves: Modeling of dynamic signature forgery, *IEEE Transactions on Systems, Man, and Cybernetics* **23**, 2, pp. 400–413.
- Brault, J.-J. and Plamondon, R. (1993b). Segmenting handwritten signatures at their perceptually important points, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **15**, 9, pp. 953–957.
- Bruce, J. W. and Giblin, P. J. (1992). *Curves and Singularities: A Geometrical Introduction to Singularity Theory*, 2nd edn., Cambridge University Press, doi:10.1017/CBO9781139172615.
- De Stefano, C., D'Elia, C., Garruto, M., Marcelli, A. and Di Freca, A. S. (2005). A wavelet based curve decomposition method for on-line handwriting, *Advances in Graphonomics: Proc. of IGS*, pp. 162–167 <https://graphonomics.net/publications.php>.
- De Winter, J. and Wagemans, J. (2006). Segmentation of object outlines into parts: A large-scale integrative study, *Cognition* **99**, 3, pp. 275 – 325, URL <http://www.sciencedirect.com/science/article/pii/S0010027705000648>.
- De Winter, J. and Wagemans, J. (2008). Perceptual saliency of points along the contour of everyday objects: A large-scale study, *Perception & Psychophysics* **70**, 1, pp. 50–64.
- Diaz-Cabrera, M., Fischer, A., Ferrer, M. A. and Plamondon, R. (2018). Dynamic signature verification system based on one real signature, *IEEE Transactions on Cybernetics*

- 48, 1, pp. 228–239, doi:10.1109/TCYB.2016.2630419.
- EA.T. Lab (2009). 000000book: An open database for graffiti markup language (gml) files, Web, URL <http://000000book.com>.
- Feldman, J. and Singh, M. (2005). Information along contours and object boundaries. *Psychological Review* **112**, 1, pp. 243–252.
- Ferrer, M. A., Diaz, M., Carmona, C. A. and Plamondon, R. (2018). iDe-Log: Iterative dual spatial and kinematic extraction of sigma-lognormal parameters, *IEEE Transactions on Pattern Analysis and Machine Intelligence* <https://doi.org/10.1109/TPAMI.2018.2879312>.
- Ferrer, M. A., Diaz-Cabrera, M. and Morales, A. (2015). Static signature synthesis: A neuromotor inspired approach for biometrics, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **37**, 3, pp. 667–680.
- Fischer, A., Plamondon, R., O'Reilly, C. and Savaria, Y. (2014). Neuromuscular representation and synthetic generation of handwritten whiteboard notes, in *Proceedings of the 14th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pp. 222–7, <https://doi.org/10.1109/ICFHR.2014.45>.
- Fong, W. C. (2003). Why Chinese painting is history, *The Art Bulletin* **85**, 2, pp. 258–280.
- Franke, K. and Rose, S. (2004). Ink-deposition model: The relation of writing and ink deposition processes, in *IEEE Proceedings of the Ninth International Workshop on Frontiers in Handwriting Recognition (IWFHR)*, Tokyo, Japan, pp. 173–178.
- Freedberg, D. and Gallese, V. (2007). Motion, emotion and empathy in esthetic experience, *Trends in cognitive sciences* **11**, 5, pp. 197–203.
- Freyd, J. J. (1983). Representing the dynamics of a static form. *Memory & cognition* **11**, 4, pp. 342–346.
- Galbally, J., Plamondon, R., Fierrez, J. and Ortega-Garcia, J. (2012). Synthetic on-line signature generation. Part I: Methodology and algorithms, *Pattern Recognition* **45**, 7, pp. 2610–21, doi:<https://doi.org/10.1016/j.patcog.2011.12.011>.
- Heald, M. (1985). Rational approximations for the Fresnel integrals, *Mathematics of Computation* **44**, 170, pp. 459–461.
- Lang, K. and Alexa, M. (2015). The Markov pen: Online synthesis of free-hand drawing styles, in *Proceedings of the Workshop on Non-Photorealistic Animation and Rendering (NPAR)*, Istanbul, Turkey, pp. 203–215.
- Leiva, L. A., Martín-Albo, D. and Plamondon, R. (2016). Gestures à Go Go: Authoring synthetic human-like stroke gestures using the kinematic theory of rapid movements, *ACM Transactions on Intelligent Systems and Technology (TIST) — Special Issue on Causal Discovery and Inference Archive* **7**, 2, pp. 15:1–15:29, doi:10.1145/2799648.
- Leiva, L. A., Martín-Albo, D. and Plamondon, R. (2017). The kinematic theory produces human-like stroke gestures, *Interacting with Computers* **29**, 4, pp. 552–565, doi:10.1093/iwc/iww039.
- Levien, R. (2009). *From Spiral to Spline: Optimal Techniques in Interactive Curve Design*, Ph.D. thesis, EECS Department, University of California, Berkeley.
- Leymarie, F. and Levine, M. D. (1989). Curvature morphology, in *Proceedings of Vision Interface '89*, CIPPRS, London, Ontario, Canada, pp. 102–109.
- Leymarie, F. and Levine, M. D. (1992). Simulating the grassfire transform using an active contour model, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **14**, 1, pp. 56–75, doi:10.1109/34.107013.

- Leyton, M. (1987). Symmetry-curvature duality, *Computer Vision, Graphics, and Image Processing* **38**, 3, pp. 327–341, [https://doi.org/10.1016/0734-189X\(87\)90117-4](https://doi.org/10.1016/0734-189X(87)90117-4).
- Li, X., Parizeau, M. and Plamondon, R. (1998). Segmentation and reconstruction of on-line handwritten scripts, *Pattern Recognition* **31**, 6, pp. 675–684.
- Li, X. and Plamondon, R. (2006). Medial axis extraction of planar shapes, *International Journal of Applied Mathematical Analysis and Applications* **1**, 1, pp. 11–27.
- Llorens, D. *et al.* (2008). The UJIPenchars database: A pen-based database of isolated handwritten characters. in *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC)*, pp. 2647–51, <http://www.lrec-conf.org/proceedings/lrec2008/>.
- Longcamp, M., Anton, J. L., Roth, M. and Velay, J. L. (2003). Visual Presentation of Single Letters Activates a Premotor area Involved in Writing, *NeuroImage* **19**, 4, pp. 1492–1500.
- Lu, J., Yu, F., Finkelstein, A. and DiVerdi, S. (2012). HelpingHand: Example-based stroke stylization, *ACM Transactions on Graphics (TOG)* **31**, 4, p. 46.
- McCrae, J. and Singh, K. (2009). Sketching piecewise clothoid curves, *Computers and Graphics* **33**, 4, pp. 452–461.
- Mediavilla, C. (1996). *Calligraphy: From Calligraphy to Abstract Painting*, Scirpus Publications, Wommelgem, Belgium.
- Ogniewicz, R. (1992). *Discrete Voronoi Skeletons*, Ph.D. thesis, ETH Zürich, <https://doi.org/10.3929/ethz-a-000692549>.
- O'Reilly, C. and Plamondon, R. (2008). Automatic extraction of Sigma-Lognormal parameters on signatures, in *Proc. of 11th International Workshop on Frontiers in Handwriting Recognition (ICFHR)*, Concordia University, Montreal, Canada.
- Pignocchi, A. (2010). How the Intentions of the Draftsman Shape Perception of a Drawing, *Consciousness and Cognition* **19**, 4, pp. 887–898.
- Plamondon, R. (1995). A Kinematic Theory of Rapid Human Movements. Part I. Movement Representation and Generation, *Biological Cybernetics* **72**, 4, pp. 295–307, <http://dx.doi.org/10.1007/BF00202785>.
- Plamondon, R. and Guerfali, W. (1998). The 2/3 power law: When and why? *Acta Psychologica* **100**, 1, pp. 85–96.
- Plamondon, R., O'Reilly, C., Galbally, J., Almaksour, A. and Anquetil, É. (2014). Recent developments in the study of rapid human movements with the kinematic theory: Applications to handwriting and signature synthesis, *Pattern Recognition Letters* **35**, 1, pp. 225–235, <https://doi.org/10.1016/j.patrec.2012.06.004>.
- Plamondon, R. and Privitera, C. M. (1999). The segmentation of cursive handwriting: An approach based on off-line recovery of the motor-temporal information, *IEEE Transactions on Image Processing* **8**, 1, pp. 80–91, <https://doi.org/10.1109/83.736691>.
- Richards, W. and Hoffman, D. (1985). Codon constraints on closed 2D shapes, *Computer Vision, Graphics, and Image Processing* **31**, 3, pp. 265–281.
- Thiel, Y., Singh, K. and Balakrishnan, R. (2011). Elasticurves: Exploiting stroke dynamics and inertia for the real-time neatening of sketched 2D curves, in *Proceedings of the 24th annual ACM symposium on User Interface Software and Technology (UIST)*, Santa Barbara, CA, USA, pp. 383–392, <https://doi.org/10.1145/2047196.2047246>.
- Todorov, E. (2004). Optimality principles in sensorimotor control, *Nature Neuroscience*

- 7, 9, pp. 907–915, <https://doi.org/10.1038/nm1309>.
- Viviani, P. and Schneider, R. (1991). A developmental study of the relationship between geometry and kinematics in drawing movements, *Journal of Experimental Psychology: Human Perception and Performance* **17**, 1, pp. 198–218.
- Witkin, A. (1983). Scale-space filtering, in *Proceedings of the Eighth International Joint Conference on Artificial Intelligence (IJCAI)*, Vol. 2, Karlsruhe, West Germany, pp. 1019–22.
- Zitnick, C. L. (2013). Handwriting beautification using token means, *ACM Transactions on Graphics (TOG)* **32**, 4, p. 53, <https://doi.org/10.1145/2461912.2461985>.