Expressive Curve Editing with the Sigma Lognormal Model

Daniel Berio[†1] , Frederic Fol Leymarie and Réjean Plamondon

[1]Goldsmiths University of London, [2]Polytechnique Montréal

Figure 1: *Sigma Lognormal trajectory, the corresponding action plan (in red) and renderings with different kinematic dependent brushes.*

Abstract

We describe a practical application of the Sigma Lognormal model of handwriting movements for computer graphics applications that require the interactive or procedural definition of artistic or calligraphic traces. The method allows to easily edit curves with physiologically plausible kinematics that can be exploited in order to generate expressive brush renderings, natural looking stroke animations and easily generate stylistic variations of a trace.

CCS Concepts

•***Computing methodologies*** $\rightarrow$ ***Parametric curve and surface models;*** *Non-photorealistic rendering; Physical simulation;*

1. Introduction

Many computer graphics applications and interfaces are aimed at mimicking the effect of hand drawn strokes and curves. In computer aided design (CAD) applications, the two leading approaches for this task are either (i) user freehand input through a digitiser device such as a trackpad, mouse or tablet or (ii) the interactive definition of the control points of parametric curves, with the most ubiquitous representation being quadratic or cubic Bézier splines. In the former method, the input is often smoothed or neatened/faired [MS09, Zit13, TSB11] in order to reduce digitisation artefacts or possible hesitations caused by ergonomic limitations of the input device with respect to a physical drawing medium. The output is then commonly converted to a parametric curve representation in order to achieve resolution independence and to allow corrections by the adjustment of control points. However, both the placement and the adjustment of control points can be counter intuitive, especially when the desired effect is to mimic the trace and curving behaviour that would be produced by the skilled motions of a well trained artist. Mimicking such visual qualities often requires placing control points at locations that influence the curve geometry but do not reflect any perceptually salient feature along its trace [YSW*17].

In our current research, we explore a "movement centric" approach to curve generation in which we describe a curve through a smooth motion underlying its production rather than through an explicit a priori definition of its geometry. In this paper, we focus on the application of a physiologically plausible model of handwriting [LMAP17], which describes the kinematics of arbitrarily complex pen movements through the superimposition of a sequence of target directed sub-movements. Rather than defining a drawing/writing trace with a freehand input, the user defines the spatial evolution of the trajectory through the specification of a *motor plan*, which consists of a sparse sequence of target loci (Fig. 1, left). This representation is similar to the *control polygon* typically used for spline based interfaces and can be easily edited with an interactive user interface (Sec. 4). The targets are positioned near salient points along the generated trace [DWW08, YSW*17] and directly reflect an action plan [MPS13] that we propose is advantageous for editing traces that mimic the visual qualities of a skillful artistic gesture, such as the ones observed in traditional calligraphy or modern graffiti.

The output of the method is a smooth trajectory that reflects the kinematics of a movement with properties that are very simi-

† corresponding author: d.berio@gold.ac.uk

lar to the ones that are observed in human handwriting and drawing movements. While in previous works we explored the application of this method for mimicking calligraphic traces [BL15], in this paper we focus on a complete description of an updated and computationally efficient formulation of the trajectory formation model (Sec. 3). The smooth kinematics produced by the method can be exploited to generate natural looking stroke animations (please refer to the associated submitted video), expressive brush renderings of the trajectory (Sec. 5) and the proposed representation naturally leads a method that allows the generation of stylistic variations of a given trace (Sec. 6).

2. Background overview

Our method produces movements that are consistent with a number of principles that have been observed in the rich history of studies of human movement and in particular handwriting. The velocity profile of rapid and straight reaching motions can be described by a variably asymmetric "bell shaped" speed profile [Mor81] More complex human movements are smooth [FH85] and can further be represented as the composition of a discrete number of sub-movements, which are often referred to as "strokes" [Mor81, FH05]; where each such stroke is also described with the characteristic bell-shaped speed profile. This results in an inverse proportional relation between speed and absolute curvature of the trajectory [LTV83] also observed in human movements.

The simulation of a movement has been previously used to generate sketch based renderings of 3D models [HS07], pencil drawings [AWI*09] or calligraphic like traces [Hae90]. Our methodology differs in that we define a user interface (UI) that uses a control polygon similar to the one used in spline based methods. This allows a precise manipulation of the trajectory with a method that is familiar to the typical users of CAD applications. Yan et al. [YSW*17] have developed an interface for editing Bézier curves in which key points correspond with curvature extrema along the generated curve. In our work, we also follow a similar rationale that curvature extrema are intuitive locations for the definition of a curve, but, in our case this property of the interface follows from the intrinsic (inverse) relation between speed and curvature of human-like movements. However, in our method the trajectory does not strictly interpolate the key points, but the proximity of a curvature extrema to a key point varies depending on the trajectory smoothness in the corresponding region. Also relevant to our approach is the work by Saito et al. [SKCN08] who vary brush thickness based on curvature. In our method, we are able to achieve a similar effect by directly exploiting the kinematics of the generated trajectory.

3. Trajectory generation

In principle, many different models from the fields of computational motor control, robotics and graphonomics can be exploited to generate curves through the kinematics of a human like movements. Here we focus on the application of the Sigma Lognormal ($\Sigma\Lambda$) Model [P*14], one model of handwriting that is part of the "Kinematic Theory of Rapid Human Movements" [Pla95] , which has been shown to give remarkably accurate reconstruction of the kinematics of human reaching and handwriting movements. The $\Sigma\Lambda$ model describes complex handwriting trajectories via the vectorial superimposition of N time shifted target, or strokes, each described by a time shifted lognormal function (Fig. 2, right):

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

which represents the impulse response to a (centrally generated) command occurring at time t_{0i}. The parameters μ_i and σ_i 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 (Fig. 2).

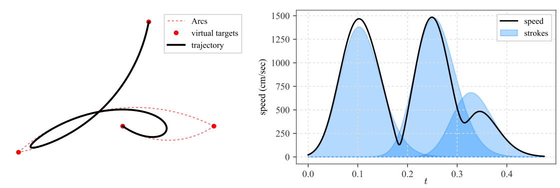


Figure 2: $\Sigma\Lambda$ *trajectory (left) with the corresponding action plan and the lognormal components (right).*

3.1. Weighted Signal-Lognormal model

While the original formulation of the model [LMAP17] describes the trajectory evolution in terms of pen tip velocity with superimposition of ballistic strokes, for the sake of interactive CAD applications, we propose a *weighted* parameterisation of the model, which allows to efficiently compute the pen tip position at a given time t in parametric form. The weight of each stroke is given by:

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

which avoids the necessity of numerically integrating the velocity and provides a performance advantage, since the error function (erf) is readily available in most programming languages and numerical packages. The planar evolution of a trajectory with N strokes can be described with an initial position $\boldsymbol{p}_0$ followed by a sequence of targets $(\boldsymbol{p}_1,\ldots,\boldsymbol{p}_m)$. Each stroke evolves between a pair of consecutive targets $\boldsymbol{p}_{i-1},\boldsymbol{p}_i$ and is parametrised with an amplitude D_i and a direction θ_i given by the norm and orientation of the vector $\boldsymbol{p}_i-\boldsymbol{p}_{i-1}$. The curvilinear evolution of a stroke is given by a circular arc with central angle δ_i and can be computed with

$$\boldsymbol{d}_i(t) = D_i\begin{bmatrix}(\cos(\theta_{0i}-\delta_i w_i(t))-\cos(\theta_{0i}))(2\sin(\delta_i/2))^{-1}\\ (\sin(\theta_{0i}-\delta_i w_i(t))-\sin(\theta_{0i}))(2\sin(\delta_i/2))^{-1}\end{bmatrix} \tag{3}$$

where $\theta_{0i}=\theta_i+(\pi+\delta_i)/2$ and where in eq. 3 we take care of setting δ_i to a small non-zero value for straight lines in order to avoid numerical precision issues or divisions by zero when the arc internal angle is too small. The position along the trajectory is then computed using a vectorial sum of each stroke with:

$$\boldsymbol{p}(t) = \boldsymbol{p}_0 + \sum_{i=1}^{N}\boldsymbol{d}_i(t). \tag{4}$$

In order to simplify the definition of the timing parameters for interactive applications, we keep the parameters μ_i, σ_i fixed to a user defined values, the variation of which determines the overall duration and skewedness of the lognormal. 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 $t_{01} = 0$. The parameter Δt_i then intuitively determines the smoothness of the trajectory similarly to weights in NURBS curves; smaller values increase the lognormal overlap and consequently produce a smoother trajectory in the vicinity of the virtual target (Fig. 2).

4. User interaction

The previously defined trajectory model, together with the virtual target and circular arc representation can easily be edited interactively. We implement a simple UI in which the user is able to drag $N+1$ target positions (including the initial point $\boldsymbol{p}_0$), such that each target point, with the exception of the first, is paired with a handle with direction indicating the internal angle δ_i and length specifying the time overlap parameter Δt_i (Fig. 3). The user can also click to create a new target point, resulting in a new initially straight stroke with default values of $\delta_i = 0$ and $\Delta t_i = 0.5$ and average overlap between successive lognormals.

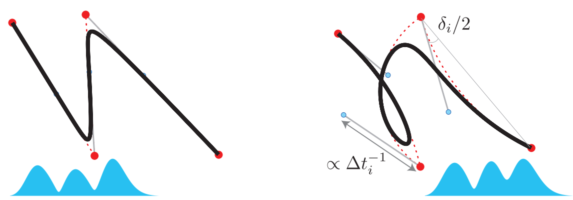


Figure 3: *Example UI for editing $\Sigma\Lambda$ trajectories, with speed profiles for each trajectory shown in cyan below. Left: defaults when user adds new targets. Right: trajectory after some manipulations. The length of the handles (defined with a blue dot) is inversely proportional to the value of Δt_i and the angle of the handle with respect to the vector between two consecutive targets is $= \delta_i/2$.*

The resulting UI is very similar to the ones used in traditional methods such as Bézier curves. However, our proposed method facilitates the dynamic production of curves used in art forms such as calligraphy or graffiti. Our targets are located in proximity of curvature extrema along the generated trajectory, which are known to be highly informative [Att54] and perceptually salient [DWW08], and prove good candidates for the interactive definition of curves [LS09, YSW*17]. At the same time the user is effectively editing a plan for an intended motion with a representation that reflects the concatenation of a series of simple reaching/aiming movements. Such a target mapping is consistent with the hypothesis of an effector independent representation or "motor plan" of a movement in the human nervous system [FDCM15].

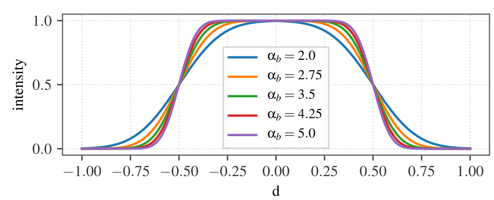
5. Expressive brush rendering

One of the advantages of generating curves through the simulation of a movement, is that the smooth kinematics can be exploited to drive the implementation of expressive rendering methods. For example, in prior work the same type of handwriting model has been exploited to generate realistic renditions of signature pen traces [FDCM15] using an ink deposition model [FR04]. Here we try to achieve an effect evocative of instances of ink-calligraphy and graffiti made with markers or spray paint with a simple yet flexible brush model that allows to mimic different types of drawing media. Our method builds upon the assumption that the amount of paint deposited is inversely proportional to the speed of the drawing tool. While this is not intended as an accurate model of a brush or pen, it produces qualitatively convincing patterns and provides a simple way to generate traces that accentuate the perceived dynamism of a trace.

We describe a variably smooth brush texture using again the error function to generate a variably smooth "hat" curve:

$$\frac{1}{2} + \frac{1}{2}\mathrm{erf}\left[\alpha\,(1-d)\right] \tag{5}$$

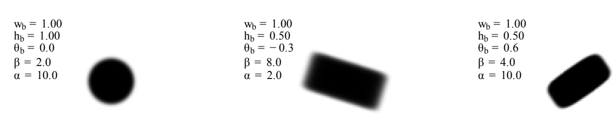
where the parameter α determines the top hat flatness of the curve.



We then generate a variably sized and rotated brush texture with normalised coordinates $(u, v) \in [0,1]$ by using the distance to a superellipse with

$$d = \sqrt{\left|\frac{u\cos\theta_b - v\sin\theta_b}{w_b}\right|^{\beta} + \left|\frac{u\sin\theta_b + v\cos\theta_b}{h_b}\right|^{\beta}} \tag{6}$$

where θ_b determines the brush rotation and w_b, h_b respectively determine the relative width and height:



We further use the traditional "dabbing" procedure to sweep the brush along the trajectory and scale the brush using an inverse function of the trajectory speed:

$$r(t) = r_{min} + (r_{max} - r_{min})\exp\left(-\frac{\bar{v} + |\dot{\boldsymbol{p}}(t)|}{\bar{v}}\right) \tag{7}$$

scaled by the mean $\bar{v}$ of the speed for the whole trajectory (Fig. 4).

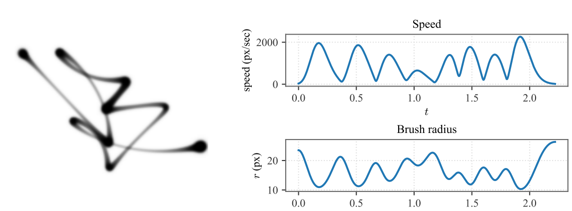


Figure 4: *Dabbing a brush with variable width (bottom right) depending on the speed (top right).*

The brush size is varied between a range $[r_{min}, r_{max}]$, which allows to adjust the amount of speed dependent scaling in the generated image. The speed $|\dot{\boldsymbol{p}}(t)|$ can be computed exactly by using the original form of the $\Sigma\Lambda$ model [P*14] or also easily computed by forward differencing the trajectory $\boldsymbol{p}(t)$, which is faster and sufficiently accurate for this application.

6. Kinematic variations of style

The proposed curve representation leads to a definition of a subset of the possible stylisations of a letter form, in which different "hand styles" are produced by kinematic variations of a movement that follows a common structure. The structural component is given by the sequence of targets (Fig. 5, top-left) and different stylistic variations are easily generated by varying the parameters of the $\Sigma\Lambda$ model. Because such parameters directly reflect the kinematic features of a natural movement, their perturbation produces variations in the trace that are consistent with the ones that would be produced from real human movements.

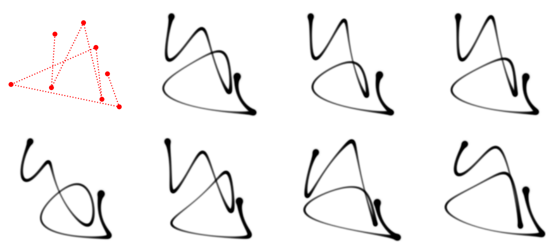


Figure 5: *Target structure of a letter "a" (top left) and kinematic variations of its trace generated by perturbing $\Sigma\Lambda$ parameters.*

Such variability is not a by-product of a set of instances, computed afterward, but is rather intrinsically built in the abstract representation of a pattern. As a result, it is possible to easily achieve effects ranging from the subtle variations such as the ones seen in multiple writing/drawing instance by the same author (Fig. 5, top row), to more drastic effects that mimic different stylisations of a letter form (Fig. 5, bottom row).

7. Conclusion

We have presented an interactive method for curve generation aimed at applications that mimic the visual and kinematic qualities of calligraphy and gesture-based drawing. We propose that a movement centric approach to curve generation is a useful and intuitive tool for this type of applications and provides a powerful new tool in addition to existing curve generation methods. Future avenues of work could include improving the user interface based on additional feedback from users and exploring efficient methods that convert the $\Sigma\Lambda$ representation to the more ubiquitous piecewise spline models for greater interoperability with existing software.

Acknowledgement

This work has been partly supported by UK's EPSRC Centre for Doctoral Training in Intelligent Games and Game Intelligence (IGGI; grant EP/L015846/1).

References

[Att54] ATTNEAVE F.: Some informational aspects of visual perception. *Psychological review 61*, 3 (1954), 183. 3

[AWI*09] ALMERAJ Z., WYVILL B., ISENBERG T., GOOCH A. A., GUY R.: Automatically mimicking unique hand-drawn pencil lines. *Computers & Graphics 33*, 4 (2009), 496–508. 2

[BL15] BERIO D., LEYMARIE F. F.: Computational Models for the Analysis and Synthesis of Graffiti Tag Strokes. In *Computational Aesthetics* (2015), Rosin P., (Ed.), Eurographics Association, pp. 35–47. 2

[DWW08] DE WINTER J., WAGEMANS J.: Perceptual saliency of points along the contour of everyday objects: A large-scale study. *Perception & Psychophysics 70*, 1 (2008), 50–64. 1, 3

[FDCM15] FERRER M. A., DIAZ-CABRERA M., MORALES A.: Static signature synthesis: A neuromotor inspired approach for biometrics. *IEEE Transactions on Pattern Analysis and Machine Intelligence 37*, 3 (2015), 667–680. 3

[FH85] FLASH T., HOGAN N.: The coordination of arm movements. *Journal of Neuroscience 5*, 7 (1985), 1688–1703. 2

[FH05] FLASH T., HOCHNER B.: Motor primitives in vertebrates and invertebrates. *Current opinion in neurobiology 15*, 6 (2005), 660–6. 2

[FR04] FRANKE K., ROSE S.: Ink-deposition model: The relation of writing and ink deposition processes. In *Frontiers in Handwriting Recognition, 2004. IWFHR-9 2004. Ninth International Workshop on* (2004), IEEE, pp. 173–178. 3

[Hae90] HAEBERLI P.: Paint by numbers: Abstract image representations. In *ACM SIGGRAPH Computer Graphics* (1990), vol. 24, ACM, pp. 207–214. 2

[HS07] HOUSE D. H., SINGH M.: Line drawing as a dynamic process. In *Computer Graphics and Applications, 2007. PG'07. 15th Pacific Conference on* (2007), IEEE, pp. 351–360. 2

[LMAP17] LEIVA L. A., MARTÍN-ALBO D., PLAMONDON R.: The kinematic theory produces human-like stroke gestures. *Interacting with Computers* (2017), 1–14. 1, 2

[LS09] LEVIEN R., SÉQUIN C. H.: Interpolating splines: Which is the fairest of them all? *Computer-Aided Design and Applications 6*, 1 (2009), 91–102. 3

[LTV83] LACQUANITI F., TERZUOLO C., VIVIANI P.: The law relating the kinematic and figural aspects of drawing movements. *Acta psychologica 54*, 1 (1983), 115–130. 2

[Mor81] MORASSO P.: Spatial control of arm movements. *Experimental Brain Research 42*, 2 (1981), 223–7. 2

[MPS13] MARCELLI A., PARZIALE A., SENATORE R.: Some observations on handwriting from a motor learning perspective. In *AFHA* (2013), pp. 6–10. 1

[MS09] MCCRAE J., SINGH K.: Sketching piecewise clothoid curves. *Computers and Graphics (Pergamon) 33*, 4 (2009), 452–461. 1

[P*14] PLAMONDON R., ET AL.: Recent developments in the study of rapid human movements with the kinematic theory. *Pattern Recognition Letters 35* (2014), 225–35. 2, 4

[Pla95] PLAMONDON R.: A Kinematic Theory of Rapid Human Movements. Part I . Movement Representation and Generation. *Biological cybernetics 72*, 4 (1995), 295–307. 2

[SKCN08] SAITO S., KANI A., CHANG Y., NAKAJIMA M.: Curvature-based stroke rendering. *The Visual Computer 24*, 1 (2008), 1–11. 2

[TSB11] THIEL Y., SINGH K., BALAKRISHNAN R.: Elasticurves: Exploiting stroke dynamics and inertia for the real-time neatening of sketched 2d curves. In *Proc. of the 24th annual ACM symposium on User interface software and technology* (2011), ACM, pp. 383–392. 1

[YSW*17] YAN Z., SCHILLER S., WILENSKY G., CARR N., SCHAEFER S.: k-curves: interpolation at local maximum curvature. *ACM Transactions on Graphics (TOG) 36*, 4 (2017), 129. 1, 2, 3

[Zit13] ZITNICK C. L.: Handwriting beautification using token means. *ACM Transactions on Graphics (TOG) 32*, 4 (2013), 53. 1