EUROGRAPHICS Workshop on Sketch-Based Interfaces and Modeling (2008)
C. Alvarado and M.- P. Cani (Editors)

Sketching Piecewise Clothoid Curves

J. McCrae and K. Singh

Dynamic Graphics Project, University of Toronto, Canada

Abstract

We present a novel approach to sketching 2D curves with minimally varying curvature as piecewise clothoids. A stable and efficient algorithm fits a sketched piecewise linear curve using a number of clothoid segments with G^2 continuity based on a specified error tolerance. Further, adjacent clothoid segments can be locally blended to result in a G^3 curve with curvature that predominantly varies linearly with arc length. We also handle intended sharp corners or G^1 discontinuities, as independent rotations of clothoid pieces. Our formulation is ideally suited to conceptual design applications where aesthetic fairness of the sketched curve takes precedence over the precise interpolation of geometric constraints. We show the effectiveness of our results within a system for sketch-based road and robot-vehicle path design, where clothoids are already widely used.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Line and Curve Generation

1. Introduction

Curves are ubiquitous in Computer Graphics, as primitives to construct shape or define shape features, as strokes for sketch-based interaction and rendering or as paths for navigation and animation. Motivated originally by curve and surface design for engineering applications, complex shapes are typically represented in a piecewise manner, by smoothly joining primitive shapes (see Figure 1). Traditionally, research on curve primitives has focused on parametric polynomial representations defined using a set of geometric constraints, such as Bezier or NURBS curves [Far90]. Such curves have a compact, analytically smooth representation and possess many attractive properties for curve and surface design. Increased computing power, however, has made less efficient curve primitives like the clothoid a feasible alternative for interactive design. Dense piecewise linear representations of continuous curves have also become increasingly popular. Desirable geometric properties, however, are not intrinsically captured by these polylines but need to be imposed by the curve creation and editing techniques used [GBS03, TBSR04, CS04].

An important curve design property is *fairness* [FRSW87, qSzL89, MS92], which attempts to capture the visual aesthetic of a curve. Fairness is closely related to how little and how smoothly a curve bends [MS92] and for planar curves, described as curvature continuous curves with a small number of segments of almost piecewise linear curvature [FRSW87].

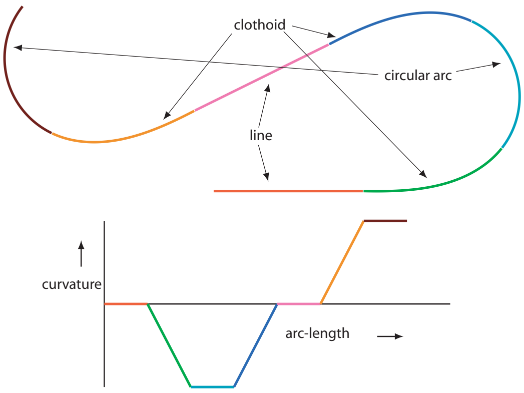

Figure 1: *A curve composed of clothoids, line and circular-arc segments.*

The family of curves whose curvature varies linearly with arc-length were described by Euler in 1774 in connection with a coiled spring held taut horizontally with a weight at its extremity. Studied in various contexts in science and engineering, such a curve is also referred to as an Euler spiral, Cornu spiral, linarc, lince or clothoid (see Figure 2).

Clothoids are especially useful in transportation engineering, since they can be navigated at constant speed by linear steering and a constant rate of angular acceleration. Rollercoasters are frequently composed of sequences of clothoid loops. While intrinsic geometric splines like clothoids were introduced in computer aided design in 1972 [NMK72] and subsequently developed as transition curves for road design [MW92, WM05], they have had little recent exposure in mainstream Computer Graphics. In this paper, we exploit the fairness properties of clothoids to fit 2D strokes for sketch-based applications.

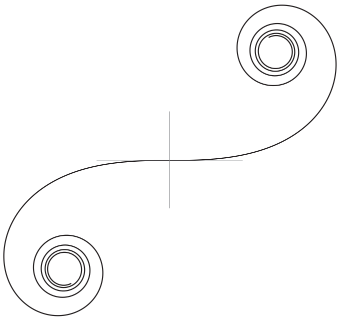

Figure 2: *Clothoid: a curve whose curvature varies linearly with arc-length, also known as an Euler spiral, Cornu spiral or linarc. The above clothoid has a curvature range* $[-1.15, 1.15]$ *and arc-length* 100 *(or* $t \in [-5.362, 5.362], B = 3.72$*).*

1.1. Problem statement

Polyline stroke data often needs to be denoised and processed into fair 2D curves for further use in many sketch-based applications. This is usually done using smoothing filters [TBSR04] or by cubic or high-order spline fitting [Pav83, Pra87]. Iterative smoothing is best suited to removing high-frequency sketching noise and tends to produce low-frequency wiggles in the curve (local pockets of smooth curvature based on filter size). Spline fitting results, though visually smooth, frequently exhibit poor quality curvature plots (see Figure 3). We present a new approach to processing sketch strokes using clothoids, that intrinsically favour line and circular arc segments and result in holistically fair G^2 curves.

1.2. Overview of our approach

Our algorithm for fitting a sequence of G^2 clothoid segments to polyline stroke data is a two-step process (see Figure 4). We first fit a piecewise linear approximation to the discrete curvature of the stroke as a function of arc-length, with control over the tradeoff between fitting error and the number of linear pieces. The start and end curvature values of each linear piece uniquely determine a line, circular arc or clothoid curve segment. These segments further assemble together uniquely with G^2 continuity into a single composite curve. The next step involves determining a single $2D$ rigid transform that aligns this composite curve with the sketched stroke to minimize the error of the stroke from the transformed curve. We are able to solve for this transform efficiently by formulating the error as a weighted least squares optimization problem. While many sketch-based applications do not require precise interpolation of points and tangents, we show how this can be achieved by inserting or appending short spline segments to enforce interpolation (see Figure 12), if necessary. The resulting curve can also be made G^3 by linearly blending the adjacent clothoid segments locally (see Figure 11). Alternatively, sharp corners can be allowed by thresholding spikes in curvature to be segment boundaries and independently rotating these segments (see Figure 10).

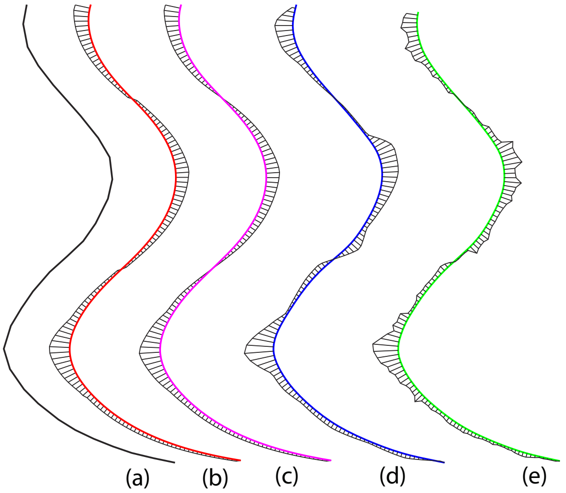

Figure 3: *Stroke fairing: (a) A sketched stroke. (b) Clothoid fitting the stroke (a). (c) Cubic spline fitting the clothoid curves in (b). (d) Cubic spline fitting the stroke (a). (e) Laplacian smoothing (4 iterations at 10%) the stroke (a). Curvatures are plotted uncolored along the length of processed strokes (b-d) to evaluate smoothness.*

1.3. Contributions

We develop a new formulation for efficiently fitting intrinsic spline primitives such as clothoids, to dense polyline data. While we focus on clothoids our algorithmic framework is applicable to any curve primitive with a characteristic curvature profile. The resulting curves are robust to sketching noise and are particularly well suited to sketch-based applications. We show a number of enhancements to the basic approach, including sharp corners, blended G^3 curves and point interpolation. Finally, we have implemented our results within a sketch-based application for track design (see Figure 5), where the clothoid segments provide not only aesthetically pleasing curves but are also required downstream, from an engineering standpoint.

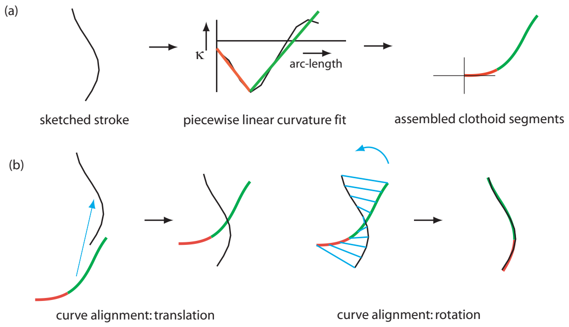
(a)
κ
arc-length
sketched stroke
piecewise linear curvature fit
assembled clothoid segments
(b)
curve alignment: translation
curve alignment: rotation

Figure 4: *Clothoid fitting: (a) Discrete stroke curvature is approximated as a piecewise linear function uniquely defining clothoid segments. (b) A rigid 2D transform minimizes the weighted least squares error between the composite clothoid and the sketched stroke.*

2. Related work

We now survey prior work specifically relating to curve and surface fairing in general and on clothoids in particular. A popular feature of cubic splines is that they provide a linear approximation to the minimum strain energy configuration of a thin-beam interpolating a set of points. While least squares spline fitting is robust and efficient [Pra87], the curvature plot of the resulting spline can be highly variable (see Figure 3). Computing the actual minimum energy curve minimizes the overall bending of the curve [Mel74]. Moreton and Sequin [MS92] show, however, that minimum variation curves provide a better fairness characteristic by minimizing the overall variation of curvature along the curve allowing natural shapes like circular arcs. These curves are typically computed by nonlinear optimization techniques. In contrast, we attempt to minimize overall variation in curvature along the curve by robustly approximating it using a number of piecewise linear segments. Our composite clothoid curve is thus an appealing alternative to minimum variation curves, particularly when precise interpolation of points is traded for precise curvature control.

A more common, easy to implement, approach is to iteratively smooth the points of piecewise linear curves and surfaces directly [TBSR04]. Discrete filtering approaches vary from simple neighbour averaging to approaches that use a discrete curvature estimation to help guide the fairing process [MR07]. We similarly compute a discrete curvature estimate at points of the input polyline, but instead use these values to determine the segmentation of the curve into clothoid pieces. An additional advantage of fitting analytic curve segments like splines or clothoids over discrete methods is that the curve can be regenerated at arbitrary resolution.

Clothoids have been the subject of prior research in computer aided design. Motivated by transportation design, Meek and Walton have looked at conditions under which one or more clothoid segments can form a transition curve between two given curve segments [MW92]. They have also proposed a clothoid spline [WM05], where two clothoid pieces are used to form a parabola-like segment between every three consecutive points of a control polygon. While the resulting clothoid spline is G^2, the curve is forced through a point of zero curvature on every edge of the control polygon. A discrete formulation of clothoid using nonlinear subdivision has also been proposed [GXH01]. Clothoids have also been used as a transition curve segment for computer vision applications of occluded contour completion and inpainting [KFP03].

Originally motivated by a system for quickly sketching track layouts for game environments and road layout conceptualization by landscape architects, we find clothoids to be attractive curve primitives that qualitatively capture the natural curvature variations of human sketching well.

3. Clothoid Terminology

The clothoid spiral can be parameterized using the Fresnel integrals

$$C(t) = \int_0^t \cos \frac{\pi}{2} u^2 du, \tag{1}$$

$$S(t) = \int_0^t \sin \frac{\pi}{2} u^2 du, \tag{2}$$

as

$$\pi B \begin{pmatrix} C(t) \\ S(t) \end{pmatrix}, \tag{3}$$

where t is the arc length parameter, and πB is a positive scaling parameter that defines the slope of linear curvature variation of a family of spirals as seen in Figure 6.

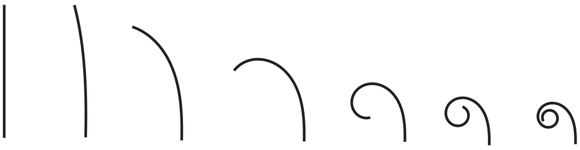

Figure 6: *Fixing arc length and an initial curvature parameter, a family of clothoid segments is formed by decreasing parameter B near infinity (left) toward zero (right).*

Clothoids can be expressed in a computationally efficient manner, using rational approximations for $C(t)$ and $S(t)$ given in [Hea85]:

$$C(t) \approx \frac{1}{2} - R(t) \sin \left(\frac{1}{2} \pi (A(t) - t^2) \right), \tag{4}$$

$$S(t) \approx \frac{1}{2} - R(t) \cos \left(\frac{1}{2} \pi (A(t) - t^2) \right), \tag{5}$$

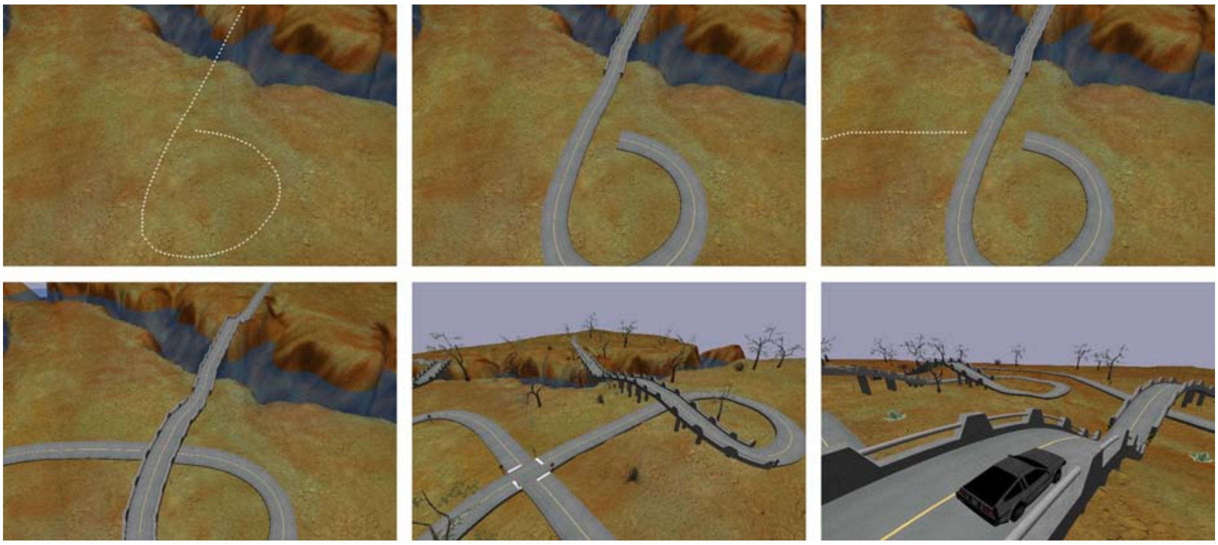

Figure 5: *Sketching clothoid splines within* Drive, *a sketch-based track modeling system.*

where

$$R(t) = \frac{0.506t+1}{1.79t^2+2.054t+\sqrt{2}},$$
$$A(t) = \frac{1}{0.803t^3+1.886t^2+2.524t+2}.$$

The maximum error of this approximation is 1.7×10^{-3}. Higher order approximations are also defined, with maximum error up to 4×10^{-8}, but we found the above sufficient.

4. Curve fitting using clothoids

We now detail our approach to curve fitting using a sequence of clothoid, circular arc and line segments (see Figure 1,13). Note that while the steps below fit a polyline, they can be used to fit any curve representation that is discretely sampled at an appropriate resolution.

4.1. Discrete curvature estimation

Discrete curvature for planar curves can be estimated at a point using the circum-circle formed with its two adjacent points or the Frenet-Serret fomulae as shown in [MR07]. Given any three sequential points p_{i-1}, p_i, p_{i+1} of the input polyline, using the vectors $v_1 = p_i - p_{i-1}$, $v_2 = p_{i+1} - p_i$, the estimated curvature at p_i is given by

$$\kappa(p_i) = \frac{2\sin\left(\frac{\theta}{2}\right)}{\sqrt{||v_1|| \cdot ||v_2||}}, \tag{6}$$

where $\theta = \arccos\left(\frac{v_1}{||v_1||} \cdot \frac{v_2}{||v_2||}\right)$. Robust statistical approaches to curvature computation that perform better in the presence of noise and irregular sampling [KSNS07] can also be used. The curvature for discretely sampled analytic curves may also be directly sampled from the analytic curve.

Each point is now mapped into curvature space, where the horizontal axis denotes arc length and the vertical axis, curvature (see Figure 4a). We adopt (positive/negative) curvature to denote (right/left) turning in this space.

4.2. Piecewise linear curvature segmentation

We now segment the curve into a minimal sequence of pieces of linearly varying curvature. A dynamic programming algorithm finds a connected set of line segments which minimize both the number of line segments used, and the error in fit with the curvature space points. The number of pieces used is minimized by assigning a penalty E_{cost} for each linear piece. We populate a matrix M with values, in a bottom-up fashion, using the following:

$$M(a,b) = \min_{a<k<b} \left\{ M(a,k)+M(k,b), E_{fit}(a,b)+E_{cost} \right\}. \tag{7}$$

$M(a,b)$ denotes the minimal cost of a configuration of connected line segments from point a to b. $M(a,b)$ entries are calculated for all $a<b$, making M strictly upper triangular. $E_{fit}(a,b)$ denotes the vertical error resulting from linear regression with the points from a to b. Expressing the linear regression line using slope and y-intercept, denoting them l_{slope} and l_{yint} respectively, we can define E_{fit} precisely as

$$E_{fit}(a,b) = \sum_{i=a}^{b} \left| l_{yint} + l_{slope} \cdot arclength(p_i) - \kappa(p_i) \right|, \tag{8}$$

where $(arclength(p_i), \kappa(p_i))$ is the curvature-space point corresponding to p_i.

The solution, a set of connected line segments in curvature space, defines the set of connected clothoid segments that will be used to fit the input curve. Figure 7 shows the effect that different values of E_{cost} has on the generated solution. In practice, we use values of E_{cost} ranging from 0.01 to 0.1.

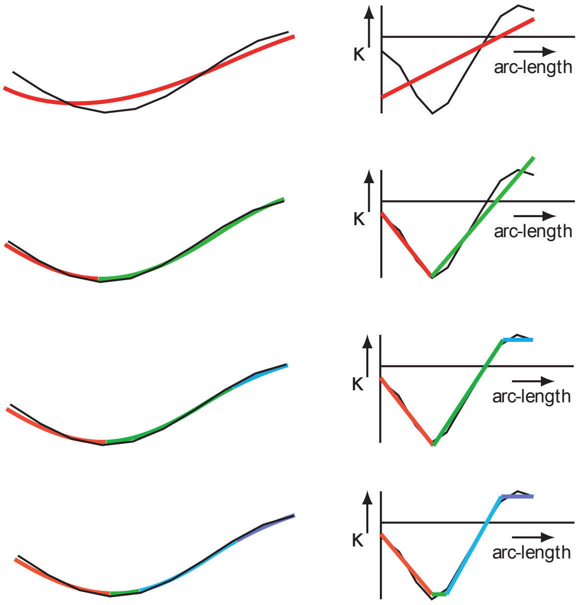

Figure 7: *The effect of E_{cost} on the generated segmentation. As E_{cost} decreases, more segments are used.*

4.3. Segment parameterization

For each clothoid segment, we have its curvature space end-points (x_i^P, y_i^P) and (x_{i+1}^P, y_{i+1}^P). y_i^P and y_{i+1}^P specify the start and end curvatures of the segment, and the difference $x_{i+1}^P - x_i^P$ specifies the arc length. These parameters uniquely map to a clothoid segment defined by the scaling parameter B, and the start and end parameter values t_1 and t_2. Since the curvature of a clothoid is $\frac{t}{B}$:

$$t_1 = y_i^P B \text{ and } t_2 = y_{i+1}^P B. \tag{9}$$

B can be expressed using the formula for arc length:

$$\begin{aligned} x_{i+1}^P - x_i^P &= \pi B(t_2 - t_1) \\ &= \pi B(y_{i+1}^P B - y_i^P B) \quad \text{(using (9))} \\ &= B^2 \pi (y_{i+1}^P - y_i^P) \\ \frac{x_{i+1}^P - x_i^P}{\pi(y_{i+1}^P - y_i^P)} &= B^2 \end{aligned}$$

and since B must be positive,

$$B = \sqrt{\frac{x_{i+1}^P - x_i^P}{\pi(y_{i+1}^P - y_i^P)}}. \tag{10}$$

Each clothoid segment is translated and rotated to connect end points and align tangents to adjacent segments resulting in an overall G^2 curve (see Figure 4a).

4.4. 2D Rigid Transformation

We now need to translate and rotate this overall curve, so as to minimize the fitting error to the input curve. We cast this as a weighted least squares minimization problem as follows: Sample a corresponding set of n points from the canonical clothoid spline, using the arc length positions from the input polyline. Define the set of corresponding n canonical points with an S superscript: $\left\{(x_0^S, y_0^S), \ldots, (x_{n-1}^S, y_{n-1}^S)\right\}$. Figure 8 shows the clothoid spline in its canonical form, with the corresponding set of n points sampled along it.

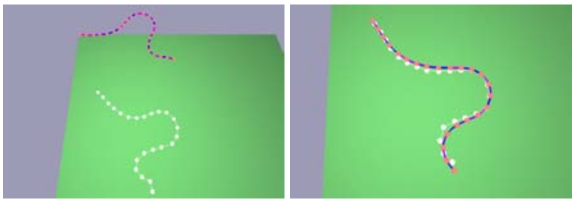

Figure 8: *The points $\left\{(x_0^S, y_0^S), \ldots, (x_{n-1}^S, y_{n-1}^S)\right\}$ on the composite curve in pink must undergo a rigid 2D transformation to match the sketched input curve in white (left). The result of the transformation (right).*

The goal is to minimize the sum of 2-norm distances between corresponding pairs of points with a rotation matrix R and translation vectors T and T^S:

$$\sum_{i=0}^{n-1} \left\| R\left(\begin{pmatrix} x_i^S \\ y_i^S \end{pmatrix} + T^S \right) + T - \begin{pmatrix} x_i \\ y_i \end{pmatrix} \right\|_2 . \tag{11}$$

Our approach is based on the solution for shape matching shown in [MHTG05]. The optimal translation vector is given by aligning the weighted centroids of both sets of points:

$$T^S = \frac{1}{\sum_{i=0}^{n-1} w_i} \begin{pmatrix} \sum_{i=0}^{n-1} w_i x_i^S \\ \sum_{i=0}^{n-1} w_i y_i^S \end{pmatrix}, \tag{12}$$

$$T = \frac{1}{\sum_{i=0}^{n-1} w_i} \begin{pmatrix} \sum_{i=0}^{n-1} w_i x_i \\ \sum_{i=0}^{n-1} w_i y_i \end{pmatrix}, \tag{13}$$

where each weight w_i specifies the relative importance of the corresponding pair of points $(x_i, y_i), (x_i^S, y_i^S)$ in the fit (see Figure 4b,9).

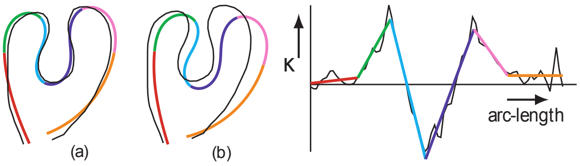

Figure 9: *Equal weights for all sample points (a), and only weighting the end-points (b), result in different rigid transforms for the same composite curve segmentation on the right.*

Define sets of points which are the relative locations to the centroids $q_i = (x_i^S, y_i^S) - T^S$ and $p_i = (x_i, y_i) - T$. To determine the rotation matrix, the problem is relaxed to finding the optimal linear transformation A, where we want to minimize $\sum_{i=0}^{n-1} w_i(Aq_i - p_i)^2$. Setting the derivatives with respect to all coefficients of A to zero yields the optimal transformation

$$A = (\sum_{i=0}^{n-1} w_i p_i q_i^T)(\sum_{i=0}^{n-1} w_i q_i q_i^T)^{-1} = A_{pq}A_{qq}. \tag{14}$$

A_{qq} can be ignored as it is symmetric and does scaling only. The optimal rotation R is then the rotational part of A_{pq}, found by a polar decomposition $A_{pq} = RS$, where $S = \sqrt{A_{pq}^T A_{pq}}$, and so $R = A_{pq}S^{-1}$.

If the matrix $A_{pq}^T A_{pq}$ is near-singular, instead the vector from the start to end point of the sketched curve given by $(x_{n-1}, y_{n-1}) - (x_0, y_0)$ is used, and its arctangent provides an estimate of the best angle of rotation.

5. Fitting extensions

5.1. Sharp corners (G^1 discontinuity)

Many sketching applications require the user to only sketch smooth strokes and handle corners by requiring two separate smooth strokes to end at a corner. Such a restriction adds a cognitive burden on the user and can be disruptive to the sketching process. To automatically handle sharp corners in our framework, we first need to detect points of G^1 discontinuity in the sketched stroke. Observe that such sharp corners appear as large spikes in curvature space (see Figure 10). Statistical approaches to curvature estimation [KSNS07] are able to robustly filter out similar spikes that may arise from noise and outliers in the sketched stroke. Simple thresholding of points with both high curvature and high variation in curvature yields our set of sharp corners. We then force a segment break at all sharp corners and flatten the curvature spike from the set of curvature points so as not to bias the subsequent fitting process. The final segmentation is a further refinement of the segments induced by the sharp corners. We now treat the composite curve as having limbs that articulate at the corners. We fit this curve by finding the optimal transformation for the first limb as in Section 4.4. The translation of each subsequent limb is now constrained but its optimal rotation may once again be solved as in Section 4.4. We use a higher weight for the corner points in this fitting to better match the user sketched corners. While a more globally optimal set of transformations may be sought, we find this greedy approach to work well in practice and the resulting curves closely match the input sketch.

5.2. G^3 continuity

It is also easy to extend the given piecewise construction to produce G^3 continuous curves. Following the curvature space linear segmentation step, between each pair of segments, we can round the corners in curvature space by performing a local blend (see Figure 11). For each segmentation point (x_i^P, y_i^P) that is the endpoint of two segments, blending occurs within a window of distance d around x_i^P. A set of blended samples can be constructed for this window, sampling with a value s such that $0 < s < 1$, each sample point is given by

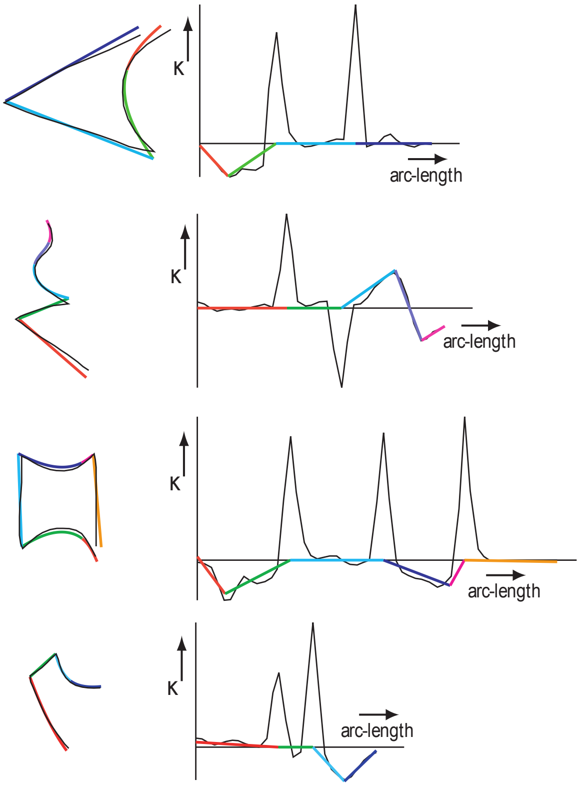

Figure 10: *Curves with sharp corners or G^1 discontinuities are automatically handled by our fitting approach.*

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x_i^P + d(2s-1) \\ y_i^P - m_1 d(s^2 + 2s - 1) + m_2 ds^2 \end{pmatrix} \tag{15}$$

where $m_1 = \frac{y_i^P - y_{i-1}^P}{x_i^P - x_{i-1}^P}$ and $m_2 = \frac{y_{i+1}^P - y_i^P}{x_{i+1}^P - x_i^P}$ are the slopes of the curvature space line segments.

Segmentation point (x_i^P, y_i^P) is then replaced by the generated set of blended samples. The samples in this region finitely approximate a quadratic function with a continuous derivative.

5.3. Geometric interpolation

While our approach is tailored towards constructing fair curves that approximate sketch strokes, it may be desirable to interpolate given geometric constraints. Performing such

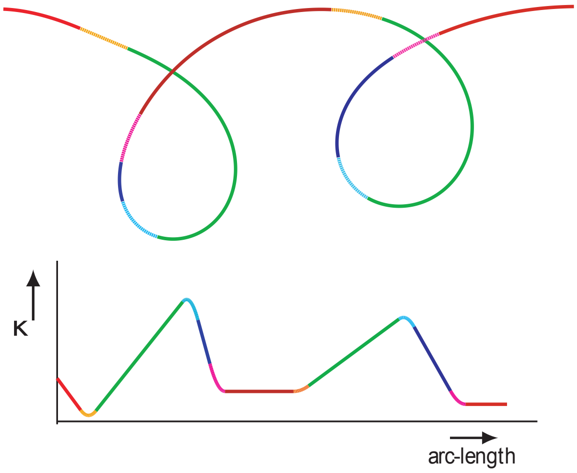

Figure 11: *A G^3 continuous curve obtained by local linear blending of adjacent clothoid segments.*

interpolation strictly using clothoids is sometimes impossible [MW92]. Instead within our system we simply use quintic Hermite splines that we locally blend into the curve generated by Section 4 to interpolate arbitrary points with G^2 continuity (see Figure 12). We note that while G^2, the use of Hermite splines can destroy the fairness properties of the overall curve.

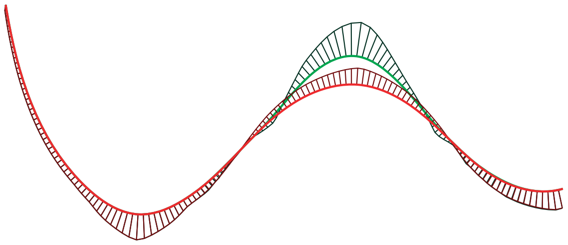

Figure 12: *The curve composed of clothoid segments (red) in Figure 3 is edited in the middle using a quintic spline (green) with G^2 continuity but with degradation in quality of the curvature plot.*

6. Sketching Applications

We have implemented our approach both as a simple sketching interface capable of generating a wide variety of aesthetic curves (see Figure 13) and as part of *Drive*, a comprehensive system for sketch-based road network design (see Figure 5). While *Drive* has a number of sophisticated features specific to the conceptual sketching of a driving experience, it is built around a simple interface for sketching clothoid curves. The framework naturally favours lines, circular arcs and clothoids which are common in road design and also desirable from a steering standpoint. In our system users can prescribe a preference for more or less segments by directly specifying E_{cost}, or by specifying an error of fit, in which case the system iteratively uses a lower E_{cost}, if the error of fit is above the given tolerance (see Figure 7). Users can also oversketch parts of curves as one might expect, in which case the track is globally refitted or blended in locally using a spline as in Section 5.3 (see Figure 14).

Our demo application is implemented in C++ using OpenGL and GLUT. It was tested on 2 systems: an AMD Athlon64 3000+ 2GHz and an Intel Xeon 2.2GHz, both with 1 GB RAM, and in both cases curves consisting of hundreds of points are generated in real time. The most computationally costly step in our approach is determining the curvature space segmentation. As a dynamic programming algorithm is used to find a global minimum solution, the number of points of the input polyline determine the number of rows and columns of the cost matrix M, leading to quadratic growth in the number computations required.

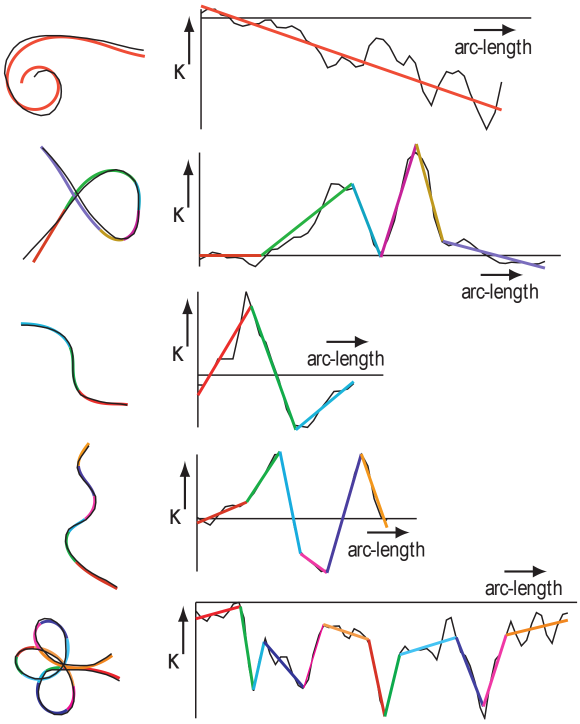

Figure 13: *Gallery of curves sketched using our system (left) with corresponding curvature profiles (right).*

7. Conclusion

We have presented an approach to fitting sketched strokes with a sequence of line, circular-arc and clothoid segments. We empirically find that clothoids tend to capture sketched strokes well and usually only a few (less than five) clothoid segments can capture a stroke with screen resolution fidelity. Figure 3 shows our fitting approach to be an appealing alternative to current approaches to stroke fairing, such as Laplacian smoothing or cubic spline fitting, particularly when a

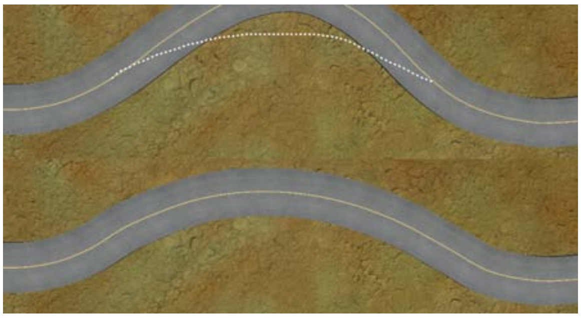

Figure 14: *Oversketching to edit curves.*

good approximation is more desirable than precise interpolation of any given point. If cubic splines are necessary for downstream use, we find that fitting the clothoid curves provides better fairness than directly fitting the input stroke. We also demonstrate our approach to work effectively within a road design system. Designers often work with characteristic shape palettes defined by French curves [Sin99], or predefined pieces. In the future we hope to explore the use of intrinsic splines such as clothoids for both palette representation and shape editing.

Acknowledgements

We thank MITACS for their research support, and Ryan Schmidt and Patricio Simari for their video production assistance.

References

[CS04] Coleman P., Singh K.: Cords: keyframe control of curves with physical properties. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Sketches* (2004), p. 60. 1

[Far90] Farin G.: *Curves and surfaces for computer aided geometric design*. Academic Press Professional, Inc., San Diego, CA, USA, 1990. 1

[FRSW87] Farin G., Rein G., Sapidis N., Worsey A. J.: Fairing cubic b-spline curves. *Comput. Aided Geom. Des. 4*, 1-2 (1987), 91–103. 1

[GBS03] Grossman T., Balakrishnan R., Singh K.: An interface for creating and manipulating curves using a high degree-of-freedom curve input device. In *CHI '03: Proceedings of the SIGCHI conference on Human factors in computing systems* (New York, NY, USA, 2003), ACM, pp. 185–192. 1

[GXH01] Guiqing L., Xianmin L., Hua L.: 3d discrete clothoid splines. In *CGI '01: Proceedings of the International Conf. on Computer Graphics* (Washington, DC, USA, 2001), IEEE Computer Society, p. 321. 3

[Hea85] Heald M. A.: Rational approximations for the fresnel integrals. *Mathematics of Computation 44*, 170 (1985), 459–461. 3

[KFP03] Kimia B. B., Frankel I., Popescu A.-M.: Euler spiral for shape completion. *Int. J. Comput. Vision 54*, 1-3 (2003), 157–180. 3

[KSNS07] Kalogerakis E., Simari P., Nowrouzezahrai D., Singh K.: Robust statistical estimation of curvature on discretized surfaces. In *SGP '07: Proceedings of the fifth Eurographics symposium on Geometry processing* (Aire-la-Ville, Switzerland, Switzerland, 2007), Eurographics Association, pp. 13–22. 4, 6

[Mel74] Melhum E.: Nonlinear splines. *Computer Aided Geometric Design* (1974), 173–207. 3

[MHTG05] Müller M., Heidelberger B., Teschner M., Gross M.: Meshless deformations based on shape matching. *ACM Trans. Graph. 24*, 3 (2005), 471–478. 5

[MR07] Mullineux G., Robinson S. T.: Fairing point sets using curvature. *Comput. Aided Des. 39*, 1 (2007), 27–34. 3, 4

[MS92] Moreton H. P., Séquin C. H.: Functional optimization for fair surface design. *SIGGRAPH Comput. Graph. 26*, 2 (1992), 167–176. 1, 3

[MW92] Meek D., Walton D.: Clothoid spline transition spirals. *Mathematics of Computation 59*, 199 (July 1992), 117–133. 2, 3, 7

[NMK72] Nutbourne A. W., McLellan P. M., Kensit R. M. L.: Curvature profiles for plane curves. *Comput. Aided Des.* (1972), 176–184. 2

[Pav83] Pavlidis T.: Curve fitting with conic splines. *ACM Trans. Graph. 2*, 1 (1983), 1–31. 2

[Pra87] Pratt V.: Direct least-squares fitting of algebraic surfaces. In *SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1987), ACM, pp. 145–152. 2, 3

[qSzL89] qing Su B., zhe Liu D.: *Computational geometry: curve and surface modeling*. Academic Press Professional, Inc., San Diego, CA, USA, 1989. 1

[Sin99] Singh K.: Interactive curve design using digital french curves. In *I3D '99: Proceedings of the 1999 symposium on Interactive 3D graphics* (New York, NY, USA, 1999), ACM, pp. 23–30. 8

[TBSR04] Tsang S., Balakrishnan R., Singh K., Ranjan A.: A suggestive interface for image guided 3d sketching. In *CHI '04: Proceedings of the SIGCHI conference on Human factors in computing systems* (New York, NY, USA, 2004), ACM, pp. 591–598. 1, 2, 3

[WM05] Walton D., Meek D.: A controlled clothoid spline. *Computers & Graphics 29* (2005), 353–363. 2, 3