GEOMETRIC ALGORITHMS FOR CURVE-FITTING

James Fagan and Alan Saalfeld
Statistical Research Division
Bureau of the Census
Washington, DC 20233

ABSTRACT

Some curve-fitting procedures may be described entirely in geometric terms, and, thus, may be shown to depend exclusively on the geometry (relative positions) of the points to which the curve is fitted. These procedures necessarily exhibit desirable invariance properties under large families of transformations such as the family of all affine transformations because the underlying geometry of the fit points behaves predictably under the families of transformations. Although the procedures are defined geometrically in order to make proofs of invariance straightforward, nevertheless, the algebraic representations of the fitted curves may be drived analytically. Conversely, whenever one can show that a curve-fitting procedure has an underlying geometric definition, then one may take advantage of geometric invariance properties.

GEOMETRY AND TRANSFORMATIONS

Geometry, which refers to the intrinsic relations among points, lines, curves, and areas, may be examined from the point of view of transformations of the plane. Geometric properties may be described and even defined in terms of those transformations. A classic illustration of such a property is the fundamental notion of **congruence** in classical Euclidean geometry. Two geometric figures are said to be **congruent** if one can be moved onto the other; or, perhaps more accurately, if the whole plane can be moved with a **rigid motion** or isometry so that one figure is brought into perfect alignment with the other. Euclidean geometry is concerned with those properties which are preserved by rigid motions of the plane (i.e., rotations, translations, reflections, and combinations of the three). Congruence involves nothing more than belonging to the same equivalence class of figures under all rigid motions. The statements of the Euclidean theorems, axioms, and propositions are given in terms of properties and descriptors which are left unaffected by rigid motions: those properties include **being parallel, being perpendicular, bisecting angles or line segments**, and **lying on a straight line** or **lying at a fixed distance from some other object.** For example, if line **A** is perpendicular to line B, and if a rigid motion is applied to the space, transforming line **A** into line **A'** and line B into line B', respectively; then line **A'** will be perpendicular to line **B'**.

Euclidean geometry in the plane is simply the study of those properties which do not change when rigid motions are applied. Classical spherical geometry deals with invariant properties under rigid motions of the sphere. Hyperbolic geometry, another non-Euclidean geometry, does not involve rigid motions in the usual sense. Nevertheless, it does involve the study of properties which remain invariant under a family of transformations; and by a kind of duality, one could define a kind of albeit unnatural "rigidness" in terms of those transformations.

The family of rigid motions is not the only family of transformations one might wish to apply to the plane. Other families give rise to other geometries. One might add the family of contractions and dilations (uniform shrinking and expanding maps) to create a geometry of similar figures (same shape but different sizes). In this expanded geometry, two similar figures are equivalent. In this geometry of similar figures, absolute distances are lost; only relative distances have meaning. Different families of transformations preserve different properties or relations of lines, points, curves, and areas. By recognizing the properties preserved by a specific family of transformations, one may use those properties to define **geometrically stable entities**, entities which themselves do not vary under the transformation family. A curve-fitting procedure can be such an entity.

CURVE-FITTING PROCEDURES

A curve-fitting procedure will be understood to mean an **exact** curve-fitting procedure throughout. An exact curve-fitting procedure has as its input an ordered finite sequence of **n** points ($p_1, p_2, p_3, \ldots, p_n$) and returns a continuous one-dimensional curve which passes through the **n** points ($p_1, p_2, p_3, \ldots, p_n$) in order. Exact curve-fitting procedures are used in cartography, for example, to reconstruct smooth representations of irregular linear features from shape files containing only coordinate values for critical or extreme points. Considerable storage can be saved if only a few points on a river feature are kept on the file and the river is drawn from those points using a curve-fitting algorithm.

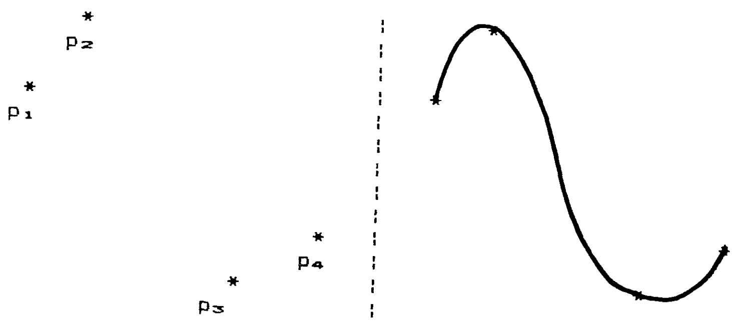

Figure 1. Four Fit Points and Cubic Polynomial Fit.

The above figure illustrates the result of fitting a cubic polynomial curve: $Y = AX^3 + BX^2 + CX + D$, to the four fit points p_1, p_2, p_3, **and** p_4.

The cubic fit is not a very good curve for a number of reasons. In the first place, a cubic polynomial will not fit every possible sequence of four points in the plane. Because the value of **Y** is expressed as a function of **X**, there can be only one **Y** value for each **X** value. The curve can only move from right to left or from left to right, but not both ways; there can be no doubling back. Nevertheless, if the points p_1, p_2, p_3, **and** p_4 have their **X** coordinates strictly increasing or strictly decreasing when the points are expressed in their sequential order, then (and only then!) can a cubic curve be fitted to the four points.

PROCEDURES WHICH COMMUTE WITH TRANSFORMATIONS

Given a sequence of points, a curve-fitting rule or procedure, and a transformation of the plane, one may perform two different composite operations:

First, one may apply the curve-fitting rule to the original points to produce a curve in the plane; then one may transform the plane to create a transformed image of the curve, which will be a new curve.

Alternatively, one may transform the original points by looking at their images under the transformation of the plane; then apply the curve-fitting rule to the transformed points.

If the resulting curves of the two composite procedures are the same, then the procedure is said to **commute** with the transformation.

In some cases the procedure will not only fail to commute with the transformation, it may even fail to be applicable to the transformed sequence of points. A simple example of such a complete lack of commutativity is the cubic polynomial fit rule or procedure with a 90^{o} rotation of the plane for the transformation applied to the four points given in the example on the previous page:

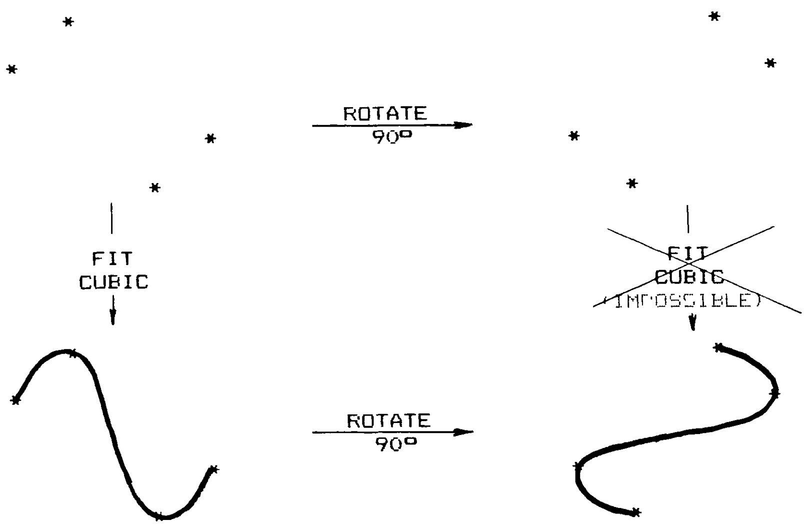

Figure 2. Transformation and Procedure do not Commute.

The cubic polynomial fit procedure will commute with some transformations of the plane. Reflections in the **X**-axis or in the **Y**-axis will commute with fitting a cubic to an acceptable set of four points. Indeed, reflections in any vertical or horizontal line will commute with the cubic fit procedure. Uniform (i.e. linear) stretching or shrinking along either axis is a transformation which will commute with the cubic polynomial fit procedure. Rotations in general will not commute, although a rotation of 180^{o} will. The collection of all transformations which commute with a curve-fitting procedure for every suitable sequence of fit points will form a **group** of transformations. The larger the group of transformations, the better the curve-fitting procedure. A good curve-fitting procedure has for its group of transformations a group which contains all of the rigid motions. A good curve-fitting procedure also works for an arbitrary sequence of points in the plane. If this is the case, then congruent sequences of fit points will result in congruent fitted curves, no matter where in the plane the fit points are positioned (as a congruency class).

A curve-fitting procedure which can be applied to any sequence of points in the plane and which commutes with all rigid motions will be called **Euclidean** or **geometric.** Euclidean curve-fitting procedures are coordinate-free in the sense that they do not depend on the positioning of the coordinate axes. Although a Euclidean procedure may be defined in terms of **X**'s and **Y**'s, a change of origin or axes will not alter the final product, a fitted curve through the given points. This is due to the fact that coordinate changes correspond to rigid motions of the plane as well.

The notion of rating a curve-fitting procedure in terms of the transformations that it commutes with can be specialized even further. Instead of focusing on the quantity of transformations which commute, one may seek procedures which commute with a few distinguished transformations. It may be particularly desirable to commute with a special family of transformations (for example, transformations which define an entire class of map projections). Curve-fitting procedures which commute with all such map projections will produce consistent results on every map image belonging to the projection class.

DEFINING CURVE-FITTING PROCEDURES

So far, "good" and "better" curve-fitting procedures have been described in terms of external associated groups of transformations. No constructive direct approaches to curve-fitting procedures have been given. The only example given so far, the cubic polynomial fit, has fallen far short of being a Euclidean procedure. In this section the relation between Euclidean procedures and Euclidean geometry and its invariants provides the key to designing good procedures.

Large groups of transformations of the plane.

Two very important groups of transformations of the plane are:

(1) The **affine** group; or **affine** transformations; and

(2) The group of **rigid motions** or isometries.

The affine group properly contains the group of rigid motions. Every rigid motion is an affine transformation.

Every affine transformation, **T**, has the form:

$$T((x,y)) = (a_1x + b_1y + c_1, a_2x + b_2y + c_2),$$

for some real constants a_1, b_1, c_1, a_2, b_2, and c_2,

such that $(a_1b_2 - a_2b_1)$ is not equal to zero.

Every rigid motion further satisfies:

(i) $a_1a_2 + b_1b_2 = 0$;
(ii) $a_1 2 + b_1 2 = 1$;
(iii) $a_2 2 + b_2 2 = 1$.

Every rigid motion is a translation (vertical and/or horizontal shifting), a rotation, a reflection, or a composite or combination of two or more of these.

Every affine transformation is either a rigid motion, a stretching or shrinking on each of the axes (with possibly unequal stretching or shrinking on each of the axes), or a composite or combination of the two.

Affine transformations preserve lines, collinearity, and parallelism. Affine transformations do **not** preserve (but rigid motions **do** preserve) perpendicularity, distances, angles and angle bisectors. This is clear in the example:

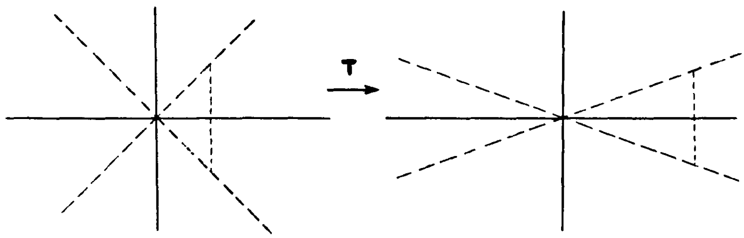

Figure 3. The Affine Transformation: T(x,y) = (3x,y).

In order to illustrate the manner in which known invariant properties can be used to define curve-fitting procedures, the following elementary example is presented:

Given a sequence of points $\{ p_1, p_2, p_3, \ldots, p_n \}$, let the "curve" for that sequence be the polygonal line made up of straight line segments linking successive points p_i and p_{i+1} for **i=1, 2,...,n-1.** The curve-fitting procedure given here could be described as, "link the successive points by straight-line segments."

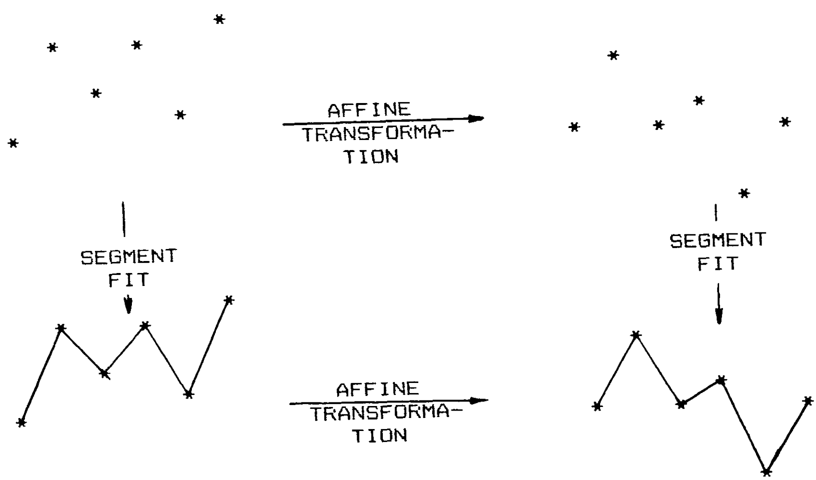

Figure 4. Commutative Diagram for Polygonal Line Fit.

This procedure commutes with all affine transformations **precisely because affine transformations send straight line segments into straight line segments** and a line segment is uniquely determined by its end points. A transformation which did not preserve straightness of lines would not commute with this fit procedure.

Admittedly the polygonal-line fit procedure is not a very interesting or elegant procedure; nevertheless, it illustrates one Euclidean (even affine) fit procedure and a simple means of proving that the procedure indeed commutes with all affine transformations. The next example is more complex and the curves produced are more attractive. The underlying approach is similar, however; and the resulting curve-fitting procedure is both Euclidean and affine.

A GEOMETRIC CURVE-FITTING CONSTRUCTION

This section describes a geometric construction for adding points to a curve one at a time. The procedure may be iterated in order to place points along a curve with any desired density. In particular, for raster plotters or raster display devices, the point generation procedure may terminate when a connected sequence of pixels or raster dots has been selected.

In order to demonstrate the commutativity of this procedure with all affine transformations, the identical construction steps are carried out on the image points of the original fit points; and the corresponding constructed points and line segments are verified to be carried over by the affine transformation at each step.

Assume for this first case that the sequence of points, $p_1, p_2, \ldots, p_n$, does not include **inflections** in the following sense:

For all $i = 2,3,\ldots,n-2$, the points p_{i-1} and p_{i+2} lie on the same side of the line determined by p_i and p_{i+1}.

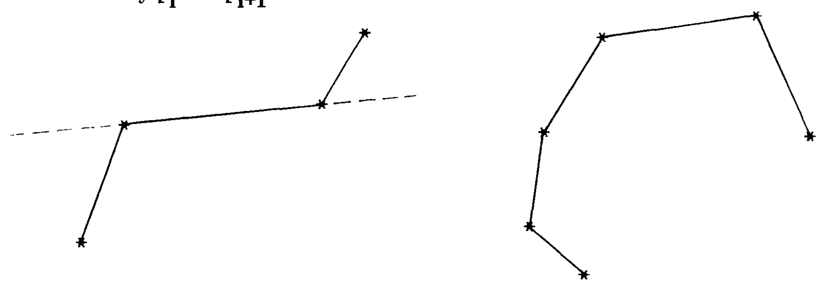

Figure 5. Point Sequences With and Without Inflections.

For this first example it is also assumed that the section of the curve that is being built is an interior section linking the points p_i and p_{i+1}, where i is neither 1 nor n-1. (The construction required at inflections and at end segments is different.)

Observe that if a point sequence is without inflections in the above sense, then the image sequence under any affine transformation is also without inflections.

Let $p_1, p_2, \ldots, p_n$ be a sequence of n points without inflections, and let $p_1', p_2', \ldots, p_n'$, be the corresponding sequence of image points under some affine transformation. Link all successive pairs of points in each sequence with a straight line segment; and then link all alternating pairs of points with additional straight line segments (shown dashed in figure 6). Next construct line l_i through each point p_i parallel to the straddling secant $p_{i-1}\,p_{i+1}$. Do the same for each p_i'. These lines will serve as tangent directions at p_i and p_i' respectively for the curves to be defined. Notice that the corresponding constructions in the original space and in the affine image space are preserved by the affine map; that is, the parallel lines, for example, constructed in the affine image space are the affine images of the parallel lines constructed in the original space. Clearly the point of intersection of a pair of constructed lines in the original space is mapped by the affine transformation to the intersection of the image lines.

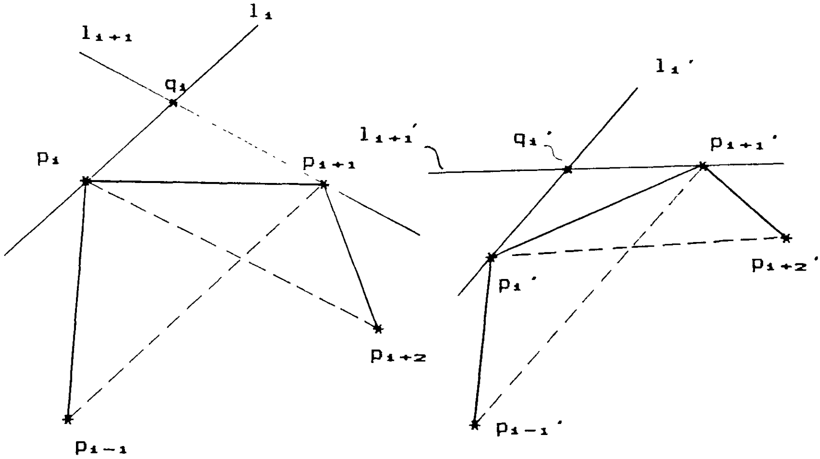

Figure 6. First Stage of Geometric Construction.

Let q_i (resp. q_i') be the intersection of l_i and l_{i+1} (resp. l_i' and l_{i+1}'). Consider the triangles $p_iq_ip_{i+1}$ and $p_i'q_i'p_{i+1}'$. The second triangle is the image of the first under the affine transformation. The next drawing illustrates the construction of a smooth curve through p_i and p_{i+1} which is tangent to p_iq_i at p_i and is tangent to $p_{i+1}q_i$ at p_{i+1}.

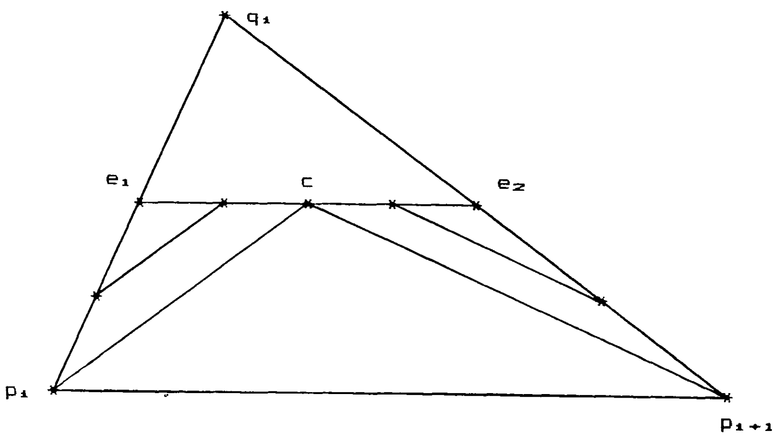

Figure 7. The Triangle for Enclosing the Curve.

Once a triangle has been constructed (in any fashion), the remaining procedure is a lopping or whittling procedure described entirely in terms of geometric characteristics of the triangle, characteristics which are preserved under affine transformations. A locus of points is described which is readily verified to be a smooth curve.

The curve described will be the upper boundary of the figure obtained by successive reduction of the triangular region. First remove a small similar triangle containing q_i by constructing a line parallel to p_ip_{i+1} halfway between q_i and p_ip_{i+1} and discarding the upper triangle and keeping the trapezoid below it. (See figure 7) Let **c** be the center of the upper base of the trapezoid, e_1 and e_2 the end points of the upper base as shown in figure 7 above. Continue by "lopping" the ends at e_1 and e_2 from the triangles p_ie_ic and $p_{i+1}e_2c$ as was done at **q**, again producing two new trapezoids. Construct triangles at the upper corners of each trapezoid, continue lopping until edges become rounded more and more.

With each successive lopping, add the center point of the new upper base of the trapezoid formed to the locus of the curve, since that center point will never be lopped off in any subsequent stage of the locus definition.

Whether one "lops triangles" or "adds" successive center points of trapezoid bases, the result is the same smooth curve. The curve is the closure of constructed points.

It is useful to observe that the triangles, trapezoids, and midpoints of the upper bases described in the construction above are all transformed to corresponding elements by affine transformations. Therefore, the curve fitting procedure described for the piece of the curve between p_i and p_{i+1} is both Euclidean and affine. It is straightforward to verify that the curve will be tangent to p_iq_i at p_i and tangent to $p_{i+1}q_i$ at p_{i+1}. By the earlier specification of the tangent directions for all pieces of the curve, it is clear that the tangent directions fit together at all interior points without inflections.

In order to finish the curve-fitting procedure description, one needs to describe how to handle inflections and end points. Procedures to handle inflections and end points which commute with all affine transformations can be developed, but the ones that the authors have been experimenting with are too elaborate to be presented here. One possible approach to constructing an affine procedure involves building triangles (using affine-invariant triangle constructions) about the fit points so that a triangle lopping procedure may be applied to each triangle.

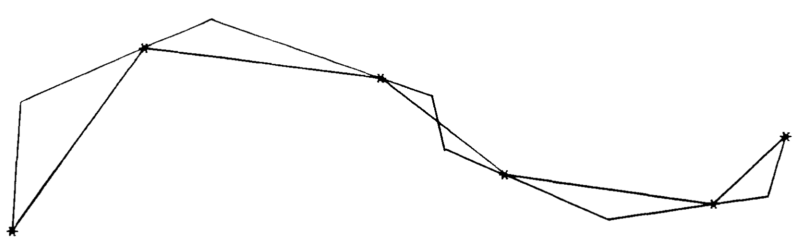

Figure 8. Possible Outcome of A Triangle Building Procedure.

After triangles have been constructed about fit points, another means of building affine-invariant curves involves finding piecewise parametrized cubic polynomial coordinate functions:

$$X_i(t) = A_{i0} + A_{i1}t + A_{i2}t^2 + A_{i3}t^3, \text{ and}$$

$$Y_i(t) = B_{i0} + B_{i1}t + B_{i2}t^2 + B_{i3}t^3;$$

where the parameter **t** above is chosen to be total cumulative straight line distance between successive points in the fit point sequence, the end points of the function pieces are the fit points, and the tangent directions match the slopes of the triangle legs. By specifying the parameter, the end points and the tangents at those end points, the cubic equations are completely defined on each interval between points p_i and p_{i+1}; and the equations mesh at the interval end points. (See White, Fagan, and Saalfeld, "On Fairing a Curve Through a Sequence of Points," for details.)

More generally, the building blocks may be higher order piecewise parametrized polynomial coordinate (PPPC) functions:

$$X_i(t) = A_{i0} + A_{i1}t + A_{i2}t2 + \ldots + A_{in}t^n,$$

$$Y_i(t) = B_{i0} + B_{i1}t + B_{i2}t2 + \ldots + B_{in}t^n,$$

defined for **t** in some interval, $[t_i, t_{i+1}]$.

The collection of PPPC functions is especially useful for building affine-invariant curve-fitting procedures because of the following result:

Lemma. If (X(t), Y(t)) is a curve given by parametrized polynomial coordinate functions of degree less than n in the parameter t on the interval $[t_i, t_{i+1}]$, and if T is an affine transformation; then T((X(t), Y(t))) is also a curve given by parametrized polynomial coordinate functions of degree less than n in the same parameter t on the same interval $[t_i, t_{i+1}]$.

The proof follows from the explicit representation of an arbitrary affine transformation seen earlier:

$$T((X,Y)) = (a_1X + b_1Y + c_1,\ a_2X + b_2Y + c_2),$$

The lemma guarantees that affine transformations will commute with the family of building blocks (PPPC functions) as a whole, moving one building block of the family into another building block of the same family. The trick of a good curve fitting procedure is to match building blocks to sequences of points in such a way that the affine transformations move the point sequences in the same way they move the associated building blocks. Differentiability and, therefore, the existence of a tangent direction for curve points is always preserved by affine transformations because the composition of a linear map with a differentiable map is also differentiable.

BIBLIOGRAPHY

Brown, K.Q., 1979, "Voronoi Diagrams from Convex Hulls," Information Processing Letters, 9(5), pp. 223-228.

Lee, D.T., and B.J. Schachter, 1980, "Two Algorithms for Constructing a Delaunay Triangulation," International Journal of Computer and Information Sciences, 9(3), pp. 219-242.

Pavlidis, T., 1982, Algorithms for Graphics and Image Processing, Rockville, MD, Computer Science Press.

Saalfeld, A., 1984, "Building Penplotter Operational Characteristics into Smoother Curve-Drawing Algorithms," Conference Proceedings: Computer Graphics 84, V.2, Fairfax, VA, National Computer Graphics Association, pp. 598-606.

White, M., and A. Saalfeld and J. Fagan, 1984, "On Fairing a Curve Through a Sequence of Points," internal Census Bureau document, submitted for publication in IEEE Computer Graphics and Applications.