STREAMLINING CURVE-FITTING ALGORITHMS THAT INVOLVE VECTOR-TO-RASTER CONVERSION

A.J. Saalfeld and R.T. O'Reagan
Statistical Research Division
Bureau of the Census
Washington, DC 20233

ABSTRACT

Curve drawing in automated cartographic applications traditionally consists of the following sequence of operations when raster or dot plotters are used to produce the maps or when the map is reproduced on a raster VDT:

1. Select a curve type or family, such as splines.
2. Select a member of that curve family.
3. Select points on the curve and approximate with a polygonal line consisting of vectors.
4. Convert vectors to a series of dots called the raster image of the vectors and plot those dots or illuminate them as pixels on a screen.

The user rarely has control over the vector-to-raster conversion. In general the conversion routine is automatic and transparent to the user. The user may, however, choose his curve and its evaluation vectors more judiciously if he is aware of the conversion procedure. In this paper common vector-to-raster conversion routines are examined in order to select vectors more efficiently and improve appearance of the eventual raster image and the speed of the computations.

New approaches for generating curves and vectors are presented here. These approaches anticipate eventual raster distortions and try to minimize them. Improved appearance of the final product is the principal objective of these new approaches, although speed in computing and drawing the map curve features will be improved as well with the new procedures.

INTRODUCTION

Automated cartography is moving beyond the "stick maps" of some of the original DIME files to publication quality graphics. The Census Bureau is studying means of improving appearance of its computerized cartographic products; and one area of interest is smooth curve drawing. Many irregular features such as rivers and shorelines are stored as sequences of critical points on digital files. In the past those critical points were merely linked with straight line segments to produce the familiar DIME "stick map." Now, at the time of drawing, the curves are reconstructed from the critical points using algebraic and analytic interpolation; and the curves are then plotted using additional software and hardware approximation and conversion methods. The Census Bureau uses a raster plotter for hard copy and both vector and raster displays for VDT's. When the output is raster or pixel images, the traditional procedure for constructing a curve image includes the following steps:

1. Identify fit point sequence and a family of analytic solutions (curve types, such as splines).

2. Select (i.e. compute) an analytic solution from the family (compute coefficients or other "closed" form).

3. Select points along the analytic solution to use for a piecewise linear approximation with vectors or line segments.

4. Convert the vectors to a raster image.

The fourth step is usually done automatically by the hardware. (Curves other than vectors may also be converted directly to rasters by some hardware routines. If those types of curves can be fitted directly to the critical points, then a step may be eliminated.)

Each of the above steps can affect the appearance of the final image adversely by introducing error due to rounding or approximation or by selecting a final image which the particular display device does not show off well. This paper will examine the way in which each step can compound problems of appearance. It will further explore means of shortcutting the four steps; and, finally, it will explore means of anticipating and reducing or avoiding problems in the final step by judiciously choosing curve families, curves, and vectors in the earlier steps.

The principal emphasis in this examination of curve drawing routines is on appearance. Curve drawing for cartography is not primarily concerned with precision of representation, although a degree of fidelity is required. A city must be on the correct side of a river. Beyond that the cartographer cares about the looks of the wiggles in the river.

For screen graphics on a Video Display Terminal (VDT), speed of computation and appearance are tradeoffs. Several very fast vector-to-raster routines based on incremental arithmetic have been developed for raster displays. These fast conversion routines generally produce considerable aliasing or "jaggies," abrupt pixel pattern interruptions. Other slower routines using partial pixel illumination reduce aliasing by blending edges with gray areas. Still other routines smooth lines by using multiple pixel thicknesses which effectively conceal the incremental shifts of the line center. For all of the above routines, the incoming vector is a given—the routine handles it as best it can. For this paper, however, the vector may also be adjusted (within reason) to influence the final raster image. This paper will focus on single pixel width, single intensity illumination representations in order to simplify discussions and illustrations. Nevertheless, methods introduced here may be extended to drawing with gray scale and drawing with multiple pixel width lines.

PROBLEMS WITH THE FOUR-STEP APPROACH TO CURVE-DRAWING

This section will illustrate some of the difficulties arising at each stage of the four-step approach to curve-drawing given in the introduction. One may work backwards from step 4 and step 1, focusing on appearance difficulties at each step:

Step 4. Convert the vectors to a raster image.

The basic problem here is that some vectors convert to nicer raster images than others. Vertical, horizontal, and 45^{o} slopes are the best, of course. Long, nearly vertical or nearly horizontal vectors for which the raster approximation contains a rare abrupt step are most striking to the eye. More frequent steps seem to be integrated (averaged) more readily, somewhat in the fashion that the eye integrates the gray pixels used for anti-aliasing.

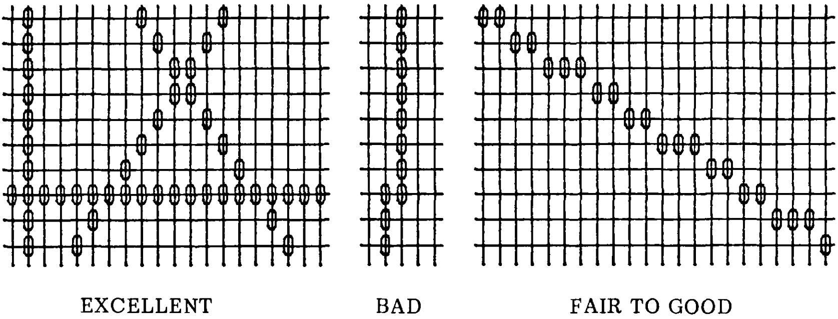

Figure 1. Raster Approximations of Various Vectors.

The above observation suggests that step 3, the selection of vectors (or rather their end points), might include some criteria for avoiding vectors that do not convert well to raster images.

Alternatively one might work with a finite collection of vectors as building blocks, allowing only those vectors which "rasterize" well to enter into the approximating polygonal line. One might assign measures of smoothness to raster images of a small collection of vectors and other measures of smoothness to junctures of such vectors, since the eye also perceives irregularities at such junctures. If this is done, the problem of vector approximation to a curve could be made combinatorial and finite, with the proper solution being that combination of vectors which optimizes the overall smoothness measure.

Step 3. Select points along the analytic solution to use for a piecewise linear approximation with vectors.

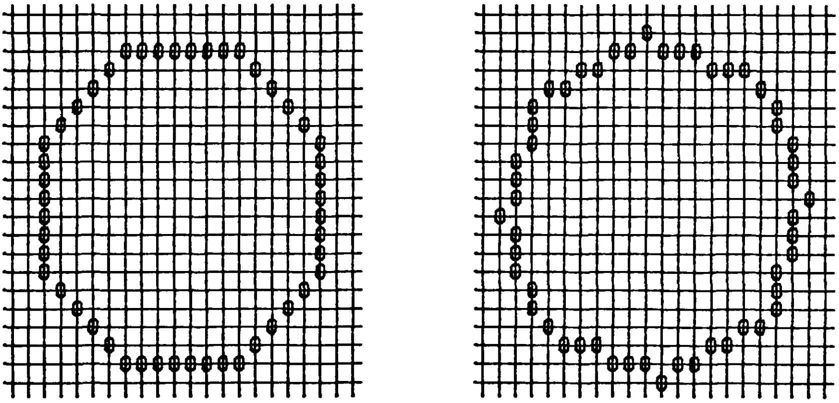

Figure 2. Raster Approximations of Regular Octagons.

For example, if one were to choose 8 points on a circle to approximate the circle by eight vectors, a regular octagon would be a good choice. However, not every regular octagon is equally good for raster conversion. A regular octagon with horizontal and vertical legs is preferable to one with legs sloping slightly off the usual axes because of small angle representation problems seen earlier.

Circle size versus pixel size comes into play in the selection of the approximating octagon as well. The 45^{o} sloping octagon sides can never be **exactly** as long as the horizontal legs, for instance. On discrete grids of rasters or pixels, rounding is an inevitable contributor to distortion.

The following series of illustrations reveals several difficulties which may occur after a smooth analytic solution has been derived for a curve-drawing problem. Additional problems regarding arriving at a smooth analytical solution will be treated in the next section. Here the focus will be on rounding: first from curve to vector end points, then from vector to raster image.

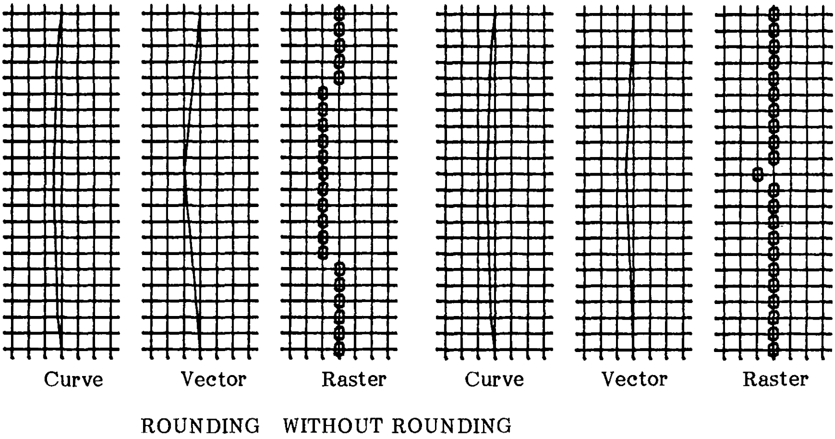

Figure 3 . Effects of Rounding and Not Rounding Vectors Before Conversion to Raster Images.

The points which are chosen to fall on a smooth analytic representation of a curve may be rounded to points on a discrete grid before vectors are determined (as shown on the left above) or they may be rounded when the vectors are converted to raster images (right drawing above).

Rounding vector end points to grid points prior to raster conversion has several advantages:

> If the vectors are computed with discrete grid end points, the raster images will be easier to study for irregularities.
>
> Furthermore, raster images of vectors whose end points are grid points may always be endowed with a central point symmetry which guarantees a more regular appearance. That symmetry may be attained by utilizing an incremental pixel selection procedure such as Bresenham's linear algorithm from each end of the vector to the center point. (See figure 4)
>
> A third advantage of rounding vectors to grid points before converting to the raster image is that the initial directional move (of the 8 possible) for an incremental procedure will always be in the correct sector. This property of the initial directional move is a consequence of positioning the vector end point exactly on the grid point, thereby eliminating rounding effects at the first pixel placement. Since this result will be true for all vectors entering a particular juncture, smoothness at junctures will be improved. (See figure 5)

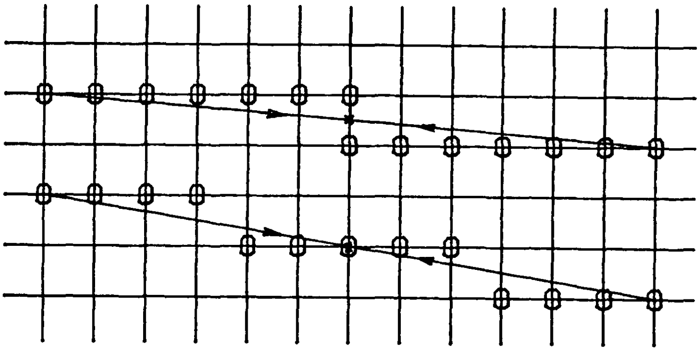

Figure 4. Results of Bresenham's Algorithm Applied from both Ends to the Center.

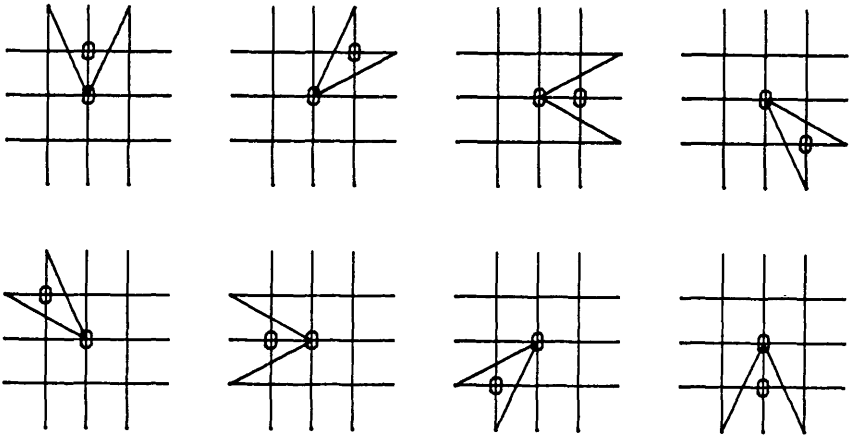

Figure 5. Sector Choices for Initial Directions.

However, the raster images will suffer more total displacement in general if the vectors are rounded prior to raster conversion.

If, on the other hand, the vectors are not rounded to the discrete grid prior to conversion to raster images, then the illuminated pixels will be closer to the continuous polygonal line curve approximation whose points actually lie on the curve. Nearness to the true smooth curve does not guarantee a smoother raster image, as the illustration below indicates.

In addition the raster image may exhibit abrupt atypical directional discontinuities precisely at the junctures of two approximating vectors, further degrading the smooth appearance of the raster image. An angle of 50°, for example, is represented by a sequence of pixels of steps predominantly 45° with some infrequent steps of 90°. A step of 45° is better at a juncture with another vector because it is only 5° from the intended direction. If both vectors at a juncture have large and opposite atypical steps, the effect can be a directional change nearly 90° greater than the intended directional change. (See figure 6)

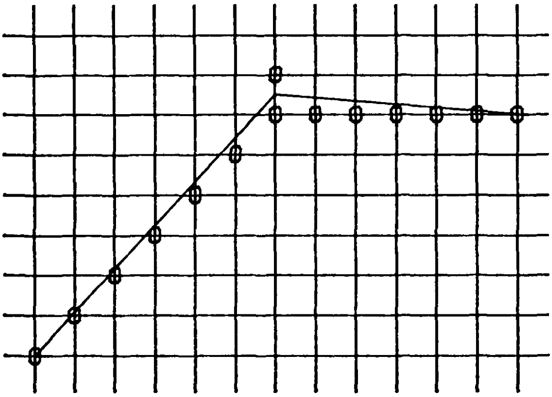

Figure 6. Dangling Pixel Due to Unrounded Vectors.

Vector selection options to approximate a continuous curve (step 3) provide many opportunities for improving ultimate appearance of the raster image. The number of vectors, the placement of vector end points along the curve, and the rounding of those end points prior to converting to raster images all affect the output. The previous step can also influence appearance:

Step 2. Select an analytic solution from the family of curves.

In general this solution will be uniquely determined by the curve family and some additional constraints placed on the curve by the user. Those constraints may involve specifying tangent values or restrictions on higher derivatives. The constraints, while somewhat arbitrary, are added to guarantee uniqueness and replicability of the fitting procedure. The complexity of the solution is directly related to the number of constraints that the user may specify. For example, a polynomial of degree **n** can handle **n+1** constraints.

Instead of picking constraints such as higher derivative values which merely serve to guarantee uniqueness of the analytic solution, the user may seek constraints which facilitate the vector selection procedure in step 3. For example, if the curve passes through grid points, then rounding the vector end points will not be necessary. Because intermediate values on the curve are critical in determining the vectors to be used, a procedure which anticipates later steps is one which chooses the particular analytic solution based on properties of intermediate values.

It is not generally easy to specify intermediate curve values prior to specifying the curve, although some such procedures do exist (see the paper by Fagan and Saalfeld in these **Proceedings**). A greater potential for improving eventual raster appearance lies in the selection of the curve family itself:

Step 1. Select a family of curves.

For some families of curves, the subsequent steps may be more easily implemented or simplified. One example of such a family is the family of parametrized polynomial coordinate functions:

$$X(t) = A_0 + A_1t + A_2t^2 + \ldots + A_nt^n.$$

$$Y(t) = B_0 + B_1t + B_2t^2 + \ldots + B_nt^n.$$

The theory of finite differences permits any polynomial to be evaluated at regular intervals by means of successive incrementing operations. For example, linear functions have a constant increment, quadratics have a constantly increasing or linear increment, etc.; thus intermediate values of polynomials may be generated using incremental arithmetic only. Methods which exploit the incremental relations are currently being studied at the Census Bureau.

One final possibility which arises in relation to the last example is that of skipping steps in the four-step procedure. An incrementally generated curve for a curve-fitting procedure has great potential for direct raster specification. If one can eliminate the need for vector approximation (steps 2 and 3) and move directly to step 4, potential gains in processing speed will be considerable.

The idea of skipping steps 2 and 3 is not new. Several slow procedures illuminate all pixels near to a curve simply by evaluating the curve on a very dense set of points. The principal deficiency of such an approach is the poor conversion results for some curves. As seen in figure 6, the nearest raster image to an arbitrary curve may contain some dangling pixels which degrade appearance. The new focus proposed here is on appearance. A procedure which selects pixels on a curve successively based on incremental tests can eliminate dangling pixels due to rounding and simultaneously increase computation speed.

BIBLIOGRAPHY

Bresenham, J.E., 1965, "Algorithm for Computer Control of Digital Plotter," IBM Syst. J., 4(1), pp. 25-30.

Bresenham, J.E., 1977, "A Linear Algorithm for Incremental Display of Circular Arcs," Communications of the ACM, 20(2), pp. 100-106.

Foley, J., and A. Van Dam, 1984, Fundamentals of Interactive Computer Graphics, Reading, MA, Addison-Wesley.

Jordan, B.W., and W.J. Lennon and B.C. Holm, 1973, "An Improved Algorithm for the Generation of Non-Parametric Curves," IEEE Transactions on Computers, pp. 1052-1060.

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

Pitteway, M., 1967, "Algorithm for Drawing Ellipses and Hyperbolae with Digital Plotter," Computer Journal, 10(3), pp. 282-289.

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.

VanAken, J.R., 1984, "An Efficient Ellipse-Drawing Algorithm," IEEE Computer Graphics and Applications, 4(9), pp. 24-35.