On the Quality Rendering of Curves for Generic Output Discrete Devices

Antonio Elias Fabris

Computer Graphics and Applied Computational Geometry Project (CGCAP)
Instituto de Matemtica e Estatstica, Universidade de So Paulo
Caixa Postal 66281, 05315-970, São Paulo-SP, Brazil
aef@ime.usp.br

Abstract

These notes present a new antialiasing pre-filtering technique which enables high quality rendering of curves for generic output discrete devices such as Cathode Ray Tube and Liquid Crystal Displays, Laser-printers, etc. It is also provided the correct handling of complex geometries including for example self intersections, small loops, cusps, curves with high and small radius of curvature. Moreover, the technique allow for rendering curves of arbitrary thickness and can be optimally tuned to the bits used for image quantization.

Keywords: Point Containment Methods, Discrete Geometry, Computer Graphics, Antialiasing.

1 Introduction

Aliasing artifacts are a consequence of the errors introduced by attempting to represent a continuous model on a discrete device. For instance, a scene containing sharp changes in colour intensity can exhibit serious defects when represented by discrete samples such as dots on the screen. The error of this sampling process was originally called aliasing in Signal Processing Theory and this terminology has been adopted in the field of Computer Graphics. Since Crow [8] identified the aliasing problem in Computer Graphics, there have been a considerable number of published papers on the subject. Most of them include antialiasing as part of the general process of rendering using an antialiasing technique developed elsewhere. A small number are dedicated to specifically developing an antialiasing technique. Among them, we distinguish two main approaches to the aliasing problem in Computer Graphics: pre- and post-filtering methods [11].

In post-filtering the ideal image is point sampled at a higher rate than can be displayed, with multiple samples per pixel, and the samples are then numerically filtered by a discrete digital filter. The advantages of post-filtering lie largely in the simplicity of the process. In some applications, such as ray tracing, super-sampling followed by post-filtering is the only reasonable solution to aliasing. The disadvantages are the greatly increased costs of computation and the failure to eliminate aliasing completely.

If a precise geometric description of the ideal image is available, say analytic or piecewise analytic descriptions, then it is theoretically possible to filter the ideal image analytically to remove high frequency spatial components before sampling and then to sample the bandwidth limited ideal image to produce alias-free images. For example, for simple geometries such as straight lines, we can pre-filter the geometry by a box filter by deriving expressions for pixel area coverage: sampling is then reduced to substitution of pixel coordinates in the expressions. The advantages of pre-filtering are that once the ideal image has been analytically pre-filtered, a single sample per pixel suffices. The expectation is that if we can properly pre-filter the ideal image, then we will generate high quality displayed images. The main disadvantages heretofore have been the restricted range of geometries that can be pre-filtered and the restricted range of filters which could be employed. Pre-filtering will however remain a technique more appropriate to high quality two-dimensional images containing lines, regions and text rather than images of three-dimensional scenes [11].

2 Previous Pre-Filtering Techniques

Gupta and Sproull [19] developed an antialiased version of the Bresenham line algorithm in which, at least notionally, the ideal line is convolved with a circularly symmetric filter. In their implementation a lookup table is constructed which contains the fractions of the volume of the conical filter intercepted by the ideal line, indexed by the perpendicular distance of the pixel center from the centre of the line. This table can be generated from a simply derived analytic expression. The circular symmetry enables a single one-dimensional table to cover lines at all possible angles. Analytic expressions for line ends, however, are difficult to derive and Gupta-Sproull resort to less precise two-dimensional tables. Whilst circular symmetry leads to a fast incremental algorithm, the conical filter does not give rise to a flat constant signal for constant sample values, thus exhibiting what Mitchell and Netravali call sample-frequency ripple. Forrest gives examples of this for various filter radii in [17]. Pitteway and Watkinson [27] describe the incorporation of area sampling in the Bresenham line algorithm, but do not handle line ends properly.

Feibush, Levoy and Cook [15] describe antialiasing of polygons in which polygons are split into triangles and the volume of the filter, conical for example, intersected by each triangle is used as a weight. In effect this amounts to a discrete approximation to the convolution integral applied to polygon fragments. A two-dimensional lookup table is used to store triangle-cone intersections for efficiency reasons. If the chosen filter is not circularly symmetric, then the lookup table is considerably more complex, being four-dimensional. Abram, Westover and Whitted [1] develop a more complex approach in which polygon-filter convolution is classified into several different cases, each of which employs a discrete approximation to the convolution integral.

No algorithms appear to guarantee that the numerical approximations expressed as a lookup tables are good enough to avoid unwanted aliasing artifacts. For instance, small objects of a scene whose data fall between the quantization steps of the look up tables will disappear completely. In this sense, table look up methods behave similarly to the post-filtering methods. According to Duff [9], exploring this fact, it is possible to make images for which methods using look up tables fail miserably. Duffs polygon scan conversion by exact convolution [9] numerically integrates the convolution integral to antialias simple polygons. Filters are more general than in other methods but are limited to bivariate polynomials which can be used to approximate sinc and other filters.

McCool [24] describes a method whereby an image consisting of Gourad-shaded triangles can be represented by simplex splines; these can then be convolved with a box spline filter to form a set of prism splines representing the filtered image. This permits analytic filtering by filters which can be constructed from box spline basis functions, a special case being tensor product B-splines. Any further filtering for reconstruction is performed by digital post-processing.

Aliased scan conversion of curves is still a topic for further research: whilst there exist efficient algorithms for circles and ellipses (although long thin ellipses still need special care), more general parametric, explicit or implicit curves generally require careful attention to both geometric and numerical detail in order to provide robust and efficient algorithms. In many cases the approach taken is to reduce the curve to a piecewise linear approximation which can then be antialiased; avoiding any visual evidence of polygonisation requires care.

Lien, Shantz and Pratt [23] develop an adaptive forward difference method for rendering curves and briefly mention a simple adaptation to their algorithm which enables an admittedly rough approximation to area sampling to be made. In effect the curve is approximated by short straight line segments. Klassen [22] remarks on the geometric problems found in rendering curves, particularly loops, cusps, and crossings, and goes on to develop a more robust approach than [23]. A curve is split into monotonic sections (with respect to the x or y axes) then adaptive forward differencing is used to divide the curve into short straight line segments which may then be antialiased by the Gupta Sproull algorithm [19] or by any other filter which can be accessed from a lookup table in a similar manner. The transitions from x major to y-major line segments (and vice versa) need special attention to avoid nicks in the output. Klassen pays particular attention to numerical detail.

Field [16] describes a fast incremental method for antialiasing circles and ellipses based on a predictor- corrector method to compute filter values. The filter employed is an approximation to area sampling. Pitteway and Banissi [26] describe an integer algorithm for rendering antialiased ellipses which employs an approximation to area sampling sufficient for a two bit per pixel system, this giving a marked improvement in the rendering of fonts composed of conic segments.

Prior methods are seen to be restricted in terms of filters, employing either box or conical filters which are known to be poor. Curves, apart from circles and ellipses, are approximated by line segments, and geometric special cases need careful treatment.

3 Discrete Bézier Curves

Discrete curves can be described by an ordered sequence of points with integer coordinates whose distance between consecutive points is less or equal to 1. A finite sequence of points is a function $L : [0..n] \to \mathbb{Z}^2$ called *list of length n*. We will denote $L(i)$ by L_i, the length of L by $\#L$ and the set of all lists by Λ.

Discrete Bézier curves are the discrete counterparts of the well-known continuous Bézier curves. Note that a list can represent a discrete curve as well as the control points of the continuous Bézier curve. One essential operation that can be defined on Bézier curves is subdivision. Subdivision of a Bézier curve will result in two Bézier curves called the *left* and the *right part*, Figure 1. More precisely, let L be a list of length n, we define $\mathcal{L} : \Lambda \to \Lambda$ and $\mathcal{R} : \Lambda \to \Lambda$ by

$$\mathcal{L}(L)_i \mapsto \frac{1}{2^i} \sum_{j=0}^{i} \binom{i}{j} L_j \tag{1}$$

$$\mathcal{R}(L)_i \mapsto \frac{1}{2^{n-i}} \sum_{j=i}^{n} \binom{n-i}{j} L_j. \tag{2}$$

$\mathcal{L}(L)$ is the left part and $\mathcal{R}(L)$ is the right part of L.

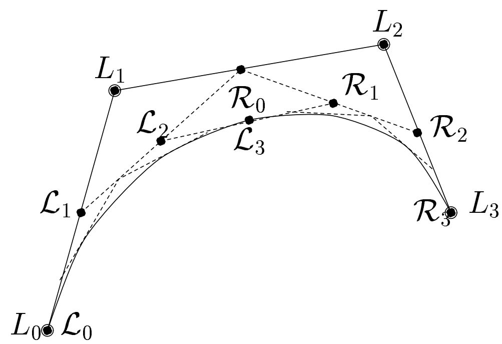

Figure 1: Left and Right Bézier Curves.

In order to implement subdivision in an integer-only algorithm, we have to map $\mathcal{L}(L)_i$ and $\mathcal{R}(L)_i$ given by (1) and (2) to the set of integers. This is done by a so-called *snapping function*. Familiar examples of these functions are *truncation* and *rounding functions*.

Using a generic snapping function, we can produce discrete versions of the left and the right parts putting

$$\mathcal{L}'(L)_i = snap(\mathcal{L}(L)_i)$$

$$\mathcal{R}'(L)_j = snap(\mathcal{R}(L)_j).$$

This process enables the construction of a recursive algorithm to generate the outline of a discrete polynomial Bézier curve. For this, we consider a list L of length n and the 8–connected diameter of L, $d = \max_{i,j}\{|L_j - L_i|_{V^8}\}$, where $|.|_{V^8}$ represents the 8-distance in $\mathbb{Z}^2$. Then, define the curve–to–polygonal conversion operator $\mathcal{C} : \Lambda \to \Lambda$ by:

$$d \leq 1 \Rightarrow \mathcal{C}(L) = \langle L_0, L_n \rangle$$

$$d > 1 \Rightarrow \mathcal{C}(L) = \mathcal{C}(\mathcal{L}'(L)) \uplus \mathcal{C}(\mathcal{R}'(L)),$$

where $\uplus$ is the concatenation between lists. In [7], Corthout and Pol prove that the operator $\mathcal{C}(L)$ is well–defined, that is, the stopping criterion can always be reached in a finite number of subdivisions and the discrete left and right parts can always be concatenated.

The conversion operator $\mathcal{C}(L)$ produces an 8-connected list. Using this fact, we can consider the discrete space $\mathbb{Z}^2$ as always 8–connected and this configuration will enable a simplification of our algorithm.

4 Antialiasing of Continuous Curves

This section presents a new technique for antialiasing two-dimensional continuous curves. Basically, the technique point-samples a pre-filtered colour intensity function of a continuous curve with an arbitrary width. Pre-filtering is carried out by using a generic class of filters and the sampling process is based on the point containment detection.

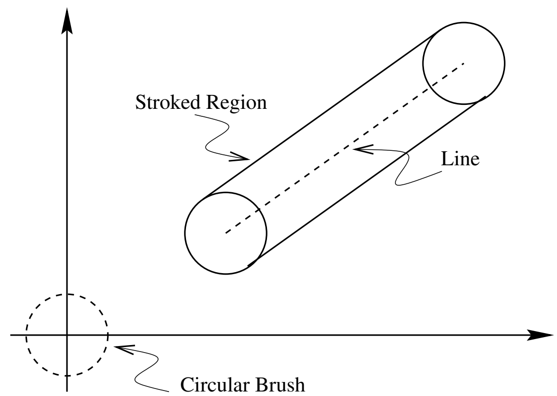

Figure 2: Stroking with a circular brush.

First we define the colour intensity function of a continuous curve in $\mathbb{R}^2$. Lines and curves may be generated by continuous sweep of a brush along the mathematical center of the line or curve (see Figure 2). Guibas et al. [18] refers to this process as convolving the curve with a brush, but in this paper we shall use the term *dilation* to denote the effect of brushing in order to avoid confusion with convolving a geometric object with a filter. The notion of dilation is used in the PostScript imaging model which is based on the notion of painting with a opaque paint on a plane [2]. In PostScript, the paint is applied by a pen or brush of user-specified width. Dilation will also be used later in this section to define the discrete image of a curve in $\mathbb{Z}^2$. For this reason we will consider an arbitrary module over a ring $K = \mathbb{Z}$ or $K = \mathbb{R}$. Let A and B be sub-sets of a module V over a ring K.

For each $v \in V$, the translation operator $t_v : V \longrightarrow V$ is defined by $t_v(w) = v+w$. The *dilation of A by B* is defined by the Minkowsky addition

$$A \oplus B = \bigcup_{b \in B} t_b(A) \; .$$

Let $\Gamma : [0,1] \longrightarrow \mathbb{R}^2$ be a given continuous curve. For each $w \in \mathbb{R}$, $w \geq 0$, denote $\mathcal{D}_{w/2} = D(w/2) \oplus \Gamma([0,1])$ where $D(w/2)$ is the disk in $\mathbb{R}^2$ with centre at the origin

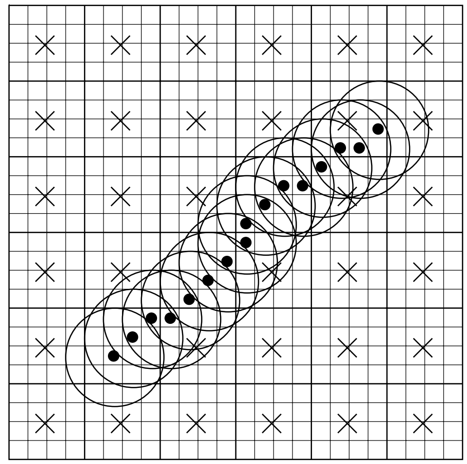

Figure 3: Discrete curve dilated by a disk.

and radius $w/2$. The *colour intensity function of the w-pixel wide curve* Γ is defined by

$$\mathcal{I}_w(x) = \begin{cases} 1 & \text{if } x \in \mathcal{D}_{w/2} \\ 0 & \text{if } x \in \mathbb{R}^2 - \mathcal{D}_{w/2} \end{cases}$$

As we have remarked earlier, the conventional approach to rendering would be to scan convert the convolved version of the curve but this is known to lead to numerical and geometric problems [11]. Instead we chose to use the *point containment algorithm* developed by Corthout et al. to implement a pre-filtering antialiasing algorithm. In [4], Corthout and Jonkers describe an algorithm for determining the containment of a point within a region bounded by discrete Bézier curves. This is extended in [5] to encompass discrete rational Bézier curves and in [6, 7] to support dilation and erosion of discrete Bézier curves and regions bounded by discrete Bézier curves by brushes which may be regions bounded by discrete Bézier curves. Corthout and Pol's thesis [7] contains full details of the mathematical theory of discrete Bézier curves, a version of the Jordan curve theorem for regions bounded by discrete curves, a formal development of the point containment algorithm and a description of its implementation in dedicated silicon, the Pharos chip fabricated by Philips. A discrete curve can be computed by subdivision of an integer grid of specified resolution. Figure 3 shows a typical 8-connected discrete curve of width 1.25 pixels computed on a grid with 4 times pixel resolution. We show a circular brush centered on each point of the discrete curve. Pixels whose center lie within the dilated curve are rendered. The accuracy of the discrete intersection test is a function of the sub-pixel resolution chosen for the discrete curve.

To give a description of the pre-filtering process, we first define our filter space.

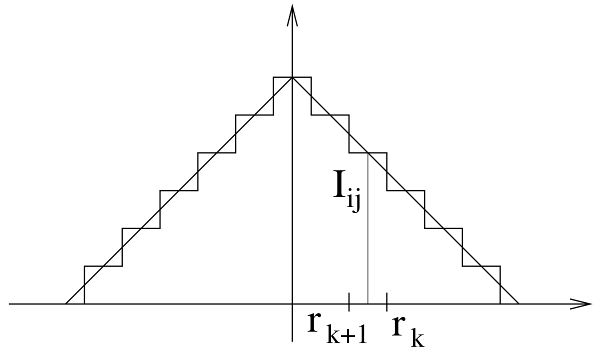

Figure 4: Stack of nested brushes.

Any practical filter in Computer Graphics is an even function having a finite support and unit integrated intensity [21]. Moreover, a filter restricted to its support is continuous. Then a filter f is a function $f : \mathbb{R} \longrightarrow \mathbb{R}$ satisfying the conditions

- $f_{\,|\,[-a,a]} \in C([-a,a])$, $\quad support(f) = [-a,a]$
- $f(x) = f(-x) \;\; \forall x \in \mathbb{R}$
- $\int_{-\infty}^{\infty} f(x)dx = 1$

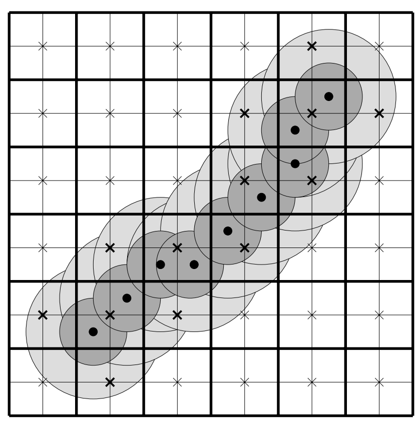

Figure 5: Nested regions along a discrete curve.

Let v_w be the convolution product between a filter f and the *w-box function* defined by

$$b_w(x) = \begin{cases} 1 & \text{if} \quad |x| \le w/2 \\ 0 & \text{if} \quad |x| > w/2 \end{cases}$$

That is,

$$v_w(x) = \int_{-\infty}^{+\infty} b_w(t)\, f(x-t)dt \;=$$

$$\begin{cases} \displaystyle\int_{-w/2}^{+w/2} f(x-t)dt & \text{if} \quad |x| \leq a + w/2 \\ \qquad 0 & \text{if} \quad |x| > a + w/2 \end{cases}$$

The convolution v_w between the cross section of the ideal curve (the w-box function) and a filter f is then approximated in a piecewise manner to create a stack of brushes, as shown in Figure 4 for the box filter convolved with the 1-box function. Let γ be a connected discrete curve obtained from Γ by some rasterization scheme. Let $0 < r_m < \cdots < r_2 < r_1 < a + w/2$ ($r_i \in \mathbb{R}$) be an uniform partition of the interval $[0, a+w/2]$ and let N_i be the γ-neighborhood obtained by stroking γ with the brush of radius r_i. The sequence $N_1, N_2, \cdots, N_m$ is a nested sequence of γ-neighborhoods [11]. Note that, although r_i are real numbers, the regions N_i are contained in the discrete plane $\mathbb{Z}^2$.

The technique presented in this paper detects whether a pixel coordinate (x_i, y_j) is inside or outside one of the disjoint regions

$$\widetilde{N}_1 = N_1 - N_2\,,\ \widetilde{N}_2 = N_2 - N_3\,,\ \cdots\ ,\ \widetilde{N}_m = N_m\ .$$

Let $\mathcal{D}$ be a mapping from $\mathbb{R}$ to $\mathbb{Z}$. The familiar truncation and rounding functions are examples of such a mapping. Let β be the available number of grey levels. The colour intensity function of the *w-pixel wide discrete image of* Γ is the function $\widetilde{\mathcal{I}}_w : \mathbb{Z}^2 \longrightarrow \mathbb{Z}$ defined by

- If (x_i, y_j) is outside N_1, $\widetilde{\mathcal{I}}_w = 0$
- For $k = 1, \cdots, m-1$, if (x_i, y_j) is inside $\widetilde{N}_k$,

$$\widetilde{\mathcal{I}}_w = \mathcal{D}(2^\beta v_w(\frac{r_{k+1} + r_k}{2}))$$

- If (x_i, y_j) is inside $\widetilde{N}_m$,

$$\widetilde{\mathcal{I}}_w = \mathcal{D}(2^\beta v_w(0))$$

This definition characterizes our antialiasing technique for two-dimensional continuous curves. Figure 5 shows a discrete curve generated at 2 times device resolution with the convolved brush centred at the points of the discrete curve giving a discrete antialiased curve which is sampled at the pixel centres to give the pixel values.

Details of the method employed for generation of the brush stack by optimal discrete piecewise approximation of the intensity profile of the filtered brush are given in [11]. On-demand generation of sections of the brush profile rather than a pre-computed set of brush slices would enable a root finding procedure to determine the fraction to arbitrary precision by interval halving.

5 Results

Greyscale illustrations were computed at 6×6 sub-pixel resolution and viewed on an Apple Macintosh with a colour monitor using the "special gamma" setting. For reproduction, the images were saved as PostScript files and printed on a Hewlett-Packard LaserJet 4 Plus printer at 600 dpi using the "calibrated colour greyscale" option in the standard print dialogue. Images viewed on the screen are rather smoother as a consequence of the blurring effect of the gaussian-type reconstruction of the CRT. Differences between filters are less noticeable than in the printed versions. Laser printing allowed us more control over greyscale than would have been possible using photography. The illustrations are best viewed from approximately 0.5 metres using a strong incandescent bulb for lighting.

A set of Bézier curves were chosen to demonstrate particular geometric features [11]:

- a cubic curve with two near-vertical and one near-horizontal portions;
- a parabola with a small radius of curvature;
- a parabola with a large radius of curvature;
- a cubic with a cusp;
- a small loop obtained by a slight perturbation of the cusp's control points;
- a cubic with two inflection points also obtained by a slight perturbation of the cusp's control points;
- a large loop.

Plate 1 shows the curves drawn aliased using the point containment algorithm. In Plates 2–4 we use a circular brush convolved with a variety of filters to create stacks of circular brushes. Plate 2 shows the seven curves rendered as one pixel wide curves using the Mitchell–Netravali filter [26]. In Plate 3 we demonstrate the rendering of the low curvature parabola using a variety of filters. On balance the Mitchell-Netravali filter proved the best compromise between sharpness and smoothness or lack of braiding. The poor performance of box filtering is obvious. Plate 4 illustrates the ability of the technique to render curves with a variety of thicknesses and also the correct handling of a tight loop which is progressively filled in as the curve thickness increases.

6 Discussions, Conclusions and Further Work

We have presented a pre-filtering technique to render cur- ves with arbitrary thickness using a generic class of filters. In [11, 12] we describe how stacks of brushes can be generated to approximate the convolved brush to any required accuracy. We pay a price in terms of efficiency. In [7], Corthout and Pol show that the major disadvantage of Point Containment algorithms – their quadratic time complexity with increasing resolution – can be counteracted by applying coherence detection of large uniformly coloured parts of the output bitmap. Using quadtrees and the convex hull properties of discrete Bézier curves the time complexity of the Point Containment algorithm can be reduced to quasi-linear. In [13, 14], Fabris et al. reduce even more this time complexity: they present a method whose time complexity depends not on the resolution but only on the perimeter of the polygon boundary. Moreover, as the tests for Point Containment are independent for different pixels of the image, Point Containment based algorithms can be implemented in a variety of parallel configurations with great gains of performance [10].

Cases where the curve has more than one intersection with a pixel, for example where a curve crosses itself, needs further investigation. As presently implemented, we detect the smallest (highest) brush intersected, thus computing a maximum value over all the curve's points within the pixel. There are no apparent problems with the looping curves in Plates 2 and 4. Properly, global knowledge of the curve configuration should be used, rather than serial exploration. The problem is related to the bulge elimination problem discussed by Bloomental in the context of generating implicit branching surfaces by convolution of skeletons [3] and the solution may lie there or in investigation of level curves [20, 25].

As presently implemented, our technique covers only the antialiasing of curves as strokes and assumes black curves drawn over a white background permitting write-only image generation. Future implementations will include antialiasing of region boundaries and the use of read-modify-write (lerping). More complex brushes such as orientable brushes and brushes defined by closed sequences of Bézier curves need investigation.

7 Acknowledgment

The work reported in this paper was supported by Grant Number 97/03055-6 from the Researcher Support Foundation of the State of São Paulo (FAPESP).

References

[1] G. Abram, L. Westover, and T. Whitted. Efficient alias-free rendering using bitmasks and look-up tables. *Computer Graphics (SIGGRAPH'85 Proceedings)*, 19:53–59, July 1985.

[2] Adobe Systems, Inc. *PostScript Language Reference Manual.* Addison-Wesley, Reading, MA, 1985.

[3] J.I. Bloomenthal. Bulge elimination in convolution surfaces. *Computer Graphics Forum*, 16(1):1–11, 1997.

[4] M.E.A. Corthout and H.B.M. Jonkers. A new point containment algorithm for B-regions in the discrete plane. In R.A. Earnshaw, editor, *Theoretical Foundations of Computer Graphics and CAD*, NATO Advanced Study Institute Series, Series F, F40, pages 297–306. Springer-Verlag, 1988.

[5] M.E.A. Corthout and H.B.M. Jonkers. A point containment algorithm for regions in the discrete plane outlined by rational bézier curves. In J. André and R.D. Hersch, editors, *Raster Imaging and Digital Typography*, pages 169–179. Cambridge University Press, 1989.

[6] M.E.A. Corthout and E.-J.D. Pol. Supporting outline font rendering in dedicated silicon: the pharos chip. In *Raster Imaging and Digital Typography II*, pages 177–189. Cambridge University Press, 1991.

[7] M.E.A. Corthout and E.-J.D. Pol. *Point Containment and the PHAROS Chip.* PhD thesis, University of Leiden, Leiden, March 1992.

[8] F.C. Crow. The aliasing problem in computer-generated shaded images. *Communications of the ACM*, 20(11):799–805, November 1977.

[9] T. Duff. Polygon scan conversion by exact convolution. In J. André and R.D. Hersch, editors, *Proceedings of the International Conference on Raster Imaging and Digital Topography*, pages 154–169. Cambridge University Press, 1989.

[10] A.E. Fabris, M.M.O. Assis, and A.R. Forrest. Parallelization of filling algorithms on distributed memory machines using the point containment paradigm. In *SIBGRAPI'2000 Proceedings*, volume 13, pages 235–243. IEEE Computer Society Press, October 2000.

[11] A.E. Fabris and A.R. Forrest. Antialiasing of curves by discrete prefiltering. *Computer Graphics (SIGGRAPH'97 Proceedings)*, 31(3):316–327, August 1997.

[12] A.E. Fabris and A.R. Forrest. High quality rendering of two-dimensional continuous curves. In *SIBGRAPI'97 Proceedings*, volume 10, pages 10–17. IEEE Computer Society Press, October 1997.

[13] A.E. Fabris, L. Silva, and A.R. Forrest. An efficient filling algorithm for non-simple closed curves using the point-contaninment paradigm. In *SIBGRAPI'97 Proceedings*, volume 10, pages 1–9. IEEE Computer Society Press, October 1997.

[14] A.E. Fabris, L. Silva, and A.R. Forrest. Stroking discrete polynomial Bézier curves via point containment paradigm. In *SIBGRAPI'97 Proceedings*, volume 11, pages 1–9. IEEE Computer Society Press, October 1998.

[15] E.A. Feibush, M. Levoy, and R.L. Cook. Synthetic texturing using digital filters. *Computer Graphics (SIGGRAPH '80 Proceedings)*, 14(3):294–301, July 1980.

[16] D.A. Field. Algorithms for drawing anti-aliased circles and ellipses. *Computer Graphics and Image Processing*, 33(1):1–15, January 1986.

[17] A.R. Forrest. Antialiasing in practice. In R.A. Earnshaw, editor, *Fundamental Algorithms for Computer Graphics*, NATO Advanced Study Institute Series, Series F, F17, pages 113–134. Springer-Verlag, 1985.

[18] L.J. Guibas, L.H. Ramshaw, and J. Stolfi. A kinetic framework for computational geometry. In *Proceedings of 24th IEEE Symposium on the Foundations of Computer Science*, pages 100–111, 1983.

[19] S. Gupta and R.F. Sproull. Filtering edges for gray-scale displays. *Computer Graphics (SIGGRAPH '81 Proceedings)*, 15(3):1–5, August 1981.

[20] W. Heidrich, M.D. McCool, and J. Stevens. Interactive maximum projection volume rendering. In G.M. Nielson and D. Silver, editors, *Proceeding of IEEE Visualization '95*, pages 11–18, October 29 – November 3 1995.

[21] R.V. Klassen. *Device Dependent Image Construction for Computer Graphics.* PhD thesis, University of Waterloo, Ontario, July 1989.

[22] R.V. Klassen. Drawing antialiased cubic spline curves. *ACM Transactions on Graphics*, 10(1):92–108, January 1991.

[23] S.-L. Lien, M. Shantz, and V.R. Pratt. Adaptive forward differencing for rendering curves and surfaces. *Computer Graphics (SIGGRAPH '87 Proceedings)*, 21:111–118, July 1987.

[24] M.D. McCool. Analytic antialiasing with prism splines. *Computer Graphics (SIGGRAPH '95 Proceedings)*, 28:221–228, August 1995.

[25] M.D. McCool. Private communication, September 1996.

[26] D. P. Mitchell and A. N. Netravali. Reconstruction filters in computer graphics. *Computer Graphics (SIGGRAPH '88 Proceedings)*, 22:221–228, August 1988.

[27] M.L.V. Pitteway and D. Watkinson. Bresenham's algorithm with grey scale. *Communications of the ACM*, 23:625–626, 1980.

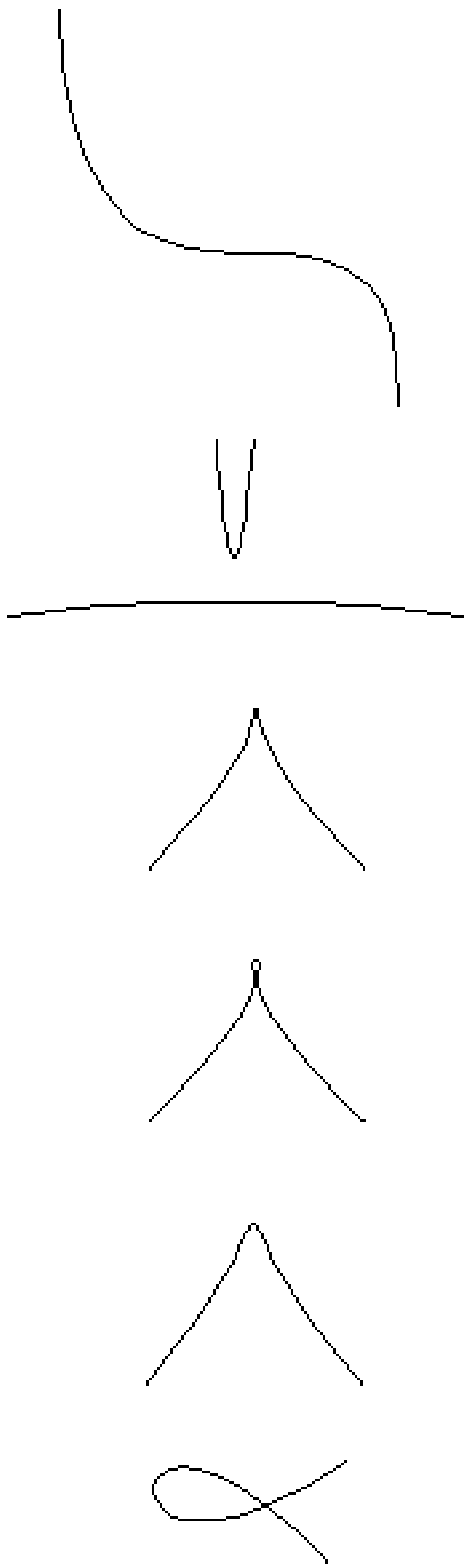

Plate 1: Aliased Test Curves

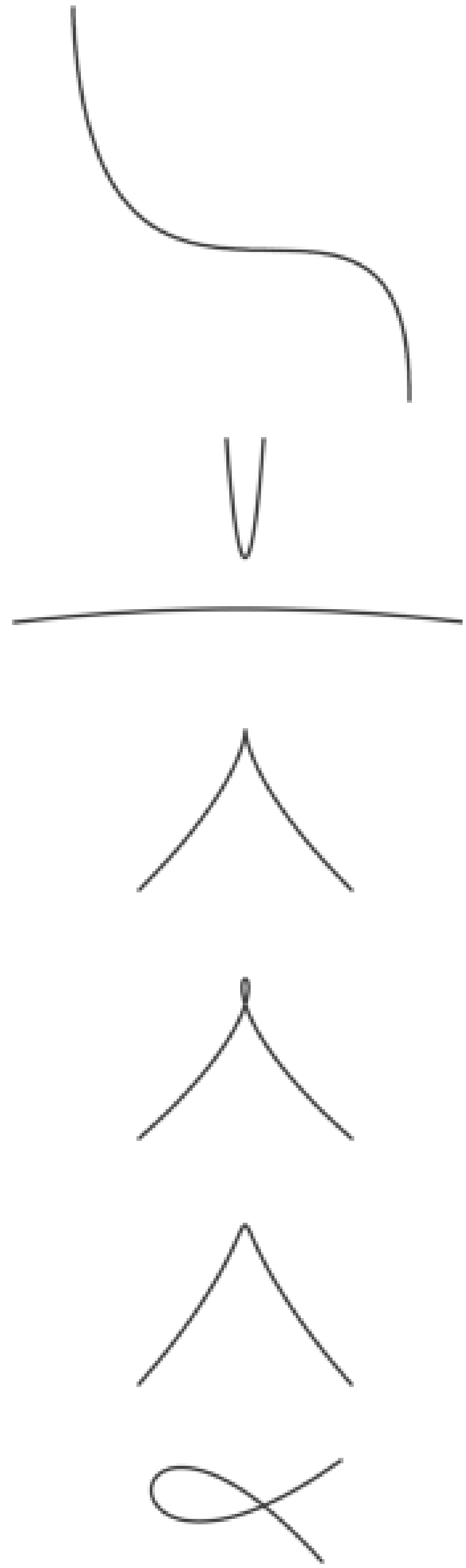

Plate 2: Anti-aliased Test Curves

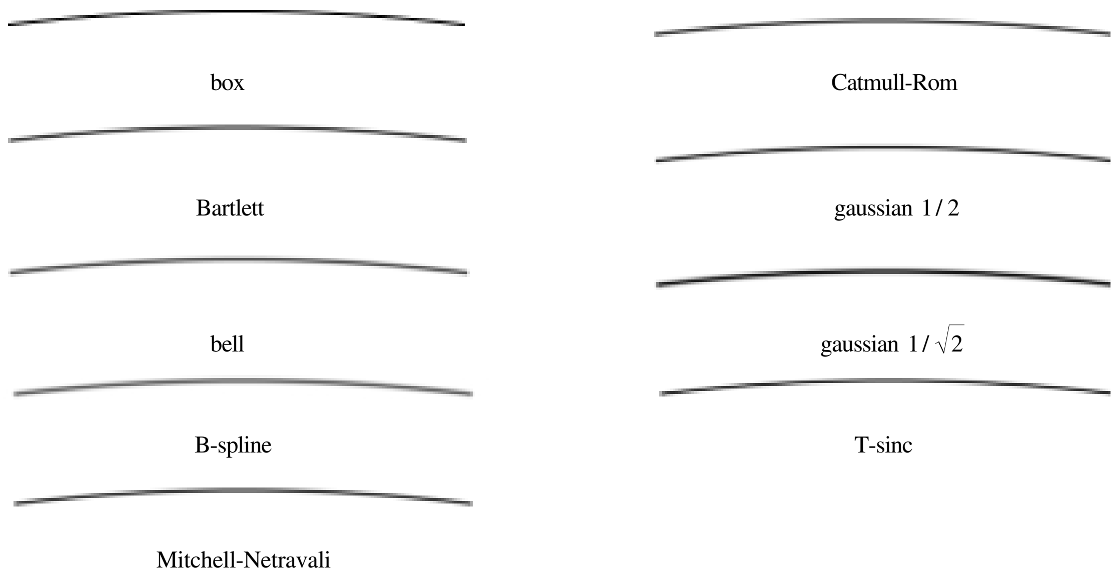

Plate 3: Parabola rendered with a variety of filters.

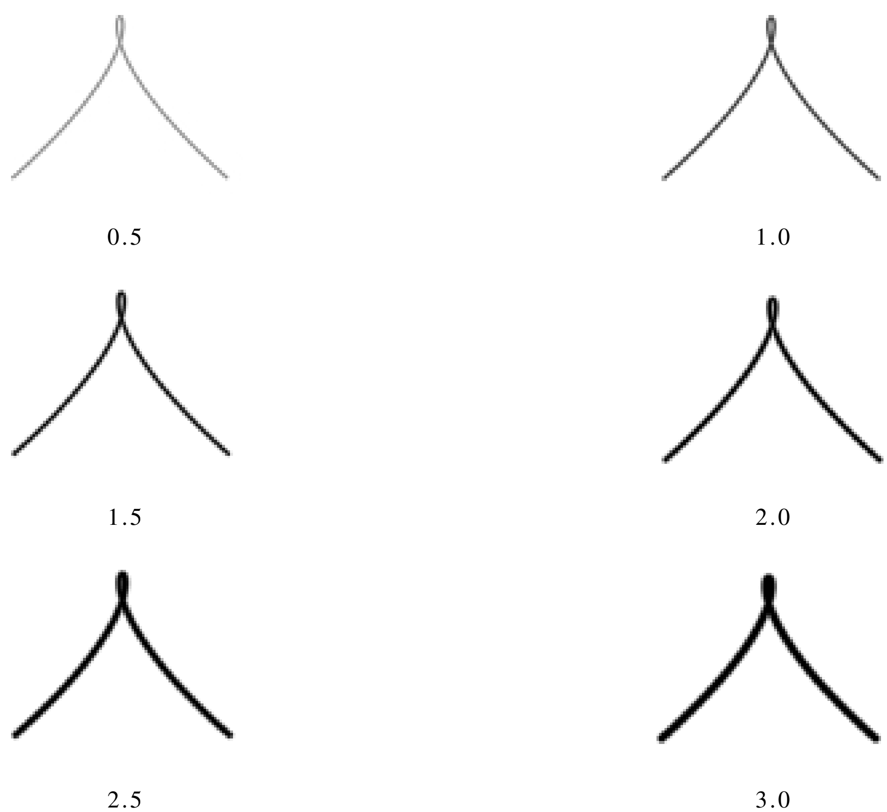

Plate 4: Varying curve width (in pixel units) for a cubic with a small loop near a cusp.