1993

Curve Fitting with Cubic A-Splines

Chandrajit L. Bajaj

Guoliang Xu

Report Number:
93-067

CURVE FITTING WITH CUBIC A-SPLINES

Chandrajit L. Bajaj
Guoliang Xu

CSD-TR-93-067
December 1993

Curve Fitting with Cubic A-Splines*

Chandrajit L. Bajaj Guoliang Xu†
Department of Computer Science
Purdue University
West Lafayette IN 47907
Tel: 317-494-6531
Fax: 317-494-0739
email: {bajaj,xuguo}@cs.purdue.edu

December 8, 1993

1 Introduction

Generating contours in image data, reconstructing digitized signals, and designing scalable fonts are only some of the several applications of spline curve fitting techniques. In this paper, we generalize past fitting schemes with conic splines [3, 13, 14, 15] and parametric splines [5, 10, 16], We exhibit efficient techniques to deal with cubic algebraic splines (A-splines) achieving fits with small number of pieces yet higher order of smoothness/continuity than prior schemes. The cubic A-splines are continuous chains of cubic implicitly defined algebraic curve segments, $f_i(x, y) = 0$, with $f_i(x, y)$ a bivariate real polynomial, and with continuity as high as C^3 at the junction points between curve segments.

The primary drawback for the widespread use of splines consisting of implicit algebraic curves is that a single implicitly defined curve may have several real components (ovals) and can possess several real singularities. In [2] we show how to isolate a non-singular and single sheeted segment of implicit algebraic curves and furthermore how to stitch these segments together to form splines. In this paper we focus on the cases of C^2 and C^3 cubic A-splines. We provide efficient algorithms for their use in fitting contour image data, ordered digital signal data, as well as randomly sampled scattered data sets. Note that parametric cubic splines can only achieve local C^2 continuity [6]. Additionally, we derive evaluation formulas for the efficient display of each of these cubic A-splines.

Related Prior Work:

Since 1960's, considerable work on polynomial spline interpolation and approximation has been done(see [6] for a bibliography). In general, spline interpolation has been viewed as a global fitting problem to scattered data[3, 5, 10, 13, 14, 15, 16]. Local interpolation by polynomials and rational functions is an old technique that traces back to Hermite and Cauchy[4]. However, local interpolation by the zero sets of piecewise polynomials (implicit algebraic curve segments)

*This work was supported in part by NSF grants CCR 92-22467, DMS 91-01424, AFOSR grant F49620-93-10138, NASA grant NAG-1-1473 and a gift from AT&T

†On leave from the Computing Center, Chinese Academy of Sciences, Beijing, 100080, China

is relatively new[2, 11, 12, 17]. The papers [11, 12] construct a family of C^1 (actually tangent continuous) and C^2 (actually curvature continuous) cubic algebraic splines. They use the following reduced form of the cubic $F(s,t,u) = as^2u+bsu^2-cst^2-dt^2u+estu$, with $a>0, b>0, c>0, d>0$, and (s,t,u) in Bernstein-Bezier (BB) coordinates over a triangle and guarantee that the segment of the curve inside the triangle is convex. Furthermore, their family of curvature continuous curves [11] can achieve C^2 continuity only if the given data is convex. Their results are a special case of the present paper, as our cubic A-splines are based on the general implicit cubic, and as we show can always be made to achieve C^3-continuity for arbitrary data, and even C^4-continuity for certain special input data.

2 Notation and Mathematical Preliminaries

Let $f(x,y)$ be a bivariate polynomial of degree n with real coefficients, and $p_i = (x_i, y_i)^T$, $i = 0,1,2$ be three affine independent points in the $xy-$plane(see Figure 2.1). Then the transform

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x_2 & x_1 & x_0 \\ y_2 & y_1 & y_0 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \end{bmatrix} \tag{2.1}$$

maps $f(x,y)$ into its barycentric coordinate form $g(\alpha_0,\alpha_1,\alpha_2) = f(x,y)$, where $0 \le \alpha_0 \le 1$, $0 \le \alpha_1 \le 1$, and $\alpha_0+\alpha_1+\alpha_2 = 1$, Note that $g(\alpha_0,\alpha_1,\alpha_2)$ can also be written as $F(\alpha_0,\alpha_1) = g(\alpha_0,\alpha_1,1-\alpha_0-\alpha_1)$, where (x,y) and (α_0,α_1) are related by

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x_2-x_0 & x_1-x_0 \\ y_2-y_0 & y_1-y_0 \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \end{bmatrix} + \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} \tag{2.2}$$

The inverse transform of (2.2) is

$$\begin{bmatrix} \alpha_0 \\ \alpha_1 \end{bmatrix} = \begin{bmatrix} y_1-y_0 & x_0-x_1 \\ y_0-y_2 & x_2-x_0 \end{bmatrix} \begin{bmatrix} x-x_0 \\ y-y_0 \end{bmatrix} / \Delta(p_0,p_1,p_2), \tag{2.3}$$

where $\Delta(p_0,p_1,p_2) = \det \begin{bmatrix} p_2 & p_1 & p_0 \\ 1 & 1 & 1 \end{bmatrix}$. In the barycentric coordinate system, $g(\alpha_0,\alpha_1,\alpha_2)$ can be expressed as the BB form (see [7, 9]).

$$g(\alpha_0,\alpha_1,\alpha_2) = \sum_{j=0}^{n}\sum_{i=0}^{n-j} b_{ij} \frac{n!}{i!j!(n-i-j)!} \alpha_0^i \alpha_1^j \alpha_2^{n-i-j}.$$

In this paper, we consider C^k continuity between adjacent algebraic curve segments. Each algebraic curve segment $f(x,y) = 0$ can be expressed locally at (non-singular) junction points in functional form as either $x = x(y)$ or $y = y(x)$. C^k continuity at the junction points is then achieved by the matching of derivatives upto order k. Relative to the triangle $p_0p_1p_2$, we use two local coordinates denoted as $(\mathcal{X},\mathcal{Y})_{(p_0,p_1)}$ and $(\mathcal{X},\mathcal{Y})_{(p_2,p_1)}$, and which are defined by shifting the origin of the xy-system to p_0 and p_2 respectively, and then rotating them in such a way that the vectors p_1-p_0 (resp. p_1-p_2) are in the same direction as the new y-axis(see Figure 2.1).

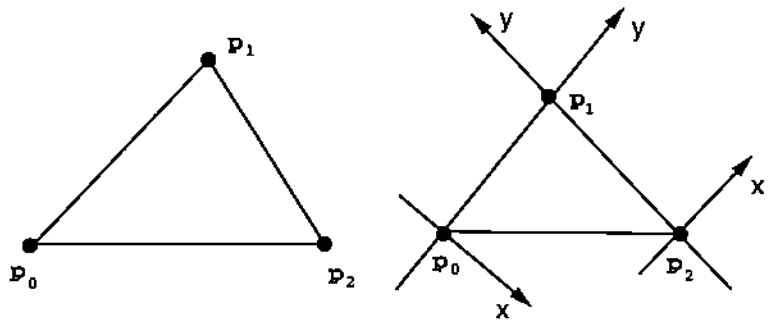

Figure 2.1: The triangle and related local systems

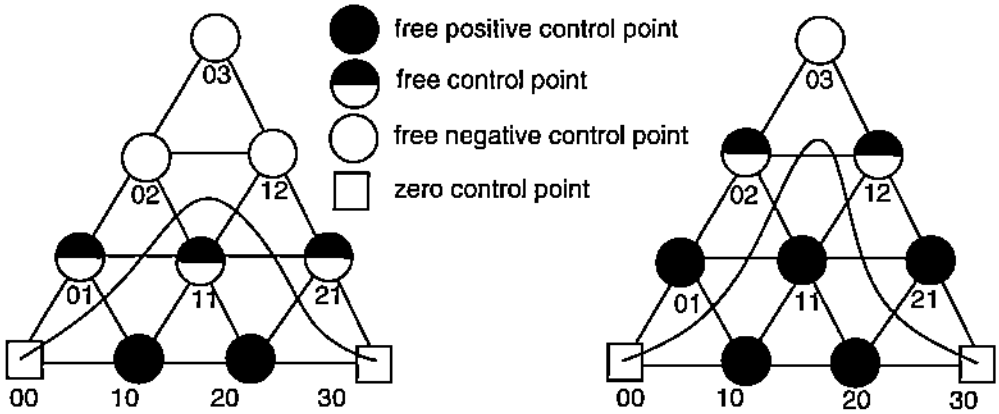

Figure 3.1: Bernstein Bezier Coefficients of a C^0 Cubic Algebraic Curve

3 A-Splines

We first state some results of arbitrary degree algebraic splines [2] and then specialize to the cases of C^2 and C^3 cubic A-splines, the focus of this paper. Let

$$F(\alpha_0, \alpha_1) = \sum_{j=0}^{n} \sum_{i=0}^{n-j} b_{ij} \frac{n!}{i!j!(n-i-j)!} \alpha_0^i \alpha_1^j (1-\alpha_0-\alpha_1)^{n-i-j}. \tag{3.1}$$

be the BB form(see [7]) of $F(\alpha_0, \alpha_1)$. Since there is constant multiplier to the equation $F(\alpha_0, \alpha_1) = 0$. We may assume $b_{0n} = -1$ if $b_{0n} \neq 0$.

Theorem 3.1 [2] *For the given polynomial $F(\alpha_0, \alpha_1)$ defined as (3.1), if there exists an integer $k(0 < k < n)$ such that*

$$b_{ij} \geq 0, \quad \text{for} \quad i = 0, 1, ...n-j; \quad j = 0, 1, ..., k-1 \tag{3.2}$$

$$b_{ij} \leq 0 \quad \text{for} \quad i = 0, 1, ...n-j; \quad j = k+1, ..., n \tag{3.3}$$

and $\sum_{i=0}^{n} b_{i0} > 0$, $\sum_{i=0}^{n-j} b_{ij} < 0$ for at least one j $(k < j \leq n)$, then

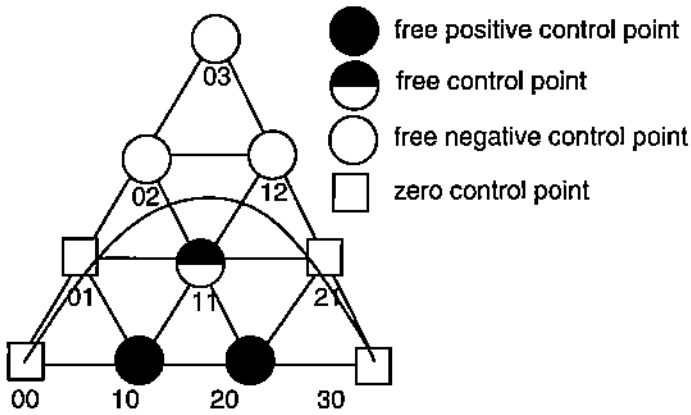

Figure 3.2: Bernstein Bezier Coefficients of a C^1 Cubic Algebraic Curve

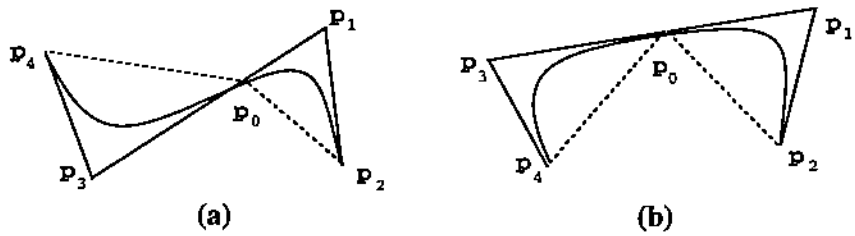

Figure 3.3: The two different cases of C^1 join polygon segments

(i) for any β that $0 < \beta < 1$, the straight line $\alpha_0 = \beta(1-\alpha_1)$, that pass through P_1 and the line segment (P_0, P_2), intersect the curve $F(\alpha_0, \alpha_1) = 0$ one and only one time(counting multiplicity) in the interior of the triangle $P_0P_1P_2$.

(ii) The value α_1 determined by $B_\beta(\alpha_1) = F(\beta(1-\alpha_1), \alpha_1) = 0$ in the interior of the triangle is an analytic function of β.

Theorem 3.1 guarantees that there is one and only one segment of $F(\alpha_0, \alpha_1) = 0$ within the standard triangle. The term "A spline" then denotes a chain of such curve segments with some continuity at the joining (junction) points. We should mention that the curve $F(\alpha_0, \alpha_1) = 0$ passes through $P_1 = (1, 0)^T$ if $b_{0n} = 0$. However, we do not use this part of the curve. In our applications in §4 using cubic A-splines, we take b_{03} to be -1. Figure 3.1 shows the two different possible cases of cubic A-splines which are C^0 at the two base end points of the triangle. (using Theorem 3.1 for $n = 3$ and $k = 0$). In each case there are 7 remaining free degrees of freedom (reduced from the original 9) which can be used for interpolating and least-squares approximation of additional data points in the interior of the triangle.

Figure 3.2 shows the case of a cubic A-spline which is C^1 at the two base end points of the triangle. (using Theorem 3.1 for $n = 3$ and $k = 1$). In this case there are 5 remaining free degrees of freedom $(b_{10}, b_{20}, b_{11}, b_{02}, b_{12})$ which can be used for interpolating and least-squares approximation of additional data points in the interior of the triangle.

Two such cubic curves defined over triangles $p_0\widehat{p_1}p_2$ and $p_4\widehat{p_3}p_0$ can be simply joined with C^1 continuity by either of the two polygon configurations as shown in Figure 3.3.

4 Fitting with Cubic A-splines

Our fitting algorithm with C^2 and C^3 cubic A-splines is as follows.

Algorithm 1 *1. Extract a contour (ordered set of points) from the given input data. See subsection 4.1.*

2. Compute breakpoints along the contour. These breakpoints points are the junction points for the cubic curves which make up the cubic A-spline. See subsection 4.2.

3. Compute derivatives at the junction points using local divided differences along the contour. For C^2 and C^3 continuity one needs upto second and third order derivatives, respectively, at these junction points. See subsection 4.3.

4. Construct cubic A-spline fits which interpolate the junction points along with the derivatives, and is least-squares approximate from all the given data between junction points. See subsection 4.4.

4.1 Extracting a Contour of Points

For contour extraction from dense image data we follow the following algorithm. The dense image data is in the form of a two dimensional array of two byte integers, one array for each planar slice through the object. The value in each cell (pixel) of the array is related to the density of the scanned object at that point in space. Each array may contain any number of contours, i.e., each slice may cut the scanned object in multiple places. To locate the contours :

1. scan for a cell on an initial edge,
2. starting at this cell hug the exterior of the cross section working from cell to cell and creating a list of two dimensional points until the beginning is reached or a dead end is found,
3. if a dead end is found backtrack,
4. if the path closes and the algorithm does not backtrack to the beginning point then smooth and compress the list of points if necessary.

In our implementation of this algorithm the following heuristic rule was used: if the density value in a cell c is within range and if the density values of all the cells surrounding c are within range, then the cell c is *acceptable*. The point list is smoothed and compressed by growing segments that are within a prescribed constant value of the original polyline. An example contour extraction is shown in the left part of Figure 4.4 from the input MRI (Magnetic Resonance Imaging) image on the right.

For arbitrarily scattered we use the alpha shape generation algorithm of [8] to extract an appropriate contour of the given scattered data points.

Examples of this algorithm are shown in Figure 4.5 for an initially unordered set of point data sampled from a human head profile.

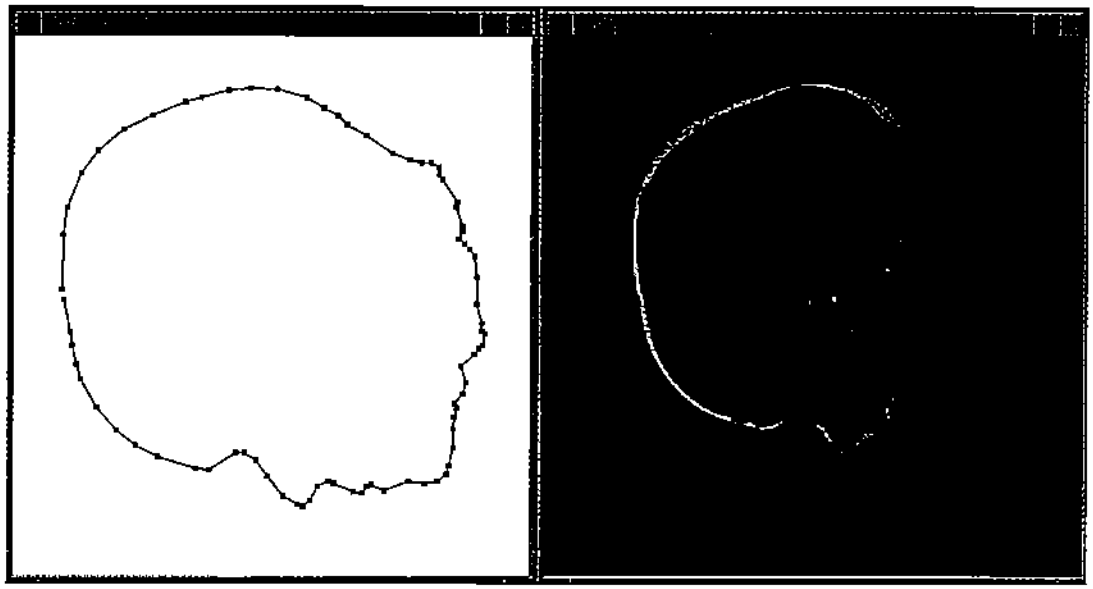

Figure 4.4: Extracting a contour from dense image data

4.2 Computation of Junction Points

The next step is to compute the junction points around the contour. A curvature adaptive scheme for the placement of the cubic curve segments is as follows. The points on a unit circle are in one-to-one correspondence with the normal directions, (or alternatively the slopes) of the line segments which make up the polygonal contour. Consider any regular k polygonal subdivision of a circle and number the k discrete normal directions n of the polygon boundary with integers from 1 to k. See also Figure 4.6.

Now number each line segment of the contour boundary with the integer i if it has the largest dot product of its normal with the i^{th} normal of the regular polygon. Under this mapping the k discrete normal directions on the circle partitions the polygonal contour on a data slice into groups where the members of a group consist of a connected sequence of line segments having the same assigned number. The endpoints of groups are the contour (junction) points whose two incident line segments have distinct assigned numbers. The line segments of each group are then replaced by a single cubic which C^2 or C^3-interpolates the group endpoints and the locally computed derivatives and simultaneously least-squares approximates the contour line segments that originally formed the group and lies within the junction points. See Figure 4.7 where junction points are computed for different polygonal subdivisions k of the unit circle. The C^2 and C^3 interpolation of the pair of endpoints and locally computed derivatives, by cubic A-splines are explained in the next subsections. If the least-squares approximation yields a poor error bound then additional cubics can be used to achieve a better bound. This operation is of course local to the the group and can be achieved by selectively refining the regular polygon edge corresponding to that group, replacing that edge by two or more edges inscribed in the circular arc subtended by that edge. The newly created normal directions are now mapped to the polygonal contour splitting the group into sub-groups. Each sub-group can now be replaced by a cubic, improving the approximation.

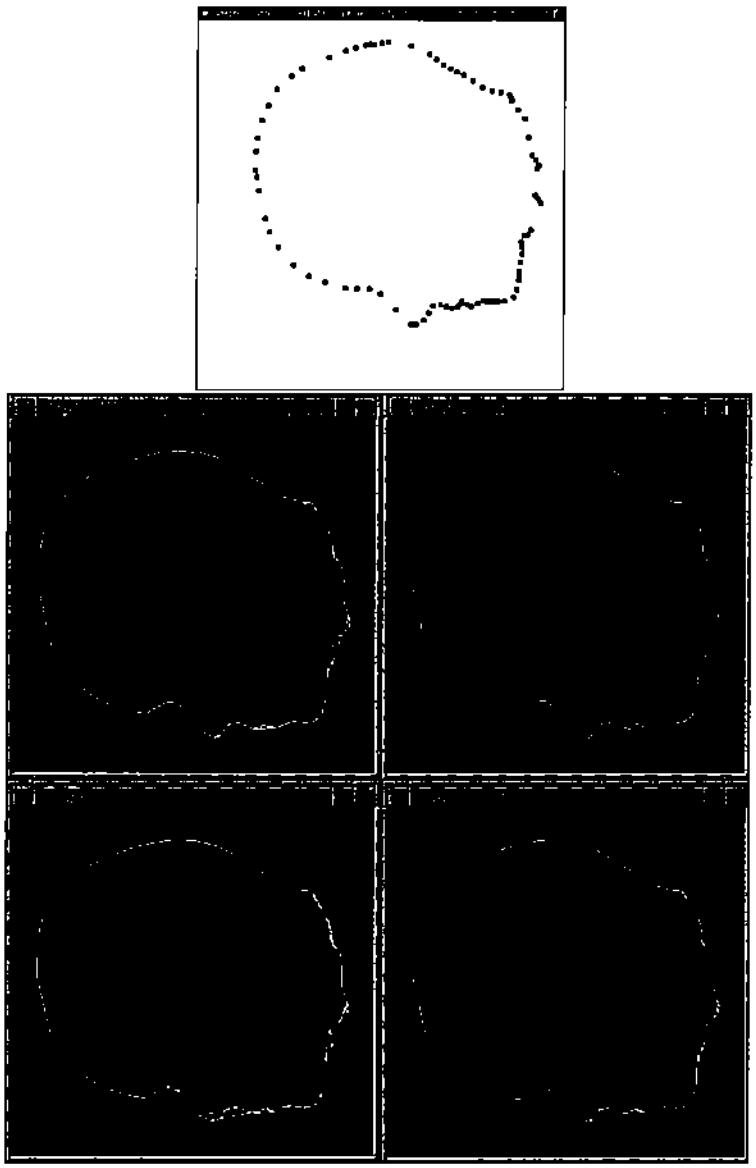

Figure 4.5: Extracting a contour from scattered data points

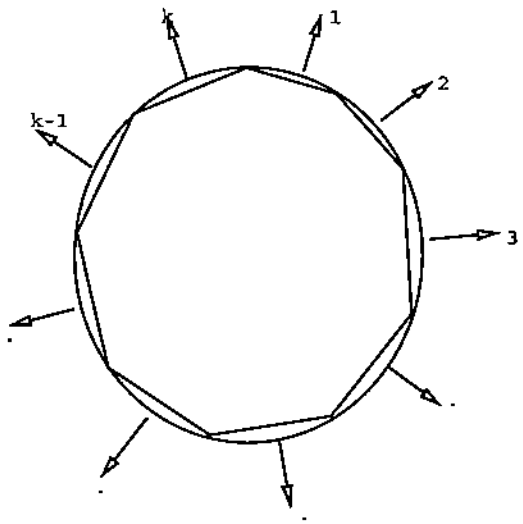

Figure 4.6: A regular subdivision of the space of normals on a planar contour

4.3 Generating Derivatives at Junction Points

There are various forms of divided-difference methods that extract geometric information around a junction point, from a given list of points [6]. Consider a sequence of points $\cdots, p_{i-2}, p_{i-1}, p_i, p_{i+1}, p_{i+2}, \cdots$ around the junction point p_i and an imaginary power series $C(t)$ from which, we assume, the digitized points near p_i arise, and whose parameter value is $t = 0$ for p_i. Then, the tangent vector of $C(t)$ at $t = 0$ can be approximated by the approximation :

$$\begin{aligned} C'(0) &\approx \frac{\sigma_i}{\mathsf{dist}(p_i, p_{i+1})}(p_{i+1} - p_i) \\ &+ \frac{1-\sigma_i}{\mathsf{dist}(p_{i-1}, p_i)}(p_i - p_{i-1}) \end{aligned}$$

where $\sigma_i = \frac{\mathsf{dist}(p_{i-1}, p_i)}{\mathsf{dist}(p_i, p_{i+1}) + \mathsf{dist}(p_{i-1}, p_i)}$ and $\mathsf{dist}(*, *)$ is the distance between two points.

Repeatedly applying this approximation formula, yields compact formulas for higher order divided-differences :

$$\Delta^j p_i = \begin{cases} p_i & \text{if } j = 0 \\ \frac{1}{j}(\frac{\sigma_i}{\mathsf{dist}(p_i, p_{i+1})}(p_{i+1} - p_i) & \\ + \frac{1-\sigma_i}{\mathsf{dist}(p_{i-1}, p_i)}(p_i - p_{i-1})) & \text{if } j > 0 \end{cases}$$

Using this divide-difference operator, a truncated power series is represented as $C_i(t) = \Delta^0 p_i + \Delta^1 p_i t + \Delta^2 p_i t^2 + \cdots + \Delta^k p_i t^k$. The higher order derivatives are the junction points are then approximated by $C'(0)$, $C''(0)$, $C'''(0)$, etc.

4.4 Exact and Least-Squares Fitting with C^2 and C^3 cubic A-splines

We shall use transform (2.1)–(2.3). Consider a C^1 cubic algebraic curve segment defined over a triangle $p_0 p_1 p_2$(see for e.g. Fig. 3.2).

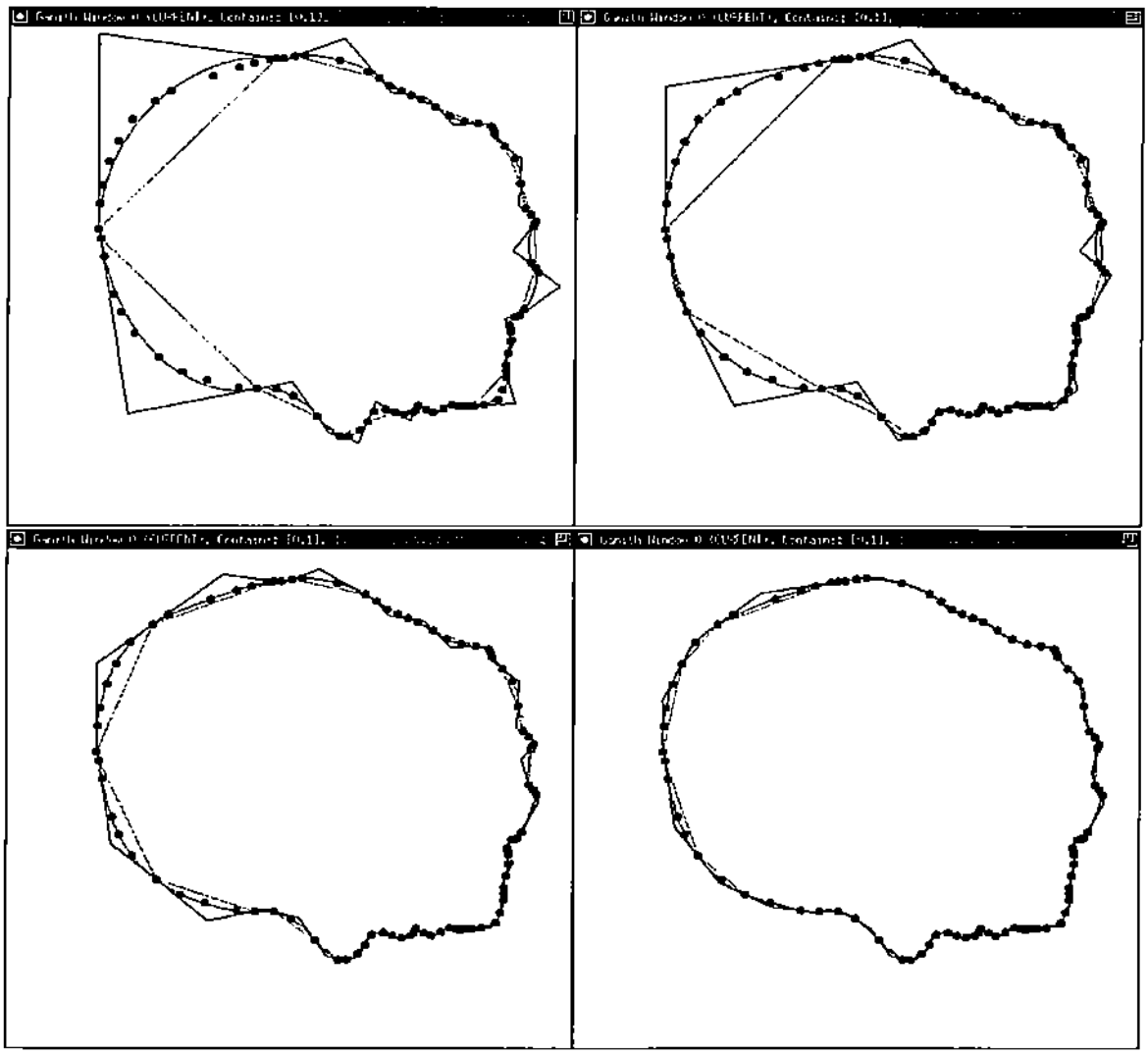

Figure 4.7: Junction Points and Cubic A-spline Fits

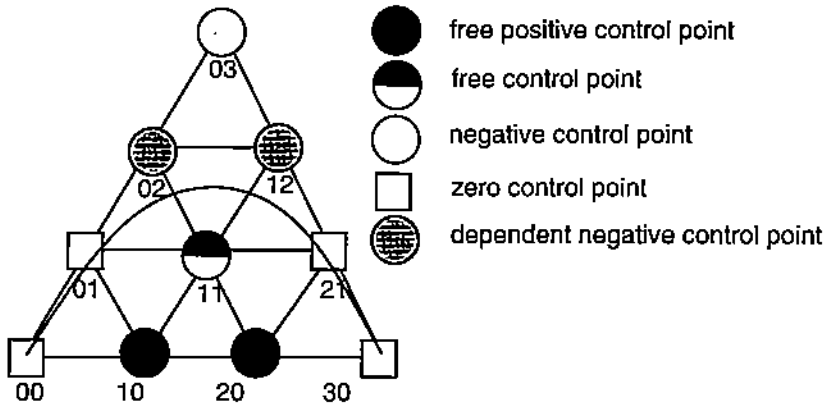

Figure 4.8: Bernstein Bezier Coefficients of a C^2 Cubic Algebraic Curve

$$\begin{aligned} F(\alpha_0,\alpha_1) &= -\alpha_1^3 + b_{10}\alpha_0(1-\alpha_0-\alpha_1)^2 + b_{20}\alpha_0^2(1-\alpha_0-\alpha_1) \\ &\quad + b_{02}\alpha_1^2(1-\alpha_0-\alpha_1) + b_{12}\alpha_0\alpha_1^2 + b_{11}\alpha_0\alpha_1(1-\alpha_0-\alpha_1) \end{aligned} \tag{4.4}$$

with

$$b_{10} > 0, \quad b_{20} > 0, \quad b_{02} \le 0, \quad b_{12} \le 0 \tag{4.5}$$

We first compute $\alpha_{0i}^{(k)} = \alpha_{0i}^{(k)}(0)$ by differentiating $F(\alpha_0,\alpha_1) = 0$ about α_1 as follows:

$$\alpha_{00}^{(1)} = 0 \qquad \alpha_{02}^{(1)} = -1$$

$$\frac{\alpha_{00}^{(2)}}{2!} = -\frac{b_{02}}{b_{10}}, \qquad \frac{\alpha_{02}^{(2)}}{2!} = \frac{b_{12}}{b_{20}}, \tag{4.6}$$

$$\frac{\alpha_{00}^{(3)}}{3!} = \frac{b_{10} - b_{10}b_{02} + b_{11}b_{02}}{b_{10}^2}, \qquad \frac{\alpha_{02}^{(3)}}{3!} = \frac{-b_{20} + b_{20}b_{12} - b_{11}b_{12}}{b_{20}^2} \tag{4.7}$$

$$\frac{\alpha_{00}^{(4)}}{4!} = -\frac{b_{11} - 2b_{10}}{b_{10}^2} \ (\text{if } b_{02} = 0), \quad \frac{\alpha_{02}^{(4)}}{4!} = -\frac{2b_{20} - b_{11}}{b_{20}^2} \ (\text{if } b_{12} = 0) \tag{4.8}$$

4.4.1 C^2 Continuity

First, we require from (4.5) and (4.6) that

$$\alpha_{00}^{(2)} \ge 0, \qquad \alpha_{02}^{(2)} \le 0 \tag{4.9}$$

Then for the local coordinate system we need to have

$$\mathcal{X}_{(p_0,p_1)}^{(2)}\Delta(p_0,p_1,p_2) \ge 0, \quad \mathcal{X}_{(p_2,p_1)}^{(2)}\Delta(p_0,p_1,p_2) \le 0 \tag{4.10}$$

That is, for each segment of the polygon where two curve segments join, we have two inequalities. We must show that these inequalities are consistent as both are related to the same second derivative value. We discuss the consistency problem by considering two join configurations of $p_4\widehat{p_3p_0}$ and

$p_0\widehat{p_1}p_2$ of Figure 3.3. We call these a Case(a)-join and a Case(b)-join, respectively. Note that for a Case(a)-join $n^T(p_2-p_0)\cdot n^T(p_4-p_0)<0$; and for a Case(b)-join $n^T(p_2-p_0)\cdot n^T(p_4-p_0)>0$.

I. If p_0, the join point of $p_4\widehat{p_3}p_0$ and $p_0\widehat{p_1}p_2$, is a Case(a)-join, then (4.5), (4.6) require that

$$-\mathcal{X}^{(2)}_{(p_0,p_1)}\Delta(p_0,p_3,p_4)\geq 0 \tag{4.11}$$

If $\mathcal{X}^{(2)}_{(p_0,p_1)}\neq 0$, then by (4.10) and (4.11) we need to have $\Delta(p_0,p_1,p_2)\Delta(p_0,p_3,p_4)\leq 0$. Unfortunately, this is not true (see (4.15) in the following). Since

$$p_3-p_0=\beta(p_1-p_0),\quad \beta<0 \tag{4.12}$$

and

$$\begin{aligned}\Delta(p_0,p_3,p_4) &= (x_4-x_0)(y_3-y_0)-(x_3-x_0)(y_4-y_0)\\ &= \beta n^T(p_4-p_0)\end{aligned} \tag{4.13}$$

$$\begin{aligned}\Delta(p_0,p_1,p_2) &= (x_2-x_0)(y_1-y_0)-(x_1-x_0)(y_2-y_0)\\ &= n^T(p_2-p_0)\end{aligned} \tag{4.14}$$

we have

$$\Delta(p_0,p_3,p_4)\Delta(p_0,p_1,p_2)=\beta n^T(p_4-p_0)\cdot n^T(p_2-p_0)>0 \tag{4.15}$$

Therefore (4.10) and (4.11) hold iff $\mathcal{X}^{(2)}_{(p_0,p_1)}=0$.

II. If p_0 is of a Case(b)-join, then (4.5), (4.6) require that $-\mathcal{X}^{(2)}_{(p_0,p_1)}\Delta(p_4,p_3,p_0)\leq 0$. Then by (4.10) we need to have $\Delta(p_0,p_1,p_2)\Delta(p_4,p_3,p_0)\geq 0$. Since $\Delta(p_4,p_3,p_0)=-\Delta(p_0,p_3,p_4)$, we have by (4.13), (4.14) that

$$\Delta(p_4,p_3,p_0)\Delta(p_0,p_1,p_2)>0 \tag{4.16}$$

Algorithm 2 *1. Let $\{q_i\widehat{v_iq_{i+1}}\}_{i=0}^m$ form a C^1 polygonal contour of the junction points. Specify the second derivative values such that $\mathcal{X}^{(2)}_{(q_i,v_i)}=0$ if q_i is of a Case(a)-join, or $\mathcal{X}^{(2)}_{(q_i,v_i)}\Delta(q_i,v_i,q_{i+1})\geq 0$ if q_i is of a Case(b)-join for $i=1,2,\ldots,m$, and $\mathcal{X}^{(2)}_{(q_0,v_0)}\Delta(q_0,v_0,q_{0+1})\geq 0$, $\mathcal{X}^{(2)}_{(q_{m+1},v_m)}\Delta(q_m,v_m,q_{m+1})\leq 0$.*

2. Next b_{02} and b_{12} are determined by (4.6).

3. The three remaining degrees of freedom $b_{10}>0$, $b_{20}>0$ and b_{11}, are used to least-squares approximate the given data within the triangle.

4.4.2 C^3 Continuity

For cubics already satisfying a C^2 join condition, now we go further and show how to achieve C^3 continuity locally. Let $p_4\widehat{p_3}p_0$ and $p_0\widehat{p_1}p_2$ be two adjacent segments of the polygon that C^1 join at p_0 (see Figure 3.3).

A. If p_0 is of a Case(a)-join, then by Theorem 3.1 we must have $\mathcal{X}^{(2)}_{(p_0,p_1)}=0$. From (4.7), we have

$$\alpha^{(3)}_{00}=\frac{6}{b_{10}},\qquad \mathcal{X}^{(3)}_{(p_0,p_1)}=\alpha^{(3)}_{00}\frac{\Delta(p_0,p_1,p_2)}{\|p_1-p_0\|^4}$$

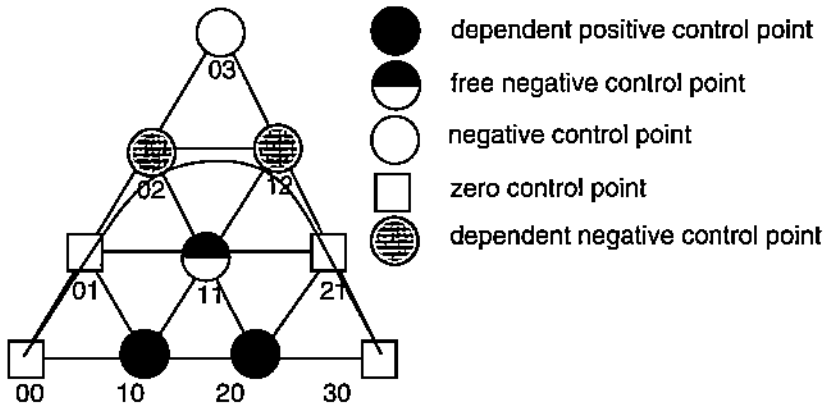

Figure 4.9: Bernstein Bezier Coefficients of a C^3 Cubic Algebraic Curve

This requires that

$$\mathcal{X}^{(3)}_{(p_0,p_1)}\Delta(p_0,p_1,p_2) > 0 \tag{4.17}$$

The other inequality related to the point p_0 is $\mathcal{X}^{(3)}_{(p_0,p_3)}\Delta(p_0,p_3,p_4) > 0$. That is, we need to have $\Delta(p_0,p_1,p_2)\Delta(p_0,p_3,p_4) > 0$ because $\mathcal{X}^{(3)}_{(p_0,p_1)} = \mathcal{X}^{(3)}_{(p_0,p_3)}$. This of course is true(see (4.15)). Therefore C^3 continuity is guaranteed by setting $\mathcal{X}^{(3)}_{(p_0,p_1)}$ as (4.17) in this case.

B. If p_0 is of a Case(b)-join, we first show that it is impossible to achieve C^3 by taking $\mathcal{X}^{(2)}_{(p_0,p_1)} = 0$. Because in this case, we require from (4.7) that $\mathcal{X}^{(3)}_{(p_0,p_3)}\Delta(p_4,p_3,p_0) < 0$ in addition to (4.17). Hence $\Delta(p_0,p_1,p_2)\Delta(p_4,p_3,p_0) < 0$ a contradiction to (4.16). Now suppose $\mathcal{X}^{(2)}_{(p_0,p_1)} \neq 0$, i.e., by (4.6),

$$\mathcal{X}^{(2)}_{(p_0,p_1)}\Delta(p_0,p_1,p_2) > 0 \tag{4.18}$$

then by (4.16), $-\mathcal{X}^{(2)}_{(p_0,p_1)}\Delta(p_4,p_3,p_0) < 0$. Hence C^2 continuity is guaranteed at p_0 from the discussion in the previous subsection. Now we take $\mathcal{X}^{(3)}_{(p_0,p_1)}$ such that b_{10} and b_{20} are positive. From (4.6) and (4.7), we have

$$\left(\frac{\alpha^{(3)}_{00}}{6} - \frac{\alpha^{(2)}_{00}}{2}\right) b_{10} = 1 - b_{11}\frac{\alpha^{(2)}_{00}}{2}, \quad \left(\frac{\alpha^{(3)}_{02}}{6} - \frac{\alpha^{(2)}_{02}}{2}\right) b_{20} = -1 - b_{11}\frac{\alpha^{(2)}_{02}}{2} \tag{4.19}$$

In order to make b_{10} and b_{20} positive, we have the following inequalities

$$\pm\begin{cases} 1 - b_{11}\frac{\alpha^{(2)}_{00}}{2} > 0 \\ \frac{\alpha^{(3)}_{00}}{6} - \frac{\alpha^{(2)}_{00}}{2} > 0, \end{cases} \qquad \pm\begin{cases} -1 - b_{11}\frac{\alpha^{(2)}_{02}}{2} > 0 \\ \frac{\alpha^{(3)}_{02}}{6} - \frac{\alpha^{(2)}_{02}}{2} > 0 \end{cases} \tag{4.20}$$

Therefore, on the triangle $p_0p_1p_2$ and at point p_0, we have the inequalities.

$$\pm\begin{cases} 1 - \frac{\|p_1-p_0\|^3 b_{11} D_2}{\Delta(p_0,p_1,p_2)} > 0 \\ \frac{\|p_1-p_0\|^4 D_3}{\Delta(p_0,p_1,p_2)} + 2\left(\frac{D_2}{\Delta(p_0,p_1,p_2)}\right)^2 \|p_1-p_0\|^4\langle p_1-p_0, p_2-p_0\rangle - \frac{\|p_1-p_0\|^3 D_2}{\Delta(p_0,p_1,p_2)} > 0 \end{cases} \tag{4.21}$$

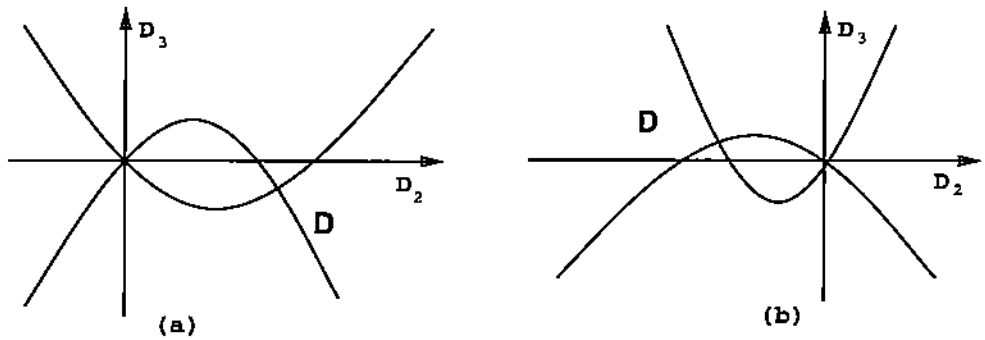

Figure 4.10: The feasible domain D

where $D_k = \frac{\chi^{(k)}_{(p_0,p_1)}}{k!}$, and at p_2

$$\pm \begin{cases} -1 - \frac{\|p_1-p_2\|^3 b_{11} D_2}{\Delta(p_0,p_1,p_2)} > 0 \\ \frac{\|p_1-p_2\|^4 D_3}{\Delta(p_0,p_1,p_2)} + 2\left(\frac{D_2}{\Delta(p_0,p_1,p_2)}\right)^2 \|p_1-p_2\|^4 \langle p_1-p_2, p_2-p_0\rangle - \frac{\|p_1-p_2\|^3 D_2}{\Delta(p_0,p_1,p_2)} > 0 \end{cases} \tag{4.22}$$

where $D_k = \frac{\chi^{(k)}_{(p_2,p_1)}}{k!}$. For $p_4\widehat{p_3p_0}$ at p_0, we have

$$\pm \begin{cases} -1 + \frac{\|p_3-p_0\|^3 b_{11} D_2}{\Delta(p_4,p_3,p_0)} > 0 \\ \frac{\|p_3-p_0\|^4 D_3}{\Delta(p_4,p_3,p_0)} + 2\left(\frac{D_2}{\Delta(p_4,p_3,p_0)}\right)^2 \|p_3-p_0\|^4 \langle p_3-p_0, p_0-p_4\rangle + \frac{\|p_3-p_0\|^3 D_2}{\Delta(p_4,p_3,p_0)} > 0 \end{cases} \tag{4.23}$$

For each segment of polygon, there is one parameter b_{11} to be determined. This parameter can be used to interpolate some point in the least square sense within the triangle, or to achieve C^1 continuity.

If we take $b_{11} \leq 0$, then by (4.18), we know that the first inequality of +(4.21)(take + sign) and the first inequality of −(4.23) (take − sign) are always true. Hence the feasible domain for D_2 and D_3 is determined at p_0 by

$$sign\ \Delta(p_0,p_1,p_2) \begin{cases} D_3 - a_1 D_2^2 - b_1 D_2 > 0 \\ D_3 - a_2 D_2^2 + b_2 D_2 < 0 \end{cases}$$

where $a_1 = \frac{2\langle p_1-p_0, p_0-p_2\rangle}{\Delta(p_0,p_1,p_2)}$, $b_1 = \frac{1}{\|p_1-p_0\|}$, $a_2 = \frac{2\langle p_3-p_0, p_4-p_0\rangle}{\Delta(p_4,p_3,p_0)}$, $b_2 = \frac{1}{\|p_3-p_0\|}$. Let $\mathbf{D} = \mathbf{D}(p_0,p_1,p_2,p_3,p_4)$ be the feasible domain for D_2 and D_3 which depends on p_0, p_1, p_2, p_3 and p_4. Then only for

$$\langle p_1-p_0, p_2-p_0\rangle > 0, \quad \langle p_3-p_0, p_4-p_0\rangle > 0$$

$\mathbf{D}$ is a non-empty set. For $\Delta(p_0,p_1,p_2) > 0$(hence $D_2 > 0$) and $\Delta(p_0,p_1,p_2) < 0$(hence $D_2 < 0$), the feasible domains $\mathbf{D}$ for D_2 and D_3 are shown as Fig. 4.10.a and Fig. 4.10.b, respectively.

Algorithm 3 *1. Let $\{q_i\widehat{v_iq_{i+1}}\}_{i=0}^m$ form a C^1 polygon of the junction points and assume $\langle v_i - q_i, q_{i+1} - q_i\rangle > 0$, $\langle v_{i-1} - q_i, q_{i-1} - q_i\rangle > 0$ if $1 \leq i \leq m$. At each junction point q_i $(i = 0, 1, \ldots, m+1)$, satisfy the second and third order derivatives as follows(regard q_i, v_i, q_{i+1} as p_0, p_1, p_2 for $i \geq 0$ and q_{i-1}, v_{i-1}, q_i as p_4, p_3, p_0 for $i \leq m+1$):*

(a) $\mathcal{X}^{(2)}_{(q_i,v_i)} = 0$, $\mathcal{X}^{(3)}_{(q_i,v_i)}$ *satisfy (4.17) if* q_i *is of a Case(a)-join and* $1 \leq i \leq m$.

(b) $\mathcal{X}^{(2)}_{(q_i,v_i)}\Delta(q_i, v_i, q_{i+1}) > 0$, $\mathcal{X}^{(2)}_{(q_i,v_i)}$ *and* $\mathcal{X}^{(3)}_{(q_i,v_i)}$ *satisfy both +(4.21) and −(4.23) i.e.,* $(D_2, D_3) \in \mathbb{D}(q_i, v_i, q_{i+1}, v_{i-1}, q_{i-1})$ *if* q_i *is of a Case(b)-join and* $1 \leq i \leq m$.

(c) *For* $i = 0$ *and* $i = m+1$, $\mathcal{X}^{(2)}_{(q_0,v_0)}\Delta(q_0, v_0, q_{0+1}) \geq 0$, $\mathcal{X}^{(2)}_{(q_{m+1},v_m)}\Delta(q_m, v_m, q_{m+1}) \leq 0$, *and* $\mathcal{X}^{(3)}_{(q_0,v_0)}$ *and* $\mathcal{X}^{(3)}_{(q_{m+1},v_m)}$ *satisfy +(4.21) and −(4.22), respectively.*

2. *Then* b_{10} *and* b_{20} *are determined by (4.19);* b_{02} *and* b_{12} *are determined by (4.6) and the remaining single degree of freedom* $b_{11} \leq 0$ *is chosen by least-squares approximation of the given data points interior to the triangle.*

5 Examples

Figure 5.11 shows several examples of contour model reconstructions using C^2 and C^3 cubic A-splines using the algorithms of the previous sections. All implementations were made in our X-11 windows based algebraic curves and surfaces toolkit called GANITH [1].

References

[1] C. Bajaj and A. Royappa. *The GANITH Algebraic Geometry Toolkit.* In Proceedings of the First International Symposium on the Design and Implementation of Symbolic Computation Systems, pages 268–269. Springer-Verlag, *Lecture Notes in Computer Science No. 429*, 1990.

[2] C. Bajaj and G. Xu. *A-Splines: Local Interpolation and Approximation using* C^k*-Continuous Piecewise Real Algebraic Curves.* Computer Science Technical Report, CAPO-92-44, Purdue University, 1992.

[3] F. L. Bookstein. Fitting Conic Sections to Scattered Data. *Computer Graphics and Image Processing*, 9(00):56–71, 1979.

[4] M.A.L. Cauchy. *Cours d'analyse de l'Ecole Royale Polytechnique.* premiére, L'Imprimerie Royale, Paris, 1821.

[5] H.B. Curry and I.J. Schoenberg. On pólya frequency functions iv: the fundamental spline functions and their limits. *J. d'Analyse Math.*, 17:71–107, 1966.

[6] C. deBoor. *A Practical Guide to Splines.* Springer Verlag, 1978.

[7] C. deBoor. B-form basics. In G. Farin, editor, *Geometric Modeling: Algorithms and New Trends*, pages 347–366. SIAM, 1987.

[8] H. Edelsbrunner, D.G. Kirkpatrick, and H. Seidel. On the shape of a set of points in the plane. *IEEE Trans. on Information Theory*, 29:4:551–559, 1983.

[9] G. Farin. *Curves and Surfaces for Computer Aided Geometric Design: A Practical Guide.* Academic Press Inc.,, 1990.

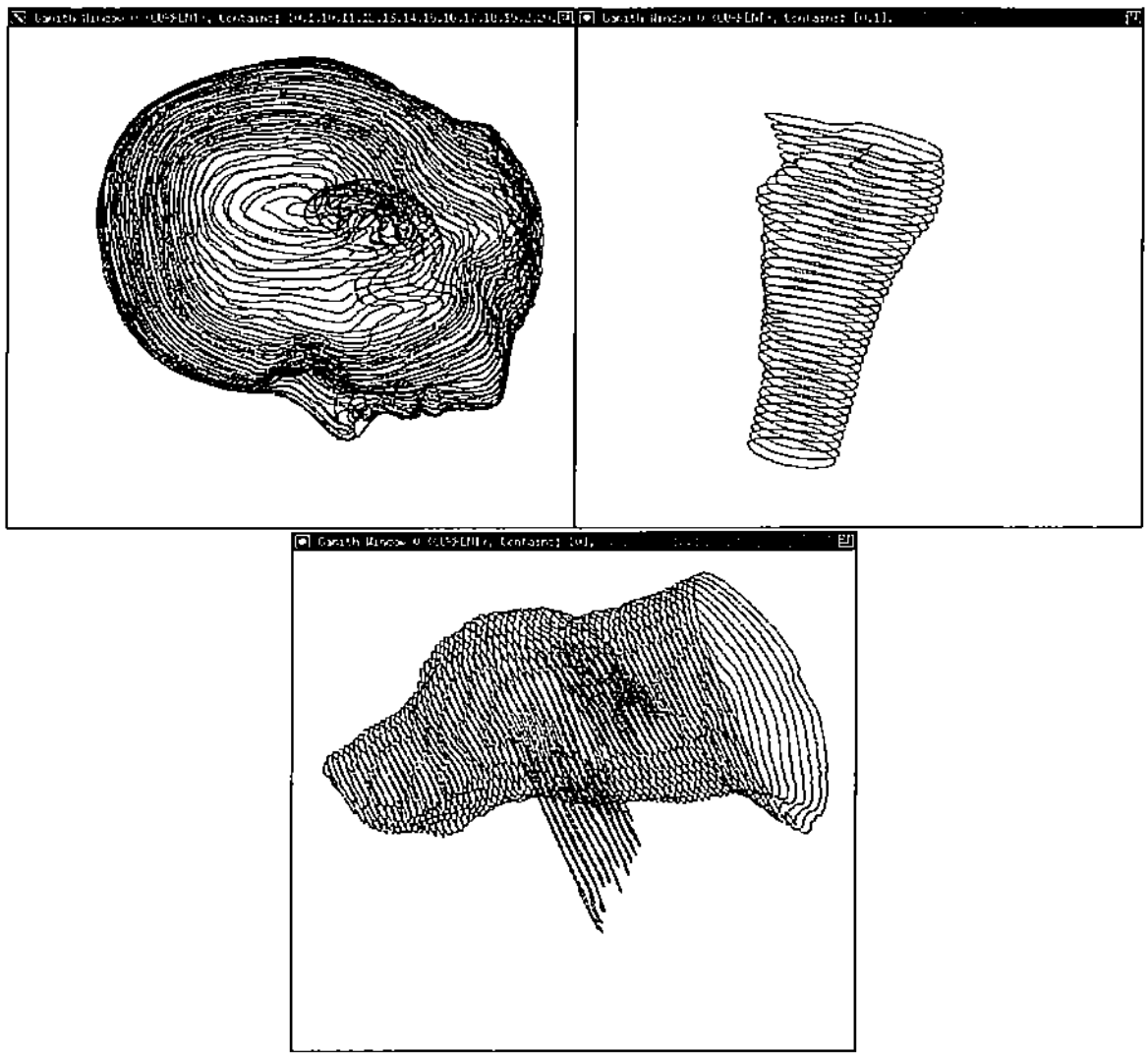

Figure 5.11: Fitting with C^2 and C^3 Cubic A-splines

[10] Sommer M. Nürnberger G., Schumarker L.L. and Strauβ H. Generalized tchebycheffian splines. *SIAM J. Math. Anal.*, 15:790–804, 1984.

[11] M. Paluszny and R. R. Patterson. A family of curvature continuous cubic algebraic splines. In Curves and Surfaces in Computer Vision and Graphics III, SPIE, pages 48–57, 1992.

[12] M. Paluszny and R. R. Patterson. A family of tangent continuous algebraic splines. *ACM Transaction on Graphics*, 12,3:209–232, 1993.

[13] T Pavlidis. Curve fitting with conic splines. *ACM Trans. on Graphics*, 2:1:1–13, 1983.

[14] V. Pratt. Techniques for conic splines. *Computer Graphics*, 19(3):151–159, 1985.

[15] P. Sampson. Fitting Conic Sections to Very Scattered Data: An Iterative Refinement of the Bookstein Algorithm. *Computer Graphics and Image Processing*, 18(00):97–108, 1982.

[16] I.J. Schoenberg and Whitney A. On pólya frequency functions iii. the positivity of translation determinants with applications to the interpolation problem by spline curves. *Trans. Amer. Math. Soc.*, 74:246–259, 1953.

[17] T.W. Sederberg. Planar piecewise algebraic curves. *Computer Aided Geometric Design*, 1(3):241–255, 1984.