Data Fitting with Cubic A-Splines*

Chandrajit L. Bajaj Guoliang Xu†
Department of Computer Science
Purdue University

Abstract

We present algorithms for constructing iso-contours from image data or fitting scattered point data C^1, C^2 or C^3 piecewise smooth chains of single sheeted real cubic algebraic curve segments called cubic A-splines (short for cubic algebraic splines). Using cubic A-splines we achieve data fitting with either a higher order of continuity or greater local flexibility for fixed continuity, than numerous prior schemes.

Key Words. *iso-contours, scattered points, curve fitting, algebraic splines, cubic*

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 [4, 16, 17, 18] and even rational parametric splines [7, 13, 19], 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 or greater local flexibility for fixed 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 achievable local 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 [3] 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 case of 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 rational parametric cubic splines can only achieve local C^2 continuity [8], compared to the local C^3 continuity of cubic A-splines. The class of rational parametric cubic curves is a strict subset of the class of cubic algebraic curves [21] and also has fewer degrees of freedom (8 versus 9 of the cubic algebraic curve). Note, of course that for fixed continuity (C^k, $k = 0, 1, 2, or 3$), the extra local degrees of freedom which the cubic A-spline segment posseses, allows for greater local flexibility and approximation of the input data.

Related Prior Work:

Since 1960's, considerable work on polynomial spline interpolation and approximation has been done(see [8] for a bibliography). In general, spline interpolation has been viewed as a global fitting problem to

*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

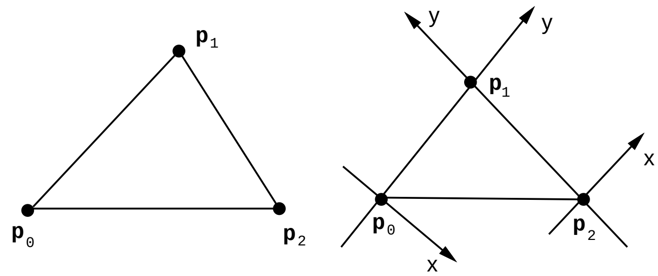

Figure 2.1: The BB triangle and related local systems

scattered data[4, 7, 13, 16, 17, 18, 19]. Local interpolation by polynomials and rational functions is an old technique that traces back to Hermite and Cauchy[6]. However, local interpolation by the zero sets of piecewise polynomials (implicit algebraic curve segments) is relatively new[3, 14, 15, 20]. The papers [14, 15] construct a family of C^1 (actually tangent continuous) and C^2 (actually curvature continuous) cubic algebraic splines. They however use a reduced form of the cubic which guarantees that each segment of the spline is convex and furthermore allows one to achieve C^2 continuity only if the input data is convex. Furthermore, their family of curvature continuous curves [14] 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[3].

2 Cubic A-Splines

Since the case of C^1 smooth cubic A-splines is dealt with in [15], here we consider C^k continuity between adjacent cubic algebraic curve segments for the cases of $k = 2and3$. 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, for $k = 2and3$, at the junction points is then achieved by the matching of derivatives upto order k. Relative to a given 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).

Let

$$F(\alpha_0,\alpha_1) = \sum_{j=0}^{n}\sum_{i=0}^{n-j} b_{ij}\frac{n!}{i!j!(n-i-j)!}\alpha_0^i\alpha_1^j(1-\alpha_0-\alpha_1)^{n-i-j}. \tag{2.1}$$

be the BB form(see [9, 11]) of $f(x,y)$ over the triangle $p_0p_1p_2$. Here we study only cubic algebraic curves and so restrict to $n = 3$. Let $\mathcal{F}_0(\mathcal{X},\mathcal{Y})$ and $\mathcal{F}_2(\mathcal{X},\mathcal{Y})$ be $f(x,y)$ under the local coordinates $(\mathcal{X},\mathcal{Y})_{(p_0,p_1)}$ and $(\mathcal{X},\mathcal{Y})_{(p_2,p_1)}$, respectively, and let $\mathcal{F}_i(0,0) = 0$ and $\frac{\partial\mathcal{F}_i(0,0)}{\partial\mathcal{X}} \neq 0$ for $i = 0,2$. Then each $\mathcal{F}_i(\mathcal{X},\mathcal{Y}) = 0$ can be expressed locally at p_i as a function of $\mathcal{Y}$, denoted $\mathcal{X}_{(p_i,p_1)}(\mathcal{Y})$. The k-th local derivatives, denoted by $\mathcal{X}^{(k)}_{(p_i,p_1)} = \mathcal{X}^{(k)}_{(p_i,p_1)}(0)$, are also well defined. Here the subscript (p_i,p_1) is to emphasize that $\mathcal{X}$ is related to the local system $(\mathcal{X},\mathcal{Y})_{(p_i,p_1)}$ that is defined by p_i and p_1. Correspondingly, $\alpha_0 = \alpha_{0i}(\alpha_1)$ is well defined locally by $F(\alpha_0,\alpha_1) = 0$ at P_i for $i = 0,2$, here $P_0 = (0,0)^T$, $P_1 = (0,1)^T$ and $P_2 = (1,0)^T$. So do the derivatives $\alpha^{(k)}_{0i} = \alpha^{(k)}_{0i}(0)$ at P_i for $i = 0,2$. Here the subscript i is to emphasize the dependency on point P_i. Now, suppose $\mathcal{X}^{(1)}_{(p_i,p_1)} = 0$, that is, the curves $\mathcal{F}_i(\mathcal{X},\mathcal{Y})$ are tangent with y-axis(this is always the case

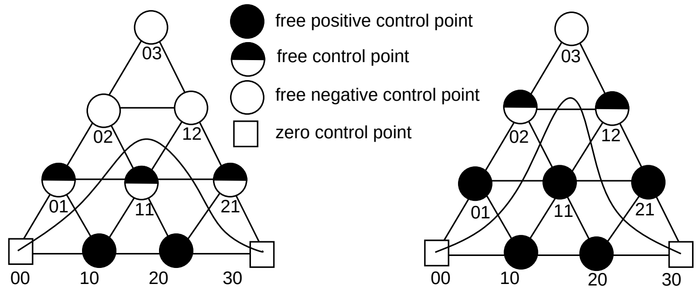

Figure 2.2: Bernstein Bezier Coefficients of a C^0 Cubic Algebraic Curve

in this paper), then we can establish, by differentiating $\mathcal{X} = \mathcal{X}_{(p_i,p_1)}(\mathcal{Y})$, the following relations among these derivatives: $\alpha_{00}^{(1)} = 0$, $\alpha_{02}^{(1)} = -1$ and

$$\alpha_{0i}^{(2)} = \frac{\|p_1 - p_i\|^3 \mathcal{X}_{(p_i,p_1)}^{(2)}}{\Delta(p_0, p_1, p_2)} \tag{2.2}$$

$$\alpha_{0i}^{(3)} = \frac{\|p_1 - p_i\|^4 \mathcal{X}_{(p_i,p_1)}^{(3)}}{\Delta(p_0, p_1, p_2)} + \frac{3\|p_1 - p_i\|^4 \langle p_1 - p_i, p_2 - p_0\rangle (\mathcal{X}_{(p_i,p_1)}^{(2)})^2}{\Delta^2(p_0, p_1, p_2)} \tag{2.3}$$

where $\Delta(p_0, p_1, p_2) = \det \begin{bmatrix} p_2 & p_1 & p_0 \\ 1 & 1 & 1 \end{bmatrix}$.

We first state some results of arbitrary degree algebraic splines [3] and then specialize to the cases of C^2 and C^3 cubic A-splines, the focus of this paper. Let $F(\alpha_0, \alpha_1)$ be defined as (2.1). 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 [3] *For the given polynomial $F(\alpha_0, \alpha_1)$ defined as (2.1), if there exists an integer $k (0 < k < n)$ such that*

$$b_{ij} \geq 0, \quad \text{for} \quad i = 0, 1, ...n - j; \quad j = 0, 1, ..., k - 1 \tag{2.4}$$

$$b_{ij} \leq 0 \quad \text{for} \quad i = 0, 1, ...n - j; \quad j = k + 1, ..., n \tag{2.5}$$

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

(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 $P_1 = (0, 1)^T$ if $b_{0n} = 0$. However, we do not use this part of the curve. In our applications in §3 using cubic A-splines, we take b_{03} to be -1. Figure 2.2 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 $n = 3$, $k = 1$ and $k = 2$). In each case there are 7 remaining free degrees of freedom (reduced from the original 9). Figure 2.3 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

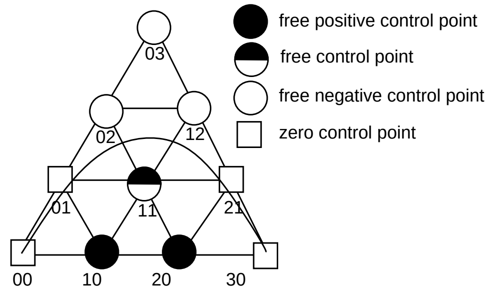

Figure 2.3: Bernstein Bezier Coefficients of a C^1 Cubic Algebraic Curve

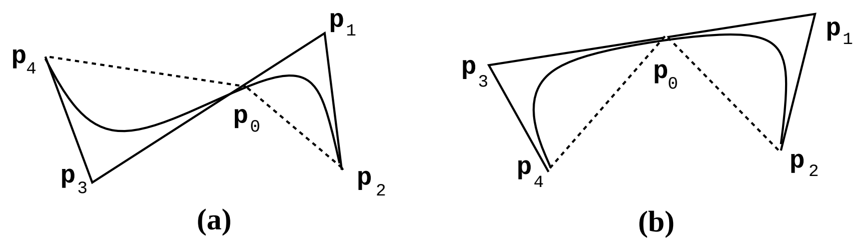

Figure 2.4: The two different cases of C^1 join polygon segments

$k = 1$). In this case there are 5 remaining free degrees of freedom ($b_{10}, b_{20}, b_{11}, b_{02}, b_{12}$). These free degrees of freedom can be used for interpolating and least-squares approximation of additional data points in the interior of the triangle.

Two cubic A-spline 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 2.4. These two polygon configurations are known as a *Case(a)-join* and a *Case(b)-join*.

3 Fitting with Cubic A-splines

Our data fitting algorithms with C^2 and C^3 cubic A-splines are as follows.

Algorithm 1.

1. *Extract a contour (ordered set of points) from the given input data. See subsection 3.1.*

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 3.2.*

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 3.3.*

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 3.4.*

3.1 Extracting an Iso-Contour from a Grey-scale Image

For iso-contour extraction from dense image data we use 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

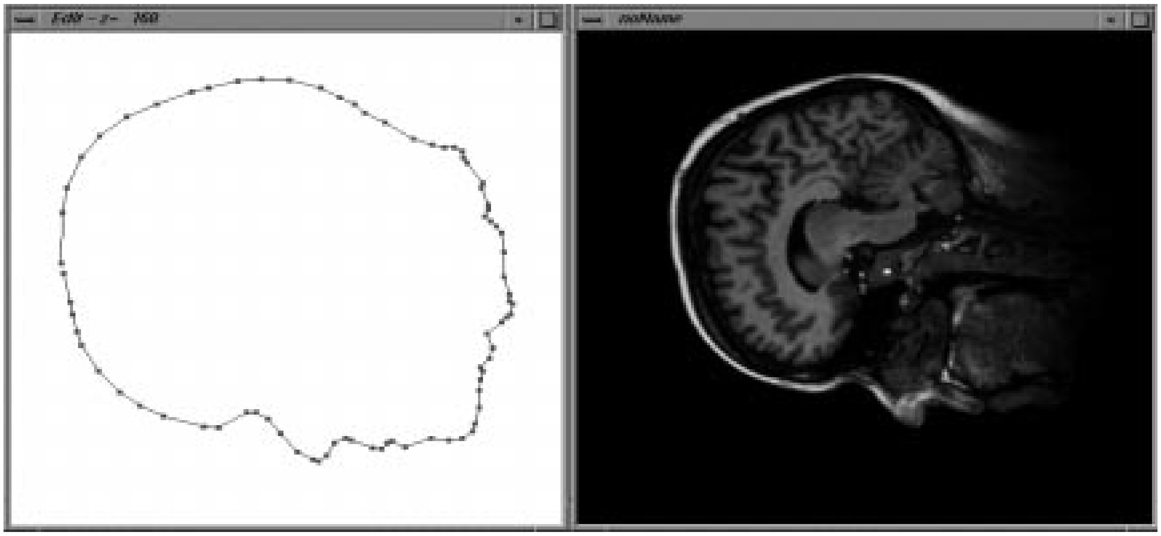

Figure 3.5: Extracting an iso-contour (left) from a dense MRI slice (right)

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 iso-contours. To locate the iso-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 3.5 from the input MRI (Magnetic Resonance Imaging) image slice on the right. Of course, more sophisticated iso-contour extraction algorithms may also be used, see for e.g. [12]

For arbitrarily scattered data we use the alpha shape generation algorithm of [10] to extract an appropriate contour of the given scattered data points. Examples of this algorithm are shown in Figure 3.6 for an initially unordered set of point data sampled from a human head profile.

3.2 Computation of Junction Points

The next step is to compute the junction points around the contour. We use a curvature adaptive scheme for the placement of the cubic curve segments that is given in [5]. 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 $\mathbf{n}$ of the polygon boundary with integers from 1 to k. See also Figure 3.7.

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

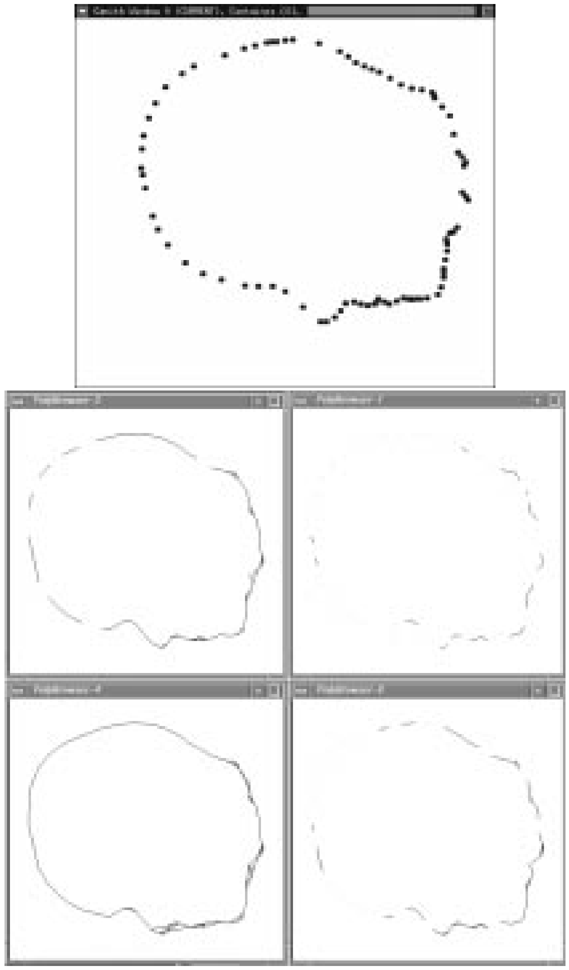

Figure 3.6: Extracting a contour from scattered data points

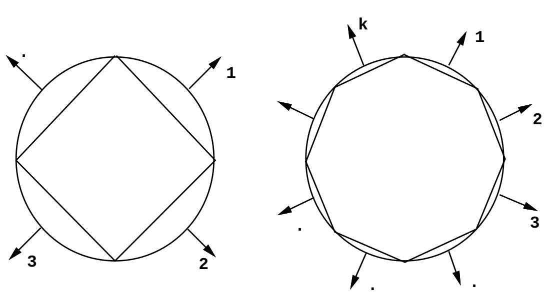

Figure 3.7: Regular subdivision of the space of normals on a planar contour

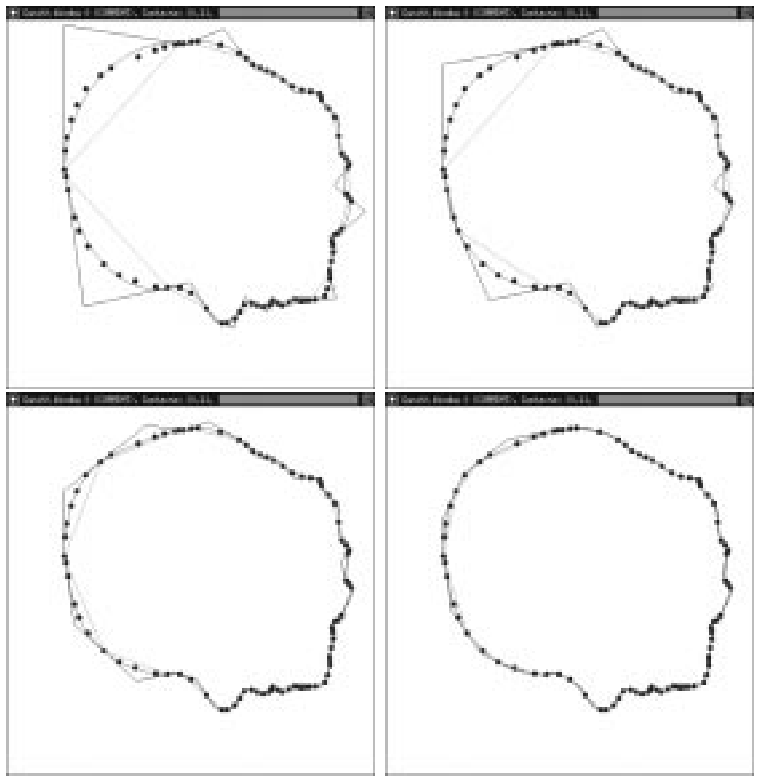

Figure 3.8: Junction Points and Cubic A-spline Fits

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 3.8 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.

3.3 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 [8]. 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:

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

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 [2] for higher order divided-differences:

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

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 at the junction points are then approximated by $C'(0)$, $C''(0)$, $C'''(0)$, etc. From these derivatives, we can easily compute the local derivatives $\mathcal{X}^{(k)}_{(p_i,p_{i+1})}$ and $\mathcal{X}^{(k)}_{(p_i,p_{i-1})}$ defined in §2.

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

Consider a C^1 cubic algebraic curve segment defined over a triangle $p_0p_1p_2$(see for e.g. Fig. 3.9)

$$\begin{aligned} F(\alpha_0, \alpha_1) \;=\; & -\alpha_1^3 + b_{10}\alpha_0(1-\alpha_0-\alpha_1)^2 + b_{20}\alpha_0^2(1-\alpha_0-\alpha_1) \\ & + b_{02}\alpha_1^2(1-\alpha_0-\alpha_1) + b_{12}\alpha_0\alpha_1^2 + b_{11}\alpha_0\alpha_1(1-\alpha_0-\alpha_1) \end{aligned} \tag{3.6}$$

with

$$b_{10} > 0, \quad b_{20} > 0, \quad b_{02} \le 0, \quad b_{12} \le 0 \tag{3.7}$$

By differentiating $F(\alpha_0, \alpha_1) = 0$ about α_1 we have the following formulas for $\alpha_{0i}^{(k)} = \alpha_{0i}^{(k)}(0)$:

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

$$\frac{\alpha_{00}^{(2)}}{2!} = -\frac{b_{02}}{b_{10}}, \qquad \frac{\alpha_{02}^{(2)}}{2!} = \frac{b_{12}}{b_{20}}, \tag{3.8}$$

$$\frac{\alpha_{00}^{(3)}}{3!} = \frac{b_{10} - b_{10}b_{02} + b_{11}b_{02}}{b_{10}^2}, \qquad \frac{\alpha_{02}^{(3)}}{3!} = \frac{-b_{20} + b_{20}b_{12} - b_{11}b_{12}}{b_{20}^2} \tag{3.9}$$

From these formulas and relations (2.2)–(2.3) and the sign requirement (3.7), we can derive the following algorithm for constructing C^2 continuous A-spline curve(see [3] for detail):

Algorithm 2. *Let $\{q_i\widehat{v_iq_i}_{+1}\}_{i=0}^m$ form a C^1 polygonal contour of the junction points(see Figure 3.10).*

1. *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 $\mathcal{X}^{(2)}_{(q_i,v_i)}\Delta(q_i, v_i, q_{i+1}) \ge 0$ if q_i is of a Case(b)-join for $i = 1, 2, \ldots, m$, and $\mathcal{X}^{(2)}_{(q_0,v_0)}\Delta(q_0, v_0, q_{0+1}) \ge 0$, $\mathcal{X}^{(2)}_{(q_{m+1},v_m)}\Delta(q_m, v_m, q_{m+1}) \le 0$.*
2. *Compute b_{02} and b_{12} by (3.8) and (2.2) for each triangle. Determine the three remaining degrees of freedom $b_{10} > 0$, $b_{20} > 0$ and b_{11} by least-squares approximation of the given data within the triangle, or via a default choice if there are not enough data points within the triangle.*

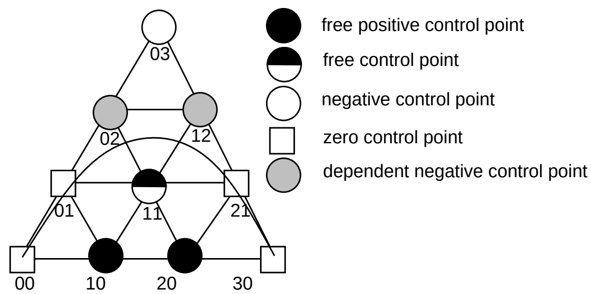

Figure 3.9: Bernstein Bezier Coefficients of a C^2 Cubic Algebraic Curve

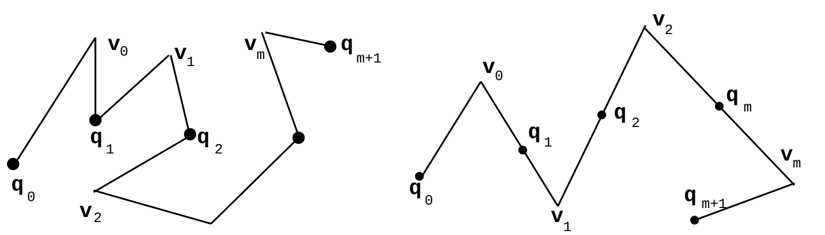

Figure 3.10: A C^0 polygon and a C^1 polygon

For achieving C^3 continuity, we specify the second and third local derivatives at the junction points. These derivatives need to satisfy some of the following conditions in order to have the coefficients of the BB-form have the required signs(see (3.7) and Figure 3.11):

$$\mathcal{X}^{(3)}_{(p_0,p_1)}\Delta(p_0,p_1,p_2) > 0 \tag{3.10}$$

$$\left(\frac{\alpha^{(3)}_{00}}{6} - \frac{\alpha^{(2)}_{00}}{2}\right) b_{10} = 1 - b_{11}\frac{\alpha^{(2)}_{00}}{2}, \quad \left(\frac{\alpha^{(3)}_{02}}{6} - \frac{\alpha^{(2)}_{02}}{2}\right) b_{20} = -1 - b_{11}\frac{\alpha^{(2)}_{02}}{2} \tag{3.11}$$

On the triangle $p_0p_1p_2$ and at point p_0 we have the inequalities.

$$\pm\left\{\begin{array}{l} 1 - \frac{||p_1-p_0||^3 b_{11} D_2}{\Delta(p_0,p_1,p_2)} > 0 \\ \frac{||p_1-p_0||^4 D_3}{\Delta(p_0,p_1,p_2)} + 2\left(\frac{D_2}{\Delta(p_0,p_1,p_2)}\right)^2 ||p_1-p_0||^4 \langle p_1-p_0, p_2-p_0\rangle - \frac{||p_1-p_0||^3 D_2}{\Delta(p_0,p_1,p_2)} > 0 \end{array}\right. \tag{3.12}$$

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

$$\pm\left\{\begin{array}{l} -1 - \frac{||p_1-p_2||^3 b_{11} D_2}{\Delta(p_0,p_1,p_2)} > 0 \\ \frac{||p_1-p_2||^4 D_3}{\Delta(p_0,p_1,p_2)} + 2\left(\frac{D_2}{\Delta(p_0,p_1,p_2)}\right)^2 ||p_1-p_2||^4 \langle p_1-p_2, p_2-p_0\rangle - \frac{||p_1-p_2||^3 D_2}{\Delta(p_0,p_1,p_2)} > 0 \end{array}\right. \tag{3.13}$$

where $D_k = \frac{\mathcal{X}^{(k)}_{(p_2,p_1)}}{k!}$. For $p_4\widehat{p_3}p_0$ at p_0, we have

$$\pm\left\{\begin{array}{l} -1 + \frac{||p_3-p_0||^3 b_{11} D_2}{\Delta(p_4,p_3,p_0)} > 0 \\ \frac{||p_3-p_0||^4 D_3}{\Delta(p_4,p_3,p_0)} + 2\left(\frac{D_2}{\Delta(p_4,p_3,p_0)}\right)^2 ||p_3-p_0||^4 \langle p_3-p_0, p_0-p_4\rangle + \frac{||p_3-p_0||^3 D_2}{\Delta(p_4,p_3,p_0)} > 0 \end{array}\right. \tag{3.14}$$

Algorithm 3. *Let* $\{q_i\widehat{v_i q_{i+1}}\}_{i=0}^m$ *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* $1 \le i \le m$.

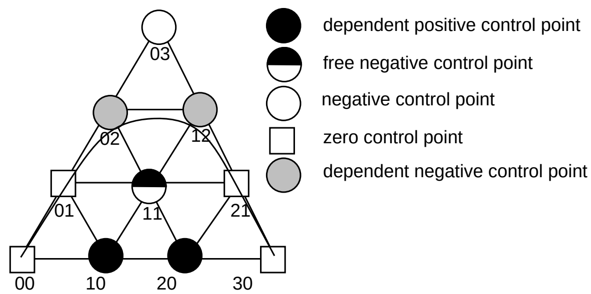

Figure 3.11: Bernstein Bezier Coefficients of a C^3 Cubic Algebraic Curve

1. *At each junction point q_i ($i = 0, 1, \ldots, m+1$), specify 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 (3.10) 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 +(3.12) and −(3.14) 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 +(3.12) and −(3.13), respectively.*
2. *For each triangle, compute b_{10} and b_{20} using (3.11); compute b_{02} and b_{12} using (3.8). 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 or via a default choice if there are not enough data points within the triangle.*

3.5 Surplus Degrees of Freedom

For the above C^2 and C^3 data fitting algorithms, after satisfying the derivatives at the junction points, there still exists three and one remaining degrees of freedom, respectively. These degrees of freedom can be used to locally control the shape of the curve in each triangle. For example, if b_{10} and b_{20} are given in Algorithm 2, then b_{11} can be chosen so that the curve in the triangle can be as high as the top vertex (when b_{11} tends to ∞) or as low as the bottom edge (when b_{11} tends to $-\infty$). In Algorithm 3, the curve can vary between the two limit curves(one corresponding to $b_{11} = 0$, other corresponding to $b_{11} = -\infty$). As we early mentioned in both Algorithm 2 and Algorithm 3, we currently use these degrees of freedom to least-squares approximate points in the triangle and based on the sign requirements (3.7). However, there exist the possibility that there are not enough data points within a triangle to determine these coefficients. Default choices of values for the undetermined coefficients are used in this case. One method used to determine these default values of the coefficients is to locally approximate a quadratic or linear curve with the triangle, which tends to avoid sharp changes in the geometry of the spline curve. The linear or quadratic approximation is easily achieved by using degree elevation formulas (see [11]). The second approach is to minimize the energy over the triangle on which the curve is defined. That is,

$$min = \int\!\!\int_{\Delta} \left(\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2 \right) dx\, dy$$

where $f(x, y) = 0$ is the curve in the triangle Δ in xy-system.

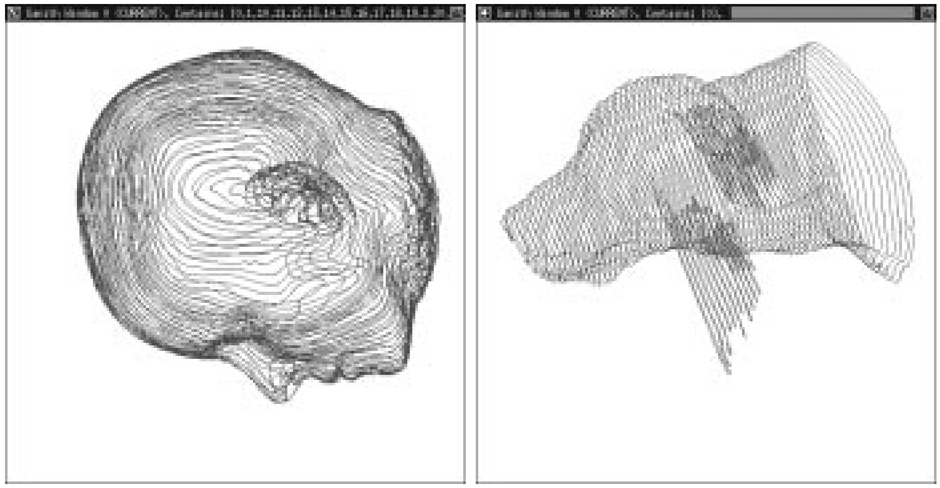

Figure 3.12: Iso-Contours from volume MRI using C^2 and C^3 Cubic A-splines

4 Conclusions and Examples

Cubic A-spline curves are an effective tool to fit both dense and scattered data. With a low degree cubic curve, one can achieve a high order of smoothness (C^2 or C^3), and still have remaining degrees of freedom to locally modify and control the shape of the curve. If C^1 smooth data fits are all that is required, using cubic A-splines one has as many as five remaining degrees of freedom for local shape control after satisfying the C^1 derivative constraints. These remaining or extra degrees of freedom can be used to both lower the approximation error as well as require fewer overall curve segments for the global data fit. Figure 3.12 shows examples of iso-contour reconstructions from volume MRI, using C^2 and C^3 cubic A-splines based on the algorithms of the previous sections. All implementations were made in our SHASTRA scientific design environment [1].

References

[1] V. Anupam and C. Bajaj. Collaborative Multimedia Scientific Design in SHASTRA. In *Proc. of the First ACM International Conference on Multimedia, ACM MULTIMEDIA 93*, pages 447–456. ACM Press, 1993.

[2] C. Bajaj and I. Ihm. Low degree approximation of surfaces of revolved objects. In *Proceedings of Graphics Interface'93*, pages 33–41, Toronto, Canada, 1993.

[3] 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.

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

[5] C. Bajaj. Electronic Skeletons: Modeling Skeletal Structures with Piecewise Algebraic Surfaces. In *Curves and Surfaces in Computer Vision and Graphics II*, pages 230–237, Boston, MA, 1991.

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

[7] 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.

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

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

[10] H. Edelsbrunner, D. Kirkpatrick, and R. Seidel. On the Shape of a Set of Points in the Plane. *IEEE Trans. on Information Theory*, 29:4:551–559, 1983.

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

[12] D. Fritsch. A medial description of greyscale image structure by gradient-limited diffusion. In Visualization in Biomedical Computing, pages 105–117, 1992.

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

[14] 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.

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

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

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

[18] 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.

[19] 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.

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

[21] R. Walker. *Algebraic Curves.* Springer Verlag, New York, 1978.