Active Contouring of Images with Physical A-splines*

Chandrajit L. Bajaj Valerio Pascucci
Department of Computer Sciences and TICAM
University of Texas
Austin, TX 78712

Robert J. Holt Arun N. Netravali
Bell Laboratories, Lucent Technologies
Murray Hill, NJ 07974

October 7, 1998

Abstract

A-splines are implicit real algebraic curves in Bernstein-Bézier (BB) form that are smooth. We use these in an algorithm for active contouring of images. One advantage of A-splines is that any change to the controlling weights only affects the curve locally, which results in fast convergence. Our active A-splines are also level sets of a time-dependent function with the added flexibility of a dynamic unstructured mesh. Other advantages include the ability to use lower degree polynomials than traditional polynomial parametric B-splines. A-splines also avoid the necessity of dealing with poles that can arise from rational parametric B-splines (NURBS), and they also allow an easy specification of orientation, so that they may be driven to converge to an interior or exterior contour. Our algorithm finds image contours by using a level-set method to obtain an initial close fitting polygon, constructing a physical A-spline contour by minimizing the image energy, and then minimizing the total energy by considering the energy in each spline segment individually.

1 Introduction

Many real images cannot be automatically segmented due to their complexity, low signal-to-noise ratio, undesired image features or other factors. Manual and semi-automatic tracking of images is still the dominant method for obtaining good segmentations from complicated images. Many image processing techniques find the low level features such as edge or region components in an image, and then build the global picture by assembling these components from the bottom up. Kass et al. [KWT88] suggest that higher-level information should be used in addition to the bottom up approach. They propose active contour models ('snakes') which allow the user to place an initial coarse contour of spline segments on an image. They define three kinds of energy acting on the contour. Thereafter the contour iteratively deforms itself to achieve a minimum total energy. The

*This work is supported in part by AFOSR grant F49620-97-1-0278, ONR grant N00014-97-1-0398, and a gift from Lucent Technologies

three kinds of energy are *internal spline* energy, *image* energy and *user constraint* energy. The internal spline energy tries to maintain a smooth contour. The image energy attracts contours to the desired image features. Image energies come in several forms. One is the image intensity itself, another is the edge energy, and a third is the energy of the terminations of line segments and corners. A coarse-to-fine approach by spatially smoothing image energy is entailed to reduce the possibility of trapping a contour in a local minimum. The user constraint energy can be interactively added by a user to move the contour from one local minimum into another minimum in case of unsatisfactory results. This energy model is also very attractive for a sequence of image slices in which the object does not change or move much.

A user defines some initial rough contours for the beginning slice, and thereafter the program finds accurate contours using the active contour model. These newly found contours will be used as the initial contours for the next slice. This scheme has been used by other papers in contour tracing [Coh91, CC93, Ron94].

This paper presents the following major contributions:

1. A-spline active contours: An algebraic spline (A-spline) [BX92, BX96, BCHN98] is used to represent the contour curve. One advantage of A-splines is that any change to the controlling weights only affects the curve locally. This results in fast convergence. Moreover, the degrees of freedom of an active linear contour polygon still remain in the control polygon vertices of the A-spline contour. Our active A-splines are also level sets of a time-dependent (piecewise quadratic or cubic) function with the added flexibility of a dynamic unstructured mesh (triangulation). Other advantages include the ability to use lower degree polynomials than traditional polynomial parametric B-splines (quadratic A-splines achieve local C^1 continuity with extra degrees of freedom; impossible with degree two polynomial parametric splines). A-splines also avoid the necessity of dealing with poles that can arise from rational parametric B-splines (NURBS).

2. Vector image forces: A-splines allow an easy specification of orientation, so that they may be driven to converge to an interior or exterior contour. Our active contour model shall have this polarity, dependent on whether the desired segmented contour is the inner or outer (or both) boundary of an annular region. The vector image force shall attract the active contour to the correct boundary.

The following terms will be used in this paper. An *A-spline segment* will be the portion of an A-spline between two consecutive marked points, called *junction points*, on the contour. The triangle formed by the two junction points and the point of intersection of the two tangent lines to the A-spline at the junction points will be called the *control triangle* of the A-spline segment, and the vertex at the intersection of the tangents lines is the *apex* of the control triangle. The polygon formed by connecting all the junction points is the *control polygon* of the A-spline.

The rest of this paper is as follows. Section 2 reports on prior work on active contour models. Section 3 gives background information on constructing image contours with A-splines. Section 4 provides an overall sketch of our algorithm that computes image for active contouring of images. Sections 5–7 provide the details of the three main steps of the active contouring algorithm. Section 5 presents an iterative coarsening algorithm for defining an A-spline control polygon from images using the degrees of freedom of the control vertices respecting intrinsic conditions derived from the image. Section 6 defines physical A-splines where material properties are assigned to the geometry yielding strain energy formulations for the curve. Image energy models are also provided. Section 7 presents an algorithm for energy minimization of physical A-splines for active image contouring utilizing the local degrees of freedom. Section 8 concludes the paper.

2 Prior Work on Active Contours

Since their introduction, active contour models [KWT88] have been heavily investigated. The improvements are mainly in two aspects: 1) the methods to locate the contour with minimum energy, and 2) the types of energies acting on the contour.

The snake model [KWT88] uses a finite difference method to iteratively refine the solution. Amini et al. [AWJ90] use dynamic programming, which is numerically stable, in the search window to locate the contour with minimum energy. Williams et al. [WS92] use a greedy algorithm to improve the speed of dynamic programming from $O(nm^3)$ to $O(nm)$ where n is the number of points of the contour and m is the number of positions a point can move. Fujimura et al. [FYY93] use a coarse-to-fine approach to speed up the dynamic programming time. Geiger et al. [GGCV95] lets a user place initial points instead of contours on an image. A list of uncertainty for each point is calculated. Basically, the uncertainty list shows a higher cost for a point farther away from the selected point. A search window is created from two consecutive lists. A dynamic programming method is employed to trace the contour. Cohen et al. [CC93] apply a finite-element method to achieve greater stability and faster convergence. Finally, Lobregt et al. [LV95] develop a discrete model using vertices connected by line segments. All calculations of energies are limited to vertices only. This model uses re-sampling so the distance between vertices has approximately the same length after each iteration.

The snake model shrinks into a point or a line if no image energy is present. It also creates crowded vertices at high curvature portions of contours. Lobregt et al. [LV95] use a convolution filter applied to the internal force to solve the shrinking problem. Re-sampling as well as limiting the vertices moving avoids vertex clustering. Cohen's 'balloon' model [Coh91] adds an inflating force. The thresholded edge is diffused by a Gaussian filter and is added to the image energy. This allows better localization of real edges. Ronfard [Ron94] incorporates region-based segmentation into the active contour models. Ronfard partitions the region along the contour into small regions, and computes the image force from the deformation of these partitioned regions. Malladi, Sethian, and Vemuri [MSV95, MS96, Set96] use a level set approach to make a contour front advance to the desired boundary. The advancing front is a time-dependent (piecewise bilinear) function over a static structed mesh (Cartesian image grid).

3 Image Contours with A-splines

In this section we briefly summarize, for the sake of completeness, the process of constructing an A-spline contour approximation from a closed linear polygon [BX96]. The specific steps we take are:

1. Compute breakpoints along the contour. These breakpoints points are the junction points for the A-spline segments which make up the A-spline contour.

2. For C^1 and C^2 A-splines, compute first and second derivatives at the junction points using local divided differences along the contour.

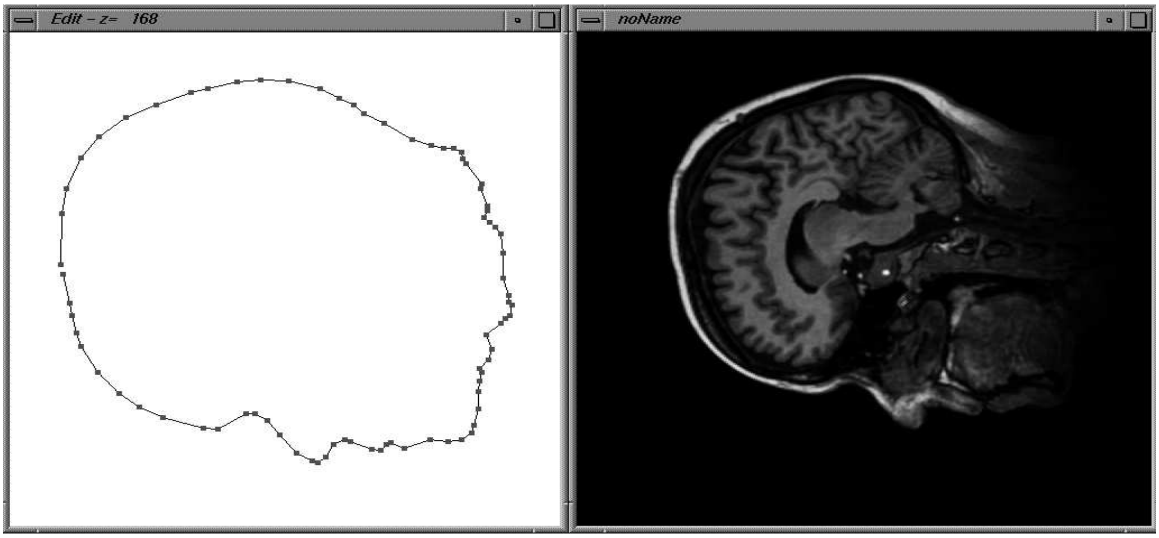

Figure 3.1: Extracting an isocontour (left) from a dense MRI slice (right)

3. Construct 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.

An A-spline curve consists of the zero contour of some piecewise smooth BB polynomials defined over a simplicial hull Σ, or a triangulation of a connected region of the space. In particular,

$$F^{(i)}(\boldsymbol{\alpha}) = \sum_{|\boldsymbol{\lambda}|=d} b^{(i)}_{\boldsymbol{\lambda}} B^d_{\boldsymbol{\lambda}}(\boldsymbol{\alpha}) = \mathbf{b}^{(i)^{\mathsf{T}}} \mathbf{B}^d = 0 \tag{3.1}$$

is the zero contour of a BB polynomial defined within the i-th simplex (triangles in 2-D) in Σ. The base functions $B^d_{\boldsymbol{\lambda}}$ are grouped into a vector $\mathbf{B}$ and the coefficients $b_{\boldsymbol{\lambda}}$ into $\mathbf{b}$. Smoothness of certain degrees and local interpolation of certain degrees are enforced by some linear equality constraints

$$\mathbf{b}^{\mathsf{T}}\mathbf{C}(\mathbf{p}) = \mathbf{0}\ , \tag{3.2}$$

and connectedness of the curve is enforced by additional linear sign inequalities

$$\mathbf{b}^{\mathsf{T}}\mathbf{S} > \mathbf{0}\ . \tag{3.3}$$

Vector $\mathbf{b}$ is a global collection of the coefficient vector $\mathbf{b}^{(i)}$ of all simplices in Σ, and $\mathbf{S}$ and $\mathbf{C}(\mathbf{p})$ are defined explicitly for A-splines with C^k continuity in [BX92]. An example contour extraction is shown in the left part of Figure 3.1 from the input MRI (Magnetic Resonance Imaging) image slice on the right. Of course, more sophisticated edge detection or contour extraction algorithms may also be used, see for e.g. [Can86].

To compute the junction points around an image contour, in [Baj91] we use a curvature adaptive scheme for the placement of the A-spline segments. In this paper, we construct junction points via an iterative contour coarsening scheme, preserving some local features of the image. There are various forms of divided-difference methods that extract geometric information around a junction

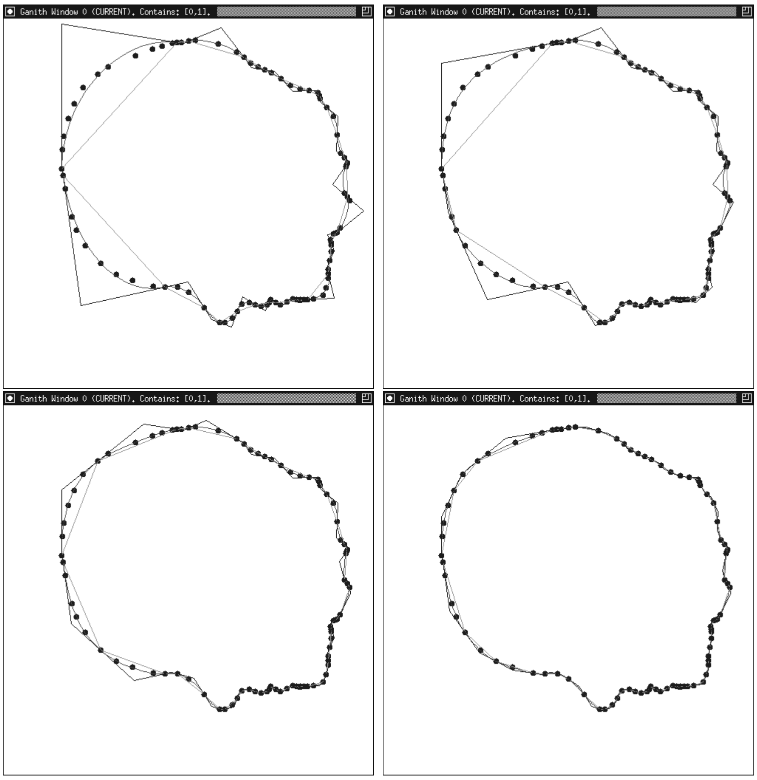
Ganith Window 0 (CURRENT). Contains: [0,1].
Ganith Window 0 (CURRENT). Contains: [0,1].
Ganith Window 0 (CURRENT). Contains: [0,1].
Ganith Window 0 (CURRENT). Contains: [0,1].

Figure 3.2: Junction Points and Cubic A-spline Fits

point, from a given list of points or dense image grids [deB78, BX96]. We use compact formulas for higher order divided differences given in [BI93].

After satisfying the derivatives at the junction points, A-splines still possess extra degrees of freedom. These degrees of freedom are used to least-squares approximate points in the triangle while satisfying the sign requirements. In the event that there are not enough data points within a triangle (zero probability for a dense image contouring), default choices of values for the undetermined coefficients are used. One method to determine these default coefficient values is to approximate locally a quadratic curve within the triangle. This tends to avoid sharp changes in the local geometry of the spline curve. The quadratic approximation is easily achieved by using degree elevation formulas (see [Far90]).

4 Algorithm Sketch for Active Contouring of Images

In this section we sketch the main steps of our algorithm for active contouring of images using C^1 quadratic and C^3 cubic A-splines. The former A-spline allows us to achieve the same degree of continuity as a traditional cubic polynomial B-spline.

- Step 1: Given an image, we apply the level set method of Sethian et al. [MSV95] to obtain a good geometric fit to the desired image boundary (often multiple contour components), and construct an A-spline contour approximation as sketched in Section 3 and detailed in Section 5.
- Step 2: Next, construct a physical A-spline contour from the geometric one obtained in

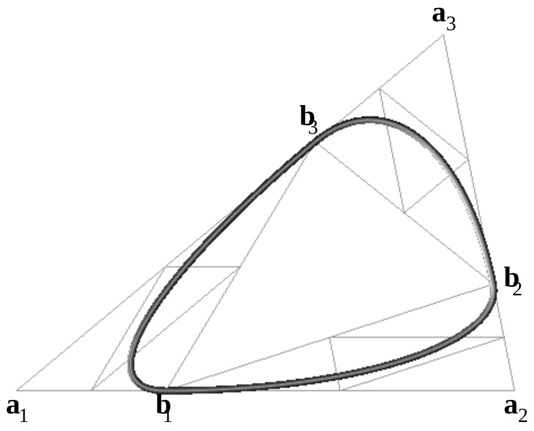

Figure 4.1: The points $\mathbf{b}_1$, $\mathbf{b}_2$, and $\mathbf{b}_3$ slide independently along the edges of $\triangle\mathbf{a}_1\mathbf{a}_2\mathbf{a}_3$.

Step 1. We define a strain energy function E_{strain}, and an image energy function E_{image} in a manner similar to that of [KWT88]. The strain energy, or internal spline energy, E_{strain} is a function of the spline's arc length and curvature [Mum94], and the image energy E_{image} is typically a function of the intensity $I(x, y)$ at each pixel of an image. If $f(x, y) = \pm I(x, y)$ the contour will be attracted to light or dark areas, while if $f = -|\nabla I(x, y)|^2$, the contour will be attracted to boundaries between light and dark regions. Details of how the energy functions are obtained are provided in Section 6.

- Step 3: The final step involves minimizing the total energy of the A-spline. We construct an energy minimizing A-spline by finding the energy minimizing spline inside each triangle using a formulation like (7.1) below. The general C^1 quadratic spline in BB coordinates has the form $b_{101}\alpha_1(1-\alpha_1-\alpha_2)-\alpha_2^2$, as shown in [BX92, BCHN98]. Thus for each triangle, finding the minimum energy is a one-variable minimization problem. A derivation of the energy minimizing process is given in Section 7.

An example of such constrained motion is shown in Figure 4.1. In this example, we have a closed contour with three junction points $\mathbf{b}_1$, $\mathbf{b}_2$, and $\mathbf{b}_3$, and three control triangles $\Delta\mathbf{b}_1\mathbf{a}_2\mathbf{b}_2$, $\Delta\mathbf{b}_2\mathbf{a}_3\mathbf{b}_3$, and $\Delta\mathbf{b}_3\mathbf{a}_1\mathbf{b}_1$. By restricting the $\mathbf{b}_i$ to slide along the edges of $\Delta\mathbf{a}_1\mathbf{a}_2\mathbf{a}_3$, we have three degrees of freedom to describe their positions as compared to six degrees of freedom with no restriction on their motion. If the interpolating splines are C^1 quadratic, we have an additional three degrees of freedom to express the splines within the control triangles.

5 A-spline Control Polygon from Images

In this section we present a method to build a control polygon with marked junction endpoints for an A-spline approximation of the desired segmented contour. We first discuss some intrinsic conditions that the junction endpoints and the control polygon need to satisfy.

Interpolation An A-spline interpolates the junction endpoints (vertices of the base) of each triangle containing the A-spline segment. This implies that such points need to lie very close to the desired segmented contour boundary C.

Tangency Adjacent A-spline segments are tangent to the edges of the polygon at the junction points. The position of the junction points can be selected arbitrarily along the interior of each edge of the polygon. We will take advantage of these degrees of freedom for the junction points.

Inflection Points Each A-spline segment is convex within its triangle [BX92]. This implies that any inflection point of the desired segmented contour boundary C must lie close to one of the junction points of the A-spline segment.

Angle Span The length of each arc approximated with an A-spline (even without inflection points) is bounded by the curvature of the curve segment. In particular the total angle span covered by an A-spline segment must be bounded by a maximum angle smaller than π (for example, a semicircle cannot be represented by a single A-spline segment unless the apex of its containing triangle lies at infinity, i. e. a projective A-spline).

We build an A-spline contour control polygon satisfying the above four conditions, as a "coarse" representation of the desired segmented contour boundary curve C, preserving level set local image and polygon features like inflection points and tangency. We first use the method of Sethian et al. [MSV95] to obtain a fine piecewise linear polygon approximation Q_0 of the desired boundary curve C. This polygon Q_0 has the advantage of being sufficiently fine to capture the required local features of C. To ensure that the final A-spline contour control polygon satisfies the above four conditions, we go through an iterative coarsening process to obtain successively coarser polygons $Q_1, Q_2, \ldots, Q_k$ with fewer sides, however preserving the necessary image features of vertices of Q_0.

The coarsening scheme applied to the polygon Q_0 differs from standard error-bounded shape simplification schemes since in this case, we additionally need to satisfy the inflection point conditions instead of being concerned on keeping a low overall error of the coarse polygons [BS96]. Note that we use Q_0 both as the initial polygon in the simplification process and as a good approximation of the segmented contour boundary curve C (see figure 5.1).

First of all we detect the reflex edges of Q_0, the edges at which the concavity of the polygon changes. Such edges need to be preserved in the simplification process since each of them contains an inflection point (see figure 5.1(b)). We will take as the inflection point the midpoint of each such edge.

To coarsen the polygon we remove every other edge in the polygon, as long as it is not a reflex. For each removed edge we extend the two adjacent edges (that we keep) up to their intersection point as in figure 5.1(c). We repeat the process as long as we can remove a non-reflex edge by extending the two edges adjacent to it. The condition for this rule to be applied is as follows. Call e the edge we want to remove, e_l and e_r the two edges at its left and right, and a the intersection point between the lines through e_l and e_r. (If e_l and e_r are parallel, point a does not exist, and edge e cannot be removed.) The edge e can be removed if the triangle T of base e and apex a contains neither e_l nor e_r. Examples of the various cases are shown in figure 5.2. In examples (a) and (b) the edge e cannot be removed since either e_l or both e_l and e_r are contained in T. Example (c) shows the case in which the edge e can be removed. For practical reasons we will restrict such condition by requiring that the angle of T at the apex a is greater than some positive value θ, such as 30° (this is shown in example (d) in the figure).

6 Physical A-splines and Strain Energy Models

In this section we capture physical information from the material properties of the A-splines. An example of this is relating computer tomography (CT) data and elasticity correlation as in [CH77]. We begin with a description of how linear, quadratic, and cubic A-splines can be parametrized, and then develop models of the strain and image energies involved.

6.1 Parametrizations of A-splines

In order to perform the various integrations along the splines, we need parametrizations for them. We proceed to derive parametrizations of linear, quadratic, and cubic A-splines. Doing the linear case first, we observe that a parametrization of a general line in BB form

$$S(\alpha_1,\alpha_2) = b_{100}\alpha_1 + b_{010}\alpha_2 + b_{001}(1-\alpha_1-\alpha_2) = 0 \tag{6.1}$$

is given by

$$\alpha_1 = u \qquad \alpha_2 = 0 \qquad (\alpha_3 = 1-u) \ , \tag{6.2}$$
$$0 \le u \le 1 \ .$$

Next, we turn our attention to the more interesting case of quadratic A-splines. A parametrization of the general quadratic with C^1 continuity at $\mathbf{p}_1$ and $\mathbf{p}_3$,

$$S(\alpha_1,\alpha_2) = 2b_{101}\alpha_1(1-\alpha_1-\alpha_2) - \alpha_2^2 \tag{6.3}$$

is obtained in [BX92] by intersecting $S(\alpha_1,\alpha_2)=0$ with the line $\alpha_1 = u(1-\alpha_2)$ to obtain

$$\alpha_1(u;\mathbf{b}) = \frac{u}{1+\sqrt{2b_{101}u(1-u)}} \qquad \alpha_2(u;\mathbf{b}) = \frac{\sqrt{2b_{101}u(1-u)}}{1+\sqrt{2b_{101}u(1-u)}} \ , \tag{6.4}$$
$$0 \le u \le 1 \ .$$

Here $\mathbf{b}$ denotes the column vector of the coefficients b_{ijk}, and in this case $\mathbf{b}$ consists of the single element b_{101}. For now we will assume $\mathbf{b}$ is constant when deriving the energy of a spline; later on when dealing with dynamic splines, $\mathbf{b}$ will be allowed to vary. Also, u parametrizes the line segment from $\mathbf{p}_3$ to $\mathbf{p}_1$.

Now we consider cubic A-splines with C^3 continuity. The equation of the general C^3 cubic is

$$\begin{aligned} S(\alpha_1,\alpha_2) \ = \ & 3b_{201}\alpha_1^2(1-\alpha_1-\alpha_2) + 3b_{120}\alpha_1\alpha_2^2 + 6b_{111}\alpha_1\alpha_2(1-\alpha_1-\alpha_2) \\ & + 3b_{102}\alpha_1(1-\alpha_1-\alpha_2)^2 + 3b_{021}\alpha_2^2(1-\alpha_1-\alpha_2) - \alpha_2^3 = 0 \ , \end{aligned} \tag{6.5}$$

in accordance with [BX96]. When one specifies the second and third order derivatives at $\mathbf{p}_1$ and $\mathbf{p}_3$, it is possible to express the four coefficients b_{201}, b_{120}, b_{102}, and b_{021} as linear functions of the remaining coefficient, b_{111}. These relations are given in [BX96], and the following cubic parametrization appears in [BX92]. When we intersect $S(\alpha_1,\alpha_2)=0$ with the line $\alpha_1 = u(1-\alpha_2)$, we obtain a

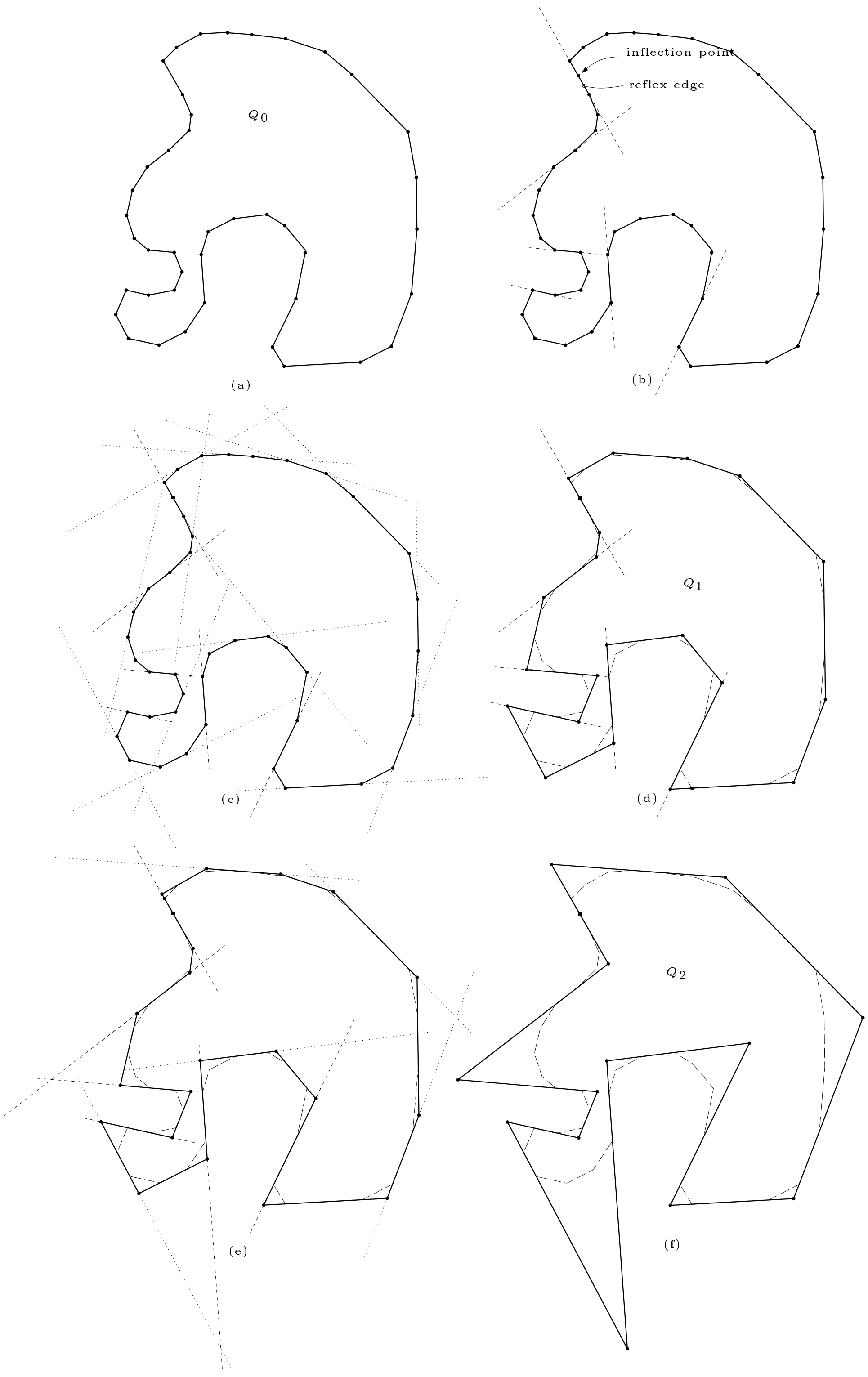

Figure 5.1: Successive steps of the simplification process applied to the polygon $Q_0(x, y)$.

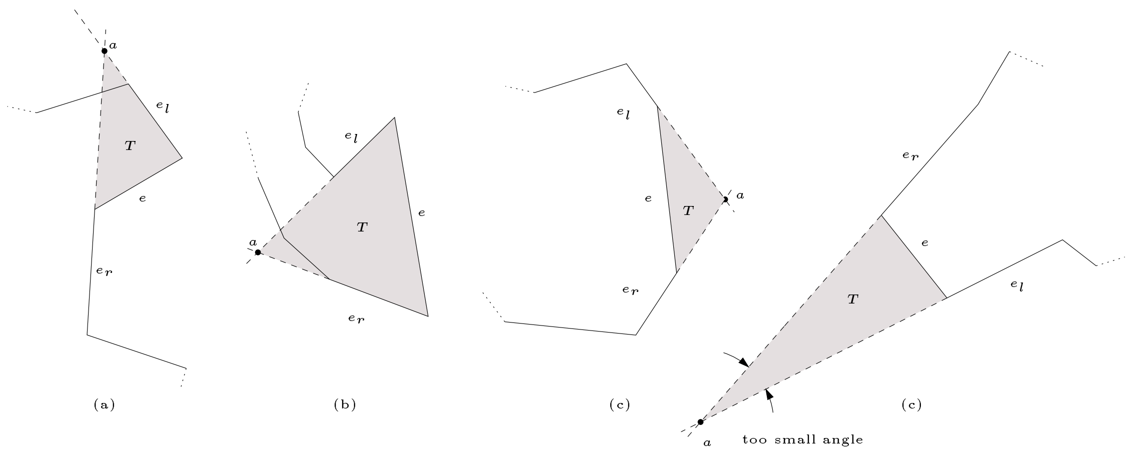

Figure 5.2: The procedure for the removal of the edge e from the polygon can fail because either e_l or e_r is contained in T as in (a) and (b), can be successful as in (c), or can be rejected because the angle of T at a is too small.

parametrization in which $\alpha_1 = u(1-\alpha_2)$, and α_2 is a root of the cubic equation in α_2. If we make the substitution $\alpha_2 = 1/(1+t)$, we obtain this cubic polynomial in t:

$$At^3 + Bt^2 + Ct - 1 = 0 \ , \tag{6.6}$$

where

$$\begin{aligned} A &= 3u(1-u)[b_{201}u + b_{102}(1-u)] \\ B &= 6u(1-u)b_{111} \\ C &= 3[b_{120}u + b_{021}(1-u)] \ . \end{aligned}$$

The inequality constraints for cubic A-splines in [BX96] are

$$b_{201} > 0 \ , \ \ b_{102} > 0 \ , \ \ b_{021} \le 0 \ , \ \ b_{120} \le 0 \ , \tag{6.7}$$

so we have $A \ge 0$ and $C \le 0$ for all $u \in [0,1]$.

Now let

$$p = \frac{3AC - B^2}{3A^2} \ , \qquad q = \frac{2B^3 - 9ABC - 27A^2}{27A^3} \ .$$

If $(q/2)^2 + (p/3)^3 \ge 0$, then since $t(u) = [1-\alpha_2(u)]/\alpha_2(u)$ is a real root of (6.6), we must have

$$t(u) = \sqrt[3]{-\frac{q}{2} + \sqrt{\left(\frac{q}{2}\right)^2 + \left(\frac{p}{3}\right)^3}} + \sqrt[3]{-\frac{q}{2} - \sqrt{\left(\frac{q}{2}\right)^2 + \left(\frac{p}{3}\right)^3}} - \frac{B}{3A} \ . \tag{6.8}$$

If $(q/2)^2 + (p/3)^3 < 0$, then p must be negative, and (6.6) has three real roots. Define

$$r = \frac{(-p)^{3/2}}{3\sqrt{3}} \ , \qquad \theta = \frac{1}{3}\cos^{-1}\left(-\frac{q}{2r}\right) \ .$$

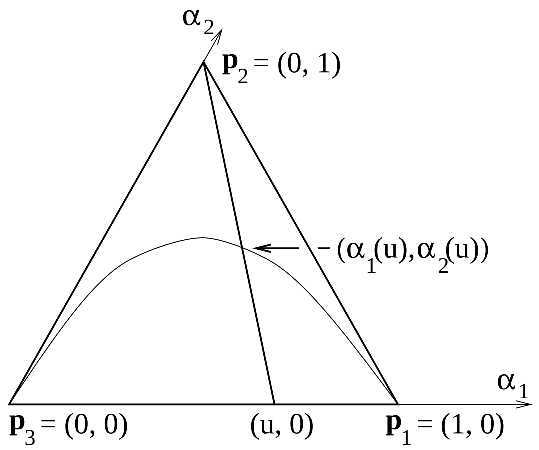

Figure 6.1: Representation of points in local BB coordinates. Point $\mathbf{p}_i$ has Cartesian coordinates (x_i, y_i), $i = 1, 2, 3$. A point $(\alpha_1(u), \alpha_2(u))$ is given by the intersection of the spline with the line connecting the points $(u, 0)$ and $(0, 1)$ $(\mathbf{p}_2)$ in BB coordinates.

Then the root we want is

$$t(u) = 2r^{1/3}\cos\theta - \frac{B}{3A} \quad . \tag{6.9}$$

The final cubic parametrization is then

$$\alpha_1(u;\mathbf{b}) = u[1-\alpha_2(u;\mathbf{b})] \ , \ \alpha_2(u;\mathbf{b}) = \frac{1}{1+t(u;\mathbf{b})} \ , \quad 0 \le u \le 1 \quad . \tag{6.10}$$

where $t(u;\mathbf{b})$ is given by (6.8) or (6.9).

6.2 Strain energy models

Let s be the arc length coordinate of a point on a plane curve C with parametrization $\mathbf{w}(a)$. For this parametrization, the elastic potential energy of a deformable curve is as follows [Mum94]:

$$E_{strain} = \int_C \left(\beta + \gamma\kappa^2\right) ds \quad . \tag{6.11}$$

The terms β and $\gamma\kappa^2$ represent the stretching and bending energy, respectively.

Let $S(\alpha_1, \alpha_2) = 0$ be an A-spline defined within a triangle $\Delta\mathbf{p}_1\mathbf{p}_2\mathbf{p}_3$ (see figure 6.1). The curved piece interpolates $\mathbf{p}_1$ and $\mathbf{p}_3$ and is tangent to $\overline{\mathbf{p}_1\mathbf{p}_2}$ and $\overline{\mathbf{p}_3\mathbf{p}_2}$ at $\mathbf{p}_1$ and $\mathbf{p}_3$ respectively. Let $F(x, y) = 0$ be the representation of the spline in Cartesian coordinates. The Cartesian coordinates of $\mathbf{p}_1$, $\mathbf{p}_2$ and $\mathbf{p}_3$ are (x_1, y_1), (x_2, y_2) and (x_3, y_3), respectively, and their local barycentric coordinates, suppressing the third coordinate $\alpha_3 = 1 - \alpha_1 - \alpha_2$, are $(1, 0)$, $(0, 1)$, and $(0, 0)$, respectively.

The following notation will be used:

$$\begin{aligned} d_{ij} &= \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \\ c_{12} &= (x_1 - x_3)(x_2 - x_3) + (y_1 - y_3)(y_2 - y_3) \\ \Delta &= \begin{vmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ 1 & 1 & 1 \end{vmatrix} \quad . \end{aligned}$$

In the linear case, when we use the parametrization (6.2), we see that $\nabla S = [0 \;\; 1]^{\mathsf{T}}$ and $\nabla^2 S$ is the 2×2 zero matrix, so $\kappa = 0$ in (6.11), which must be the case for a straight line segment. Thus only the stretching energy is present in this formulation. Equation (6.11) then reduces to

$$E_{strain} = \beta d_{13} \quad , \tag{6.12}$$

a multiple of the arc length, here the distance from $\mathbf{p}_1$ to $\mathbf{p}_3$, as expected.

In [BCHN98], these following expressions for the strain energy of a quadratic A-spline were obtained. For brevity, we define

$$b = \sqrt{2b_{101}} \quad .$$

$$\begin{aligned}
E_{strain} = \int_0^1 \Bigg\{ \beta \Bigg(& \Big\{ 4u(1-u)d_{13}^2 \;+ 2b\sqrt{u(1-u)}[(1-2u)(d_{23}^2 - d_{12}^2) + d_{13}^2] \\
& + b^2[-u(1-2u)d_{12}^2 + u(1-u)d_{13}^2 + (1-u)(1-2u)d_{23}^2]\Big\}^{1/2} \Big/ \\
& \left\{ 2\sqrt{u(1-u)} \left[1 + b\sqrt{u(1-u)} \right]^2 \right\} \Bigg) \\
+ \gamma \Bigg[2b^2\Delta^2 & \left[1 + b\sqrt{u(1-u)} \right]^4 \Big/ \Bigg(\sqrt{u(1-u)} \Big\{ 4u(1-u)d_{13}^2 \\
& + 2b\sqrt{u(1-u)}[(1-2u)(d_{23}^2 - d_{12}^2) + d_{13}^2] \\
& + b^2[-u(1-2u)d_{12}^2 + u(1-u)d_{13}^2 + (1-u)(1-2u)d_{23}^2]\Big\}^{5/2} \Bigg) \Bigg] \Bigg\} \, du \quad .
\end{aligned} \tag{6.13}$$

and

$$\begin{aligned}
E_{strain} \;=\; & \int_0^{1/\sqrt{2}} \Big[\beta \Big(\Big\{ 4v^2(1-v^2)d_{13}^2 + 2bv\sqrt{1-v^2} \left[(1-2v^2)(d_{23}^2 - d_{12}^2) + d_{13}^2 \right] \\
& \quad + b^2[-v^2(1-2v^2)d_{12}^2 + v^2(1-v^2)d_{13}^2 + (1-v^2)(1-2v^2)d_{23}^2]\Big\}^{1/2} \Big/ \\
& \quad \left[\sqrt{1-v^2} \left(1 + bv\sqrt{1-v^2} \right)^2 \right] \Big) \\
& \quad + 4\gamma\Delta^2 b^2 \Big(\; (1 + bv\sqrt{1-v^2})^4 \; \Big/ \\
& \quad \sqrt{1-v^2} \Big\{ 4v^2(1-v^2)d_{13}^2 + 2bv\sqrt{1-v^2}[(1-2v^2)(d_{23}^2 - d_{12}^2) + d_{13}^2] \\
& \quad + b^2[-v^2(1-2v^2)d_{12}^2 + v^2(1-v^2)d_{13}^2 + (1-v^2)(1-2v^2)d_{23}^2]\Big\}^{5/2} \Big) \Big] \, dv \\
+ \; & \int_0^{1/\sqrt{2}} \Big[\beta \Big(\Big\{ 4w^2(1-w^2)d_{13}^2 + 2bw\sqrt{1-w^2} \left[(1-2w^2)(d_{12}^2 - d_{23}^2) + d_{13}^2 \right] \\
& \quad + b^2[(1-w^2)(1-2w^2)d_{12}^2 + w^2(1-w^2)d_{13}^2 - w^2(1-2w^2)d_{23}^2]\Big\}^{1/2} \Big/ \\
& \quad \left[\sqrt{1-w^2} \left(1 + bw\sqrt{1-w^2} \right)^2 \right] \Big) \\
& \quad + 4\gamma\Delta^2 b^2 \Big(\; (1 + bw\sqrt{1-w^2})^4 \; \Big/
\end{aligned} \tag{6.14}$$

$$\begin{aligned}&\sqrt{1-w^2}\,\Big\{4w^2(1-w^2)d_{13}^2+2bw\sqrt{1-w^2}[(1-2w^2)(d_{12}^2-d_{23}^2)+d_{13}^2]\\&\quad+b^2[(1-w^2)(1-2w^2)d_{12}^2+w^2(1-w^2)d_{13}^2-w^2(1-2w^2)d_{23}^2]\Big\}^{5/2}\Big)\Big]\;dw\end{aligned}$$

In the interest of computational speed, an application of Simpson's rule using three points for each of the two integrals in (6.14) yields this expression for the simplified strain energy:

$$\begin{aligned}E_{strain^*} &= \beta\Bigg(\frac{b(d_{12}+d_{23})}{6\sqrt{2}}+\frac{2d_{13}}{3(2+b)}+\frac{32}{[3\sqrt{7}(8+\sqrt{7}\,b)^2]}\cdot\\&\qquad\Big\{[28d_{13}^2+4b\sqrt{7}(-3d_{12}^2+4d_{13}^2+3d_{23}^2)+b^2(-6d_{12}^2+7d_{13}^2+42d_{23}^2)]^{1/2}\\&\qquad\quad+[28d_{13}^2+4b\sqrt{7}(3d_{12}^2+4d_{13}^2-3d_{23}^2)+b^2(42d_{12}^2+7d_{13}^2-6d_{23}^2)]^{1/2}\Big\}\Bigg)\\&+\gamma\Delta^2\Bigg(\frac{\sqrt{2}}{3b^3}\left(\frac{1}{d_{12}^5}+\frac{1}{d_{23}^5}\right)+\frac{8b^2}{3(2+b)d_{13}^5}+\frac{128(8+\sqrt{7}\,b)^4b^2}{3\sqrt{7}}\cdot\\&\qquad\Big\{[28d_{13}^2+4b\sqrt{7}(-3d_{12}^2+4d_{13}^2+3d_{23}^2)+b^2(-6d_{12}^2+7d_{13}^2+42d_{23}^2)]^{-5/2}\\&\qquad\quad+[28d_{13}^2+4b\sqrt{7}(3d_{12}^2+4d_{13}^2-3d_{23}^2)+b^2(42d_{12}^2+7d_{13}^2-6d_{23}^2)]^{-5/2}\Big\}\Bigg)\;.\end{aligned}\tag{6.15}$$

To get an idea of the values b is likely to take on, consider a typical triangle with $\mathbf{P}_1=(\sqrt{3},0)$, $\mathbf{P}_2=(0,1)$, and $\mathbf{P}_3=(-\sqrt{3},0)$ in Cartesian coordinates. Here $d_{12}=d_{23}=2$ and $d_{13}=\Delta=2\sqrt{3}$. The bending energy has its minimum of 0.28 at $b=1.55$, rises sharply, asymptotic to $0.20/b^3$, as $b\to 0^+$, and rises slowly, asymptotic to $0.084b$, as $b\to\infty$. The bending energy is 25.03 when $b=0.2$, and in practice we never see smaller values of b. This corresponds to $b_{101}=0.02$. Regarding large values of b, when b is greater than 6, the spline remains extremely close to an edge of the triangle for at least half the length of the edge, as in side $\mathbf{P}_1\mathbf{P}_2$ in Figure 6.2. Here, when $x=\sqrt{3}/2$ on the spline, $y=0.475$, which is 95% of the distance from the x-axis to side $\mathbf{P}_1\mathbf{P}_2$. In practice, the splines do not stick that close to the edges of the control triangles, so we generally have $b\le 6$, or $b_{101}\le 18$.

There is a similar construction of the strain energy for the C^3 cubic A-spline, but it is so unwieldy it will not be written explicitly here. The same idea using a change of variables and Simpson's rule for the approximation of the strain energy can be used for the cubic splines.

6.3 Image energy models

In addition to the internal strain energy of an A-spline, we must also consider the energy arising from the image itself. Models of image energy dictate to what features the spline is attracted. In [KWT88], the authors describe three types of image energy functionals, which cause splines to be attracted to lines, edges, and "terminations," defined as endpoints of line segments and corners. The total image energy is then defined as

$$E_{image}=\int(w_{line}E_{line}+w_{edge}E_{edge}+w_{term}E_{term})\,ds\;\;,\tag{6.16}$$

where the $w_{(\cdot)}$ are weights, and s is arc length, parametrizing the spline.

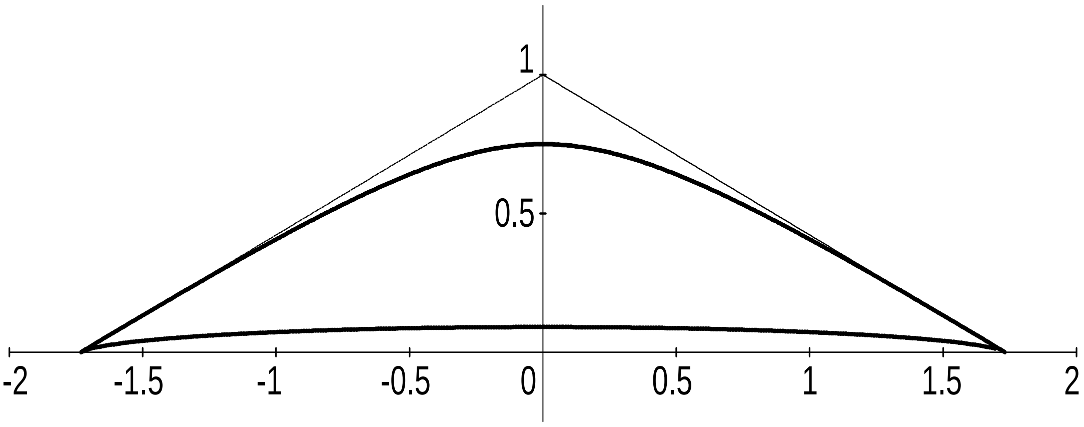

Figure 6.2: Extreme values of b_{101} occurring in practice. The lower and upper splines are those for $b_{101} = 0.02$ and $b_{101} = 18$, respectively.

The line functional is simply the image intensity,

$$E_{line} = I(x,y) = I^*(\alpha_1,\alpha_2) \ \ , \tag{6.17}$$

where the last expression is the image intensity expressed in BB coordinates. This functional causes the spline to be attracted to light or dark curves in the image, depending on the sign of w_{line}. The coordinates (x,y) and (α_1,α_2) are related by the expressions

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x_1 - x_3 & x_2 - x_3 \\ y_1 - y_3 & y_2 - y_3 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix} + \begin{bmatrix} x_3 \\ y_3 \end{bmatrix} \tag{6.18}$$

as indicated in [BCHN98].

A frequent desire is to get the spline to converge to a boundary of two regions of different intensity. For this purpose we use the edge functional, which is the opposite of the square of the magnitude of the gradient,

$$E_{edge} = -\|\nabla I\|^2 = -\left[\left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2\right] = -\nabla I^{*\mathsf{T}}\mathbf{J}\mathbf{J}^{\mathsf{T}}\nabla I^* = -\frac{d_{23}^2 {I^*}_{\alpha_1}^2 - 2c_{12}{I^*}_{\alpha_1}{I^*}_{\alpha_2} + d_{13}^2 {I^*}_{\alpha_2}^2}{\Delta^2}, \tag{6.19}$$

where $\nabla I^* = (\partial I^*/\partial\alpha_1, \partial I^*/\partial\alpha_2)^{\mathsf{T}}$.

Since the images studied will normally contain noise, instead of using the actual image intensity we should use a smoothed version of it. We shall use that in [MSV95], which is $G_\sigma * I(x,y)$, where $(G_\sigma * \cdot)$ denotes convolution with a Gaussian filter with standard deviation σ. This image energy is the same as that used in Section 5 when obtaining a piecewise linear curve that approximates an initial contour. Hereinafter, references to the image intensity $I(x,y)$ will assume that it has already been smoothed.

In [KWT88], an alternative edge energy is given,

$$E_{edge} = -(G_\sigma * \nabla^2 I)^2 \ \ ,$$

where G_σ is a bivariate Gaussian density with standard deviation σ in both variables. This functional allows the spline, or snake, to be "constrained by its own smoothness." Also, the termination energy is defined as

$$E_{term} = \frac{\partial\theta}{\partial \mathbf{n}_\perp} = \frac{\partial^2 C/\partial \mathbf{n}_\perp^2}{\partial C/\partial \mathbf{n}} = \frac{C_{yy}C_x^2 - 2C_{xy}C_xC_y + C_{xx}C_y^2}{(C_x^2 + C_y^2)^{3/2}} \quad ,$$

where $C(x,y) = G_\sigma(x,y) * I(x,y)$, $\theta = \tan^{-1}(C_y/C_x)$, $\mathbf{n} = (\cos\theta, \sin\theta)$, and $\mathbf{n}_\perp = (-\sin\theta, \cos\theta)$.

For our purposes, we are interested mainly in the ordinary edge energy given by (6.19), so in place of (6.16), we will use

$$E_{image} = -\frac{1}{\Delta^2}\int_0^1 \left(d_{23}^2 {I^*}_{\alpha_1}^2 - 2c_{12}{I^*}_{\alpha_1}{I^*}_{\alpha_2} + d_{13}^2 {I^*}_{\alpha_2}^2\right) du \quad , \tag{6.20}$$

where

$$I^* = I^*(\alpha_1(u;\mathbf{b}), \alpha_2(u;\mathbf{b})) \quad ,$$

and the parametrizations $\alpha_1(u;\mathbf{b})$ and $\alpha_2(u;\mathbf{b})$ are given by (6.2), (6.4), or (6.10) for linear, quadratic, or cubic splines respectively.

As was done with the strain energies, the integral for the image energy may be approximated using Simpson's rule. Here we will use the five points where $u = 0, 1/4, 1/2, 3/4$, and 1. The result is

$$E_{image^*} \quad = \quad -\frac{1}{12\Delta^2}\left[\hat{I}^*(0) + 4\hat{I}^*(1/4) + 2\hat{I}^*(1/2) + 4\hat{I}^*(3/4) + \hat{I}^*(1)\right] \tag{6.21}$$

where

$$\begin{aligned}\hat{I}^*(u) \quad = \quad & d_{23}^2[{I^*}_{\alpha_1}(\alpha_1(u;\mathbf{b}), \alpha_2(u;\mathbf{b}))]^2 \\ & - 2c_{12}{I^*}_{\alpha_1}(\alpha_1(u;\mathbf{b}), \alpha_2(u;\mathbf{b}))\ {I^*}_{\alpha_2}(\alpha_1(u;\mathbf{b}), \alpha_2(u;\mathbf{b})) \\ & + d_{13}^2[{I^*}_{\alpha_2}(\alpha_1(u;\mathbf{b}), \alpha_2(u;\mathbf{b}))]^2 \quad .\end{aligned}$$

or

$$E_{image^*} = -\frac{1}{12}\left[\hat{I}(0) + 4\hat{I}(1/4) + 2\hat{I}(1/2) + 4\hat{I}(3/4) + \hat{I}(1)\right] \tag{6.22}$$

where

$$\hat{I}(u) = [I_x(u)]^2 + [I_y(u)]^2 \quad .$$

For the quadratic parametrization (6.4), changing variables using (6.18) yields

$$x(u;\mathbf{b}) = \frac{x_1u + x_2b\sqrt{u(1-u)} + x_3(1-u)}{1 + b\sqrt{u(1-u)}} \qquad y(u;\mathbf{b}) = \frac{y_1u + y_2b\sqrt{u(1-u)} + y_3(1-u)}{1 + b\sqrt{u(1-u)}} \quad . \tag{6.23}$$

We observe that for $u = 0, 1/4, 1/2, 3/4, 1$,

$$\alpha_1(u;\mathbf{b}) \quad = \quad 0,\ \frac{1}{4+\sqrt{3}\,b},\ \frac{1}{2+b},\ \frac{3}{4+\sqrt{3}\,b},\ 1 \tag{6.24}$$

$$\alpha_2(u;\mathbf{b}) \quad = \quad 0,\ \frac{\sqrt{3}\,b}{4+\sqrt{3}\,b},\ \frac{b}{2+b},\ \frac{\sqrt{3}\,b}{4+\sqrt{3}\,b},\ 0, \tag{6.25}$$

$$x(u;\mathbf{b}) \quad = \quad x_3, \frac{x_1 + 3x_3 + \sqrt{3}x_2b}{4+\sqrt{3}\,b}, \frac{x_1 + x_3 + x_2b}{2+b}, \frac{x_1 + 3x_3 + \sqrt{3}x_2b}{4+\sqrt{3}\,b}, x_1, \tag{6.26}$$

$$y(u;\mathbf{b}) \quad = \quad y_3, \frac{y_1 + 3y_3 + \sqrt{3}y_2b}{4+\sqrt{3}\,b}, \frac{y_1 + y_3 + y_2b}{2+b}, \frac{y_1 + 3y_3 + \sqrt{3}y_2b}{4+\sqrt{3}\,b}, y_1, \tag{6.27}$$

for the quadratic parametrization (6.4).

Consequently, when the edge energy is written out in full in terms of the original Cartesian coordinates, we have

$$\begin{aligned} E_{image^*} \;=\; & -\frac{1}{12}\Big\{[I_x(x_3,y_3)]^2+[I_y(x_3,y_3)]^2 \\ & +4\left[I_x\left(\frac{x_1+3x_3+\sqrt{3}x_2b}{4+\sqrt{3}\,b},\frac{y_1+3y_3+\sqrt{3}y_2b}{4+\sqrt{3}\,b}\right)\right]^2 \\ & +4\left[I_y\left(\frac{x_1+3x_3+\sqrt{3}x_2b}{4+\sqrt{3}\,b},\frac{y_1+3y_3+\sqrt{3}y_2b}{4+\sqrt{3}\,b}\right)\right]^2 \\ & +2\left[I_x\left(\frac{x_1+x_3+x_2b}{2+b},\frac{y_1+y_3+x_2b}{2+b}\right)\right]^2 \\ & +2\left[I_y\left(\frac{x_1+x_3+x_2b}{2+b},\frac{y_1+y_3+x_2b}{2+b}\right)\right]^2 \\ & +4\left[I_x\left(\frac{3x_1+x_3+\sqrt{3}x_2b}{4+\sqrt{3}\,b},\frac{3y_1+y_3+\sqrt{3}y_2b}{4+\sqrt{3}\,b}\right)\right]^2 \\ & +4\left[I_y\left(\frac{3x_1+x_3+\sqrt{3}x_2b}{4+\sqrt{3}\,b},\frac{3y_1+y_3+\sqrt{3}y_2b}{4+\sqrt{3}\,b}\right)\right]^2 \\ & +[I_x(x_1,y_1)]^2+[I_y(x_1,y_1)]^2\Big\}\ . \end{aligned} \tag{6.28}$$

The cubic parametrization is much more complicated. As $u \to 0^+$ and as $u \to 1^-$, the quantity $(q/2)^2 + (p/3)^3$ is asymptotic with $[b_{021}/(27b_{102})]u^3$ and $[b_{120}/(27b_{201})](1-u)^3$, respectively, and both of these quantities are nonpositive in view of the inequality constraints (6.7). Thus we use (6.9), and find that $r \to \infty$, and hence $t(u) \to \infty$, as $u \to 0^+$ and as $u \to 1^-$. Thus we have $\alpha_2(u) \to 0$ in both cases, and we have

$$\begin{array}{ll} \alpha_1(0;\mathbf{b})=0 & \alpha_1(1;\mathbf{b})=1 \\ \alpha_2(0;\mathbf{b})=0 & \alpha_2(1;\mathbf{b})=0\ . \end{array} \tag{6.29}$$

When $u = 1/4, 1/2$, or $3/4$, the quantity $(q/2)^2 + (p/3)^3$ can be either positive or negative. If we let $A_{jk} = jb_{201} + kb_{102}$ and $B_{jk} = jb_{120} + kb_{021}$, then

$$\begin{aligned} \left[\left(\frac{q}{2}\right)^2+\left(\frac{p}{3}\right)^3\right]\Bigg|_{u=1/4} &= \frac{1024(4A_{13}B_{13}^3+9A_{13}^2+36A_{13}B_{13}b_{111}-12B_{13}^2b_{111}^2-96b_{111}^3)}{729A_{13}^4}\,, \\ \left[\left(\frac{q}{2}\right)^2+\left(\frac{p}{3}\right)^3\right]\Bigg|_{u=1/2} &= \frac{16(12A_{11}B_{11}^3+9A_{11}^2+36A_{11}B_{11}b_{111}-12B_{11}^2b_{111}^2-32b_{111}^3)}{81A_{11}^4}\,, \\ \left[\left(\frac{q}{2}\right)^2+\left(\frac{p}{3}\right)^3\right]\Bigg|_{u=3/4} &= \frac{1024(4A_{31}B_{31}^3+9A_{31}^2+36A_{31}B_{31}b_{111}-12B_{31}^2b_{111}^2-96b_{111}^3)}{729A_{31}^4}\,. \end{aligned}$$

Depending on whether these expressions are nonnegative or negative, $\alpha_i(1/4;\mathbf{b})$, $\alpha_i(1/2;\mathbf{b})$ and $\alpha_i(3/4;\mathbf{b})$, $i = 1, 2$, are computed through the use of (6.8) or (6.9).

7 Energy Minimizing Physical A-splines

In this section we use the results from Section 6 to minimize the total energy of a C^1 quadratic A-spline defined as

$$E_{total} = w_{strain} E_{strain} + w_{image} E_{image} \quad , \tag{7.1}$$

or the simplified equation

$$E_{total^*} = w_{strain} E_{strain^*} + w_{image} E_{image^*} \quad . \tag{7.2}$$

The strain and image energies may be expressed in terms of $\mathbf{b}$, a global vector of coefficients of the A-splines. In order to find the minimum energy, we need to solve $dE_{total}/d\mathbf{b} = 0$. However, as previously noted, the energy inside of each spline segment is independent of all the others, so we just need to sum over the individual energy contributions.

In the cases of the C^1 quadratic and C^3 cubic A-splines, there is a single degree of freedom among the coefficients. Thus for these cases the vector $\mathbf{b}$ may be interpreted as having a single component b, and the energy (7.1) can be expressed as a function of one parameter. The minimization of the overall energy is then found by adding up one term for each control triangle, where each term is the differentiation of the expression (7.1) with respect to one variable.

In order to differentiate the image energy (6.21) or (6.28) with respect to b, we need to evaluate several partial derivatives of I^* with respect to b. To this end, define

$$x_{i3} = x_i - x_3 \ , \qquad\qquad y_{i3} = y_i - y_3 \ , \qquad\qquad i = 1, 2 \ .$$

Then by several applications of the chain rule we obtain

$$\frac{\partial I^*}{\partial \alpha_i} = \frac{\partial I}{\partial x}\frac{\partial x}{\partial \alpha_i} + \frac{\partial I}{\partial y}\frac{\partial y}{\partial \alpha_i} = x_{i3} I_x + y_{i3} I_y \ , \quad i = 1, 2 \ . \tag{7.3}$$

$$\begin{aligned} \frac{\partial I^*_{\alpha_i}}{\partial b} &= \frac{\partial I^*_{\alpha_i}}{\partial a_1}\frac{\partial \alpha_1}{\partial b} + \frac{\partial I^*_{\alpha_i}}{\partial \alpha_2}\frac{\partial \alpha_2}{\partial b} \\ &= \left(\frac{\partial I^*_{\alpha_i}}{\partial x}\frac{\partial x}{\partial \alpha_1} + \frac{\partial I^*_{\alpha_i}}{\partial y}\frac{\partial y}{\partial \alpha_1}\right)\frac{\partial \alpha_1}{\partial b} + \left(\frac{\partial I^*_{\alpha_i}}{\partial x}\frac{\partial x}{\partial \alpha_2} + \frac{\partial I^*_{\alpha_i}}{\partial y}\frac{\partial y}{\partial \alpha_2}\right)\frac{\partial \alpha_2}{\partial b} \ , \quad i = 1, 2 \ . \end{aligned}$$

$$\begin{aligned} \frac{\partial I^*_{\alpha_1}}{\partial b} &= [x_{13}^2 I_{xx} + 2x_{13}y_{13}I_{xy} + y_{13}^2 I_{yy}]\frac{\partial \alpha_1}{\partial b} \\ &\quad + [x_{13}x_{23}I_{xx} + (x_{13}y_{23} + x_{23}y_{13})I_{xy} + y_{13}y_{23}I_{yy}]\frac{\partial \alpha_2}{\partial b} \end{aligned} \tag{7.4}$$

$$\begin{aligned} \frac{\partial I^*_{\alpha_2}}{\partial b} &= [x_{13}x_{23}I_{xx} + (x_{13}y_{23} + x_{23}y_{13})I_{xy} + y_{13}y_{23}I_{yy}]\frac{\partial \alpha_1}{\partial b} \\ &\quad + [x_{23}^2 I_{xx} + 2x_{23}y_{23}I_{xy} + y_{23}^2 I_{yy}]\frac{\partial \alpha_2}{\partial b} \end{aligned} \tag{7.5}$$

Hence we need to estimate I_{xx}, I_{xy}, and I_{yy} throughout the image as well as I_x and I_y.

The partial derivatives of α_i with respect to b are given by

$$\frac{\partial \alpha_1}{\partial b} = \frac{-u\sqrt{u(1-u)}}{\left[1+b\sqrt{u(1-u)}\right]^2} \qquad \frac{\partial \alpha_2}{\partial b} = \frac{\sqrt{u(1-u)}}{\left[1+b\sqrt{u(1-u)}\right]^2} \;, \tag{7.6}$$

and the partial derivatives of x and y with respect to b are given by

$$\frac{\partial x}{\partial b} = \frac{\sqrt{u(1-u)}[-ux_1 + x_2 - (1-u)x_3]}{[1+b\sqrt{u(1-u)}]^2} \qquad \frac{\partial x}{\partial b} = \frac{\sqrt{u(1-u)}[-ux_1 + x_2 - (1-u)x_3]}{[1+b\sqrt{u(1-u)}]^2} \tag{7.7}$$

At $u = 0, 1/4, 1/2, 3/4, 1$,

$$\frac{\partial \alpha_1}{\partial b} = 0,\; -\frac{\sqrt{3}}{(4+\sqrt{3}\,b)^2},\; \frac{2}{(2+b)^2},\; -\frac{3\sqrt{3}}{(4+\sqrt{3}\,b)^2},\; 0\;, \tag{7.8}$$

$$\frac{\partial \alpha_2}{\partial b} = 0,\; \frac{4\sqrt{3}}{(4+\sqrt{3}\,b)^2},\; -\frac{1}{(2+b)^2},\; \frac{4\sqrt{3}}{(4+\sqrt{3}\,b)^2},\; 0\;, \tag{7.9}$$

$$\frac{\partial x}{\partial b} = 0, \frac{\sqrt{3}(-x_1+4x_2-3x_3)}{(4+\sqrt{3}\,b)^2}, \frac{-x_1+2x_2-x_3}{(2+b)^2}, \frac{\sqrt{3}(-3x_1+4x_2-x_3)}{(4+\sqrt{3}\,b)^2}, 0\;, \tag{7.10}$$

$$\frac{\partial y}{\partial b} = 0, \frac{\sqrt{3}(-y_1+4y_2-3y_3)}{(4+\sqrt{3}\,b)^2}, \frac{-y_1+2y_2-y_3}{(2+b)^2}, \frac{\sqrt{3}(-3y_1+4y_2-y_3)}{(4+\sqrt{3}\,b)^2}, 0\;. \tag{7.11}$$

Next, we wish to differentiate the strain energy (6.15) with respect to b. This is a straightforward task, with the result being

$$\begin{aligned}
\frac{dE_{strain^*}}{db} = \beta \Bigg(& \frac{d_{12}+d_{23}}{6\sqrt{2}} - \frac{2d_{13}}{3(2+b)^2} - \frac{32}{21(8+\sqrt{7}\,b)^3} \cdot \\
& \{[56(6d_{12}^2 - d_{13}^2 - 6d_{23}^2) + 2b\sqrt{7}(-39d_{12}^2 + 56d_{13}^2 - 105d_{23}^2) + 7b^2(-6d_{12}^2 + 7d_{13}^2 + 42d_{23}^2)] / \\
& \qquad [28d_{13}^2 + 4b\sqrt{7}(-3d_{12}^2 + 4d_{13}^2 + 3d_{23}^2) + b^2(-6d_{12}^2 + 7d_{13}^2 + 42d_{23}^2)]^{1/2} \\
& + [-56(6d_{12}^2 + d_{13}^2 - 6d_{23}^2) + 2b\sqrt{7}(-105d_{12}^2 + 56d_{13}^2 - 39d_{23}^2) + 7b^2(42d_{12}^2 - 7d_{13}^2 - 6d_{23}^2)] / \\
& \qquad [28d_{13}^2 + 4b\sqrt{7}(3d_{12}^2 + 4d_{13}^2 - 3d_{23}^2) + b^2(42d_{12}^2 + 7d_{13}^2 - 6d_{23}^2)]^{1/2}\} \Bigg) \\
+ \quad \gamma\Delta^2 \Bigg(& -\frac{\sqrt{2}}{b^4}\left(\frac{1}{d_{12}^5} + \frac{1}{d_{23}^5}\right) + \frac{8b(4+b)}{(2+b)^2 d_{13}^5} + \frac{128(8+\sqrt{7}\,b)^3 b}{21} \cdot \\
& \Bigg\{ \frac{448\sqrt{7}d_{13}^2 + 56b(6d_{12}^2 + 13d_{13}^2 - 6d_{23}^2) + 2b^2\sqrt{7}(-75d_{12}^2 + 112d_{13}^2 - 357d_{23}^2) + 7b^3(-6d_{12}^2 + 7d_{13}^2 + 42d_{23}^2)}{[28d_{13}^2 + 4b\sqrt{7}(-3d_{12}^2 + 4d_{13}^2 + 3d_{23}^2) + b^2(-6d_{12}^2 + 7d_{13}^2 + 42d_{23}^2)]^{7/2}} \\
& + \frac{448\sqrt{7}d_{13}^2 + 56b(-6d_{12}^2 + 13d_{13}^2 + 6d_{23}^2) + 2b^2\sqrt{7}(-357d_{12}^2 + 112d_{13}^2 - 75d_{23}^2) + 7b^3(42d_{12}^2 + 7d_{13}^2 - 6d_{23}^2)}{[28d_{13}^2 + 4b\sqrt{7}(3d_{12}^2 + 4d_{13}^2 - 3d_{23}^2) + b^2(42d_{12}^2 + 7d_{13}^2 - 6d_{23}^2)]^{7/2}} \Bigg\} \Bigg).
\end{aligned}$$

The above equations provide the required expression of dE/db in terms of b so that it can be solved. The problem remains that the derivatives $I_x(x,y)$, $I_y(x,y), \ldots$ are not provided analytically but are sampled on the vertices of the regular grid displayed in figure 7.1(b). As a consequence dE/db is defined as a piecewise function. To determine the representation of dE/db we have that in each square of the image grid the derivatives can be defined as the bilinear interpolation of the four values estimated at the vertices of the square. For example they can be approximated in the

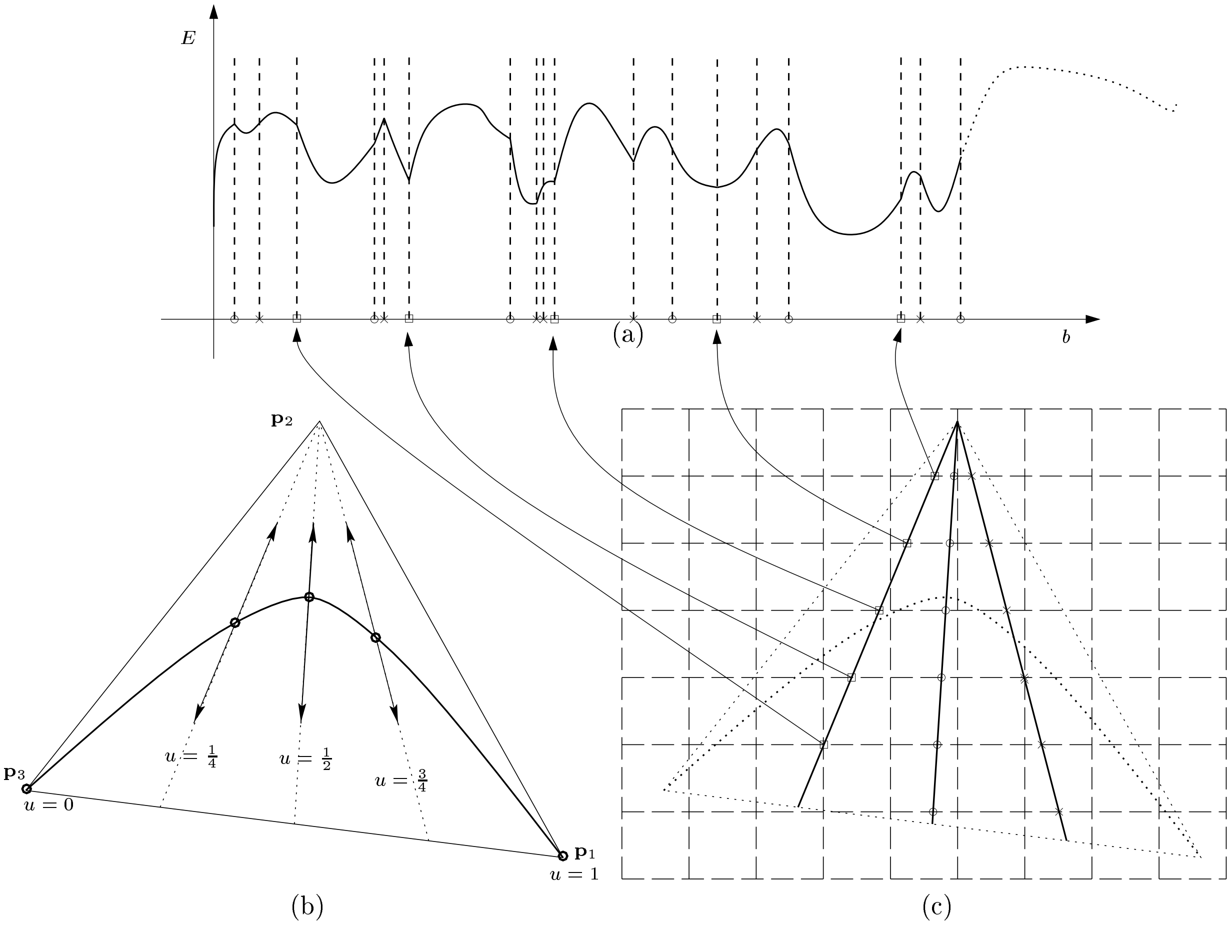

Figure 7.1: (a) Plot of energy as a function of b. The derivative may be discontinuous whenever one of the points corresponding to $u = 1/4, 1/2$, or $3/4$ on the spline crosses the edge of a cell. (b) Lines along which the three samples move when b varies from 0 to $+\infty$. (c) Image grid that determines the piecewise representation of $\alpha(1/4; \mathbf{b})$, $\alpha(1/2; \mathbf{b})$ and $\alpha(3/4; \mathbf{b})$.

region $x_i \leq x \leq x_{i+1}$, $y_j \leq y \leq y_{j+1}$ by the expression:

$$\begin{aligned} &I_x(x,y) = \\ &\quad \frac{1}{(x_{i+1}-x_i)(y_{j+1}-y_j)}[I_x(x_i,y_j)(x_{i+1}-x)(y_{j+1}-y) + I_x(x_i,y_{j+1})(x_{i+1}-x)(y-y_j) \\ &\qquad + I_x(x_{i+1},y_j)(x-x_i)(y_{j+1}-y) + I_x(x_{i+1},y_{j+1})(x-x_i)(y-y_j)]\,. \end{aligned} \tag{7.12}$$

From equations (6.24) we can determine for which values of b the spline crosses one of the lines in the image grid and hence the values of i and j that appear in (7.12) change.

In view of (6.22), we have

$$\frac{dE_{image^*}}{db} = -\frac{1}{12}\left[\frac{\partial \hat{I}(0)}{\partial b} + 4\frac{\partial \hat{I}(1/4)}{\partial b} + 2\frac{\partial \hat{I}(1/2)}{\partial b} + 4\frac{\partial \hat{I}(3/4)}{\partial b} + \frac{\partial \hat{I}(1)}{\partial b}\right]$$

where

$$\frac{\partial \hat{I}(u)}{\partial b} = 2I_x(u)\left[I_{xx}(u)\frac{\partial x(u)}{\partial b} + I_{xy}(u)\frac{\partial y(u)}{\partial b}\right] + 2I_y(u)\left[I_{xy}(u)\frac{\partial x(u)}{\partial b} + I_{yy}(u)\frac{\partial y(u)}{\partial b}\right] \quad .$$

When written out in full, this becomes

$$\begin{aligned}
\frac{dE_{image^*}}{db} \quad = \quad & -\frac{1}{12}\Bigg\{2I_x\Bigg(\frac{x_1+3x_3+\sqrt{3}x_2b}{4+\sqrt{3}\,b},\frac{y_1+3y_3+\sqrt{3}y_2b}{4+\sqrt{3}\,b}\Bigg) \\
& \cdot\Bigg[\frac{4\sqrt{3}(-x_1+4x_2-3x_3)}{(4+\sqrt{3}\,b)^2}I_{xx}\Bigg(\frac{x_1+3x_3+\sqrt{3}x_2b}{4+\sqrt{3}\,b},\frac{y_1+3y_3+\sqrt{3}y_2b}{4+\sqrt{3}\,b}\Bigg) \\
& \quad + \frac{4\sqrt{3}(-y_1+4y_2-3y_3)}{(4+\sqrt{3}\,b)^2}I_{xy}\Bigg(\frac{x_1+3x_3+\sqrt{3}x_2b}{4+\sqrt{3}\,b},\frac{y_1+3y_3+\sqrt{3}y_2b}{4+\sqrt{3}\,b}\Bigg)\Bigg] \\
& + 2I_y\Bigg(\frac{x_1+3x_3+\sqrt{3}x_2b}{4+\sqrt{3}\,b},\frac{y_1+3y_3+\sqrt{3}y_2b}{4+\sqrt{3}\,b}\Bigg) \\
& \cdot\Bigg[\frac{4\sqrt{3}(-x_1+4x_2-3x_3)}{(4+\sqrt{3}\,b)^2}I_{xy}\Bigg(\frac{x_1+3x_3+\sqrt{3}x_2b}{4+\sqrt{3}\,b},\frac{y_1+3y_3+\sqrt{3}y_2b}{4+\sqrt{3}\,b}\Bigg) \\
& \quad + \frac{4\sqrt{3}(-y_1+4y_2-3y_3)}{(4+\sqrt{3}\,b)^2}I_{yy}\Bigg(\frac{x_1+3x_3+\sqrt{3}x_2b}{4+\sqrt{3}\,b},\frac{y_1+3y_3+\sqrt{3}y_2b}{4+\sqrt{3}\,b}\Bigg)\Bigg] \\
& + 2I_x\Bigg(\frac{x_1+x_3+x_2b}{2+b},\frac{y_1+y_3+y_2b}{2+b}\Bigg) \\
& \cdot\Bigg[\frac{2(-x_1+2x_2-x_3)}{(2+b)^2}I_{xx}\Bigg(\frac{x_1+x_3+x_2b}{2+b},\frac{y_1+y_3+y_2b}{2+b}\Bigg) \\
& \quad + \frac{2(-y_1+2y_2-y_3)}{(2+b)^2}I_{xy}\Bigg(\frac{x_1+x_3+x_2b}{2+b},\frac{y_1+y_3+y_2b}{2+b}\Bigg)\Bigg] \\
& + 2I_y\Bigg(\frac{x_1+x_3+x_2b}{2+b},\frac{y_1+y_3+y_2b}{2+b}\Bigg) \\
& \cdot\Bigg[\frac{2(-x_1+2x_2-x_3)}{(2+b)^2}I_{xy}\Bigg(\frac{x_1+x_3+x_2b}{2+b},\frac{y_1+y_3+y_2b}{2+b}\Bigg) \\
& \quad + \frac{2(-y_1+2y_2-y_3)}{(2+b)^2}I_{yy}\Bigg(\frac{x_1+x_3+x_2b}{2+b},\frac{y_1+y_3+y_2b}{2+b}\Bigg)\Bigg] \\
& + 2I_x\Bigg(\frac{3x_1+x_3+\sqrt{3}x_2b}{4+\sqrt{3}\,b},\frac{3y_1+y_3+\sqrt{3}y_2b}{4+\sqrt{3}\,b}\Bigg) \\
& \cdot\Bigg[\frac{4\sqrt{3}(-3x_1+4x_2-x_3)}{(4+\sqrt{3}\,b)^2}I_{xx}\Bigg(\frac{3x_1+x_3+\sqrt{3}x_2b}{4+\sqrt{3}\,b},\frac{3y_1+y_3+\sqrt{3}y_2b}{4+\sqrt{3}\,b}\Bigg) \\
& \quad + \frac{4\sqrt{3}(-3y_1+4y_2-y_3)}{(4+\sqrt{3}\,b)^2}I_{xy}\Bigg(\frac{3x_1+x_3+\sqrt{3}x_2b}{4+\sqrt{3}\,b},\frac{3y_1+y_3+\sqrt{3}y_2b}{4+\sqrt{3}\,b}\Bigg)\Bigg] \\
& + 2I_y\Bigg(\frac{3x_1+x_3+\sqrt{3}x_2b}{4+\sqrt{3}\,b},\frac{3y_1+y_3+\sqrt{3}y_2b}{4+\sqrt{3}\,b}\Bigg) \\
& \cdot\Bigg[\frac{4\sqrt{3}(-3x_1+4x_2-x_3)}{(4+\sqrt{3}\,b)^2}I_{xy}\Bigg(\frac{3x_1+x_3+\sqrt{3}x_2b}{4+\sqrt{3}\,b},\frac{3y_1+y_3+\sqrt{3}y_2b}{4+\sqrt{3}\,b}\Bigg)
\end{aligned}$$

$$+\frac{4\sqrt{3}(-3y_1+4y_2-y_3)}{(4+\sqrt{3}\,b)^2} I_{yy}\left(\frac{3x_1+x_3+\sqrt{3}x_2 b}{4+\sqrt{3}\,b}, \frac{3y_1+y_3+\sqrt{3}y_2 b}{4+\sqrt{3}\,b}\right)\Bigg]\Bigg]\Bigg\} .$$

Examples

Consider the triangle with vertices $\mathbf{p}_1 = (1,0)$, $\mathbf{p}_2 = (1,1)$, and $\mathbf{p}_3 = (0,1)$ in Cartesian coordinates, and suppose $\beta = \gamma = 1$. Then $d_{12} = 1$, $d_{13} = \sqrt{2}$, $d_{23} = 1$, $c_{12} = 1$, and $\Delta = 1$. Also, let the image intensity be given by $I(x,y) = \tan^{-1}[k(x^2+y^2-3/2)]$, so that the image is radially symmetric about the origin, and the magnitude of its gradient is greatest on the circle $x^2+y^2 = 3/2$. Converting to BB coordinates, we have $I^*(\alpha_1,\alpha_2) = \tan^{-1}[k(\alpha_1+\alpha_2)^2+(1-\alpha_1)^2-3/2]$. Also let $w_{strain} = 1$ and $w_{image} = 1$. In order to find the value of b or b_{101} for which a quadratic spline $S(\alpha_1,\alpha_2) = 2b_{101}\alpha_1(1-\alpha_1-\alpha_2)-\alpha_2^2 = 0$ minimizes the total energy (7.1), it is necessary to find the value of b for which

$$\int_0^1 \left\{ \frac{[8u(1-u)+4b\sqrt{u(1-u)}+b^2(1-2u+2u^2)]^{1/2}}{2\sqrt{u(1-u)}[1+b\sqrt{u(1-u)}]^2} \right.$$
$$+\frac{2b^2[1+b\sqrt{u(1-u)}]^4}{\sqrt{u(1-u)}[8u(1-u)+4b\sqrt{u(1-u)}+b^2(1-2u+2u^2)]^{5/2}}$$
$$\left.-\frac{64k^2[1+b\sqrt{u(1-u)}]^6[-2u^2+2u-1-2b\sqrt{u(1-u)}+2b^2u(1-u)]}{\{4[1+b\sqrt{u(1-u)}]^4+k^2[-4u^2+4u+1+2b\sqrt{u(1-u)}-b^2u(1-u)]^2\}^2}\right\} du$$

or the equivalent expression using (6.14) and (6.20), is minimized. Using numerical integration, it is found that for $k = 0$, this value of b is 1.414, or equivalently $b_{101} = b^2/2 = 1.000$, and that the minimum energy is 3.142. Actually, it can be verified that b is exactly $\sqrt{2}$ and the minimum energy is π in this special case. Using the simplified energy (7.2), we need to find the value of b which minimizes the resulting expression, namely

$$\frac{b}{3\sqrt{2}}+\frac{2\sqrt{2}}{3(2+b)}+\frac{64(56+32\sqrt{7}b+50b^2)^{1/2}}{3\sqrt{7}(8+\sqrt{7}\,b)^2}$$
$$+\frac{2\sqrt{2}}{3b^3}+\frac{\sqrt{2}b^2}{3(2+b)}+\frac{256b^2(8+\sqrt{7}\,b)^4}{3\sqrt{7}(56+32\sqrt{7}b+50b^2)^{5/2}}$$
$$-\frac{1}{6}\left\{\frac{128k^2}{(4+k^2)^2}+\frac{512k^2(1+b)^2(2+b)^6}{[4(2+b)^4+k^2(b^2-4b+8)^2]^2}\right\} ,$$

is found to be $b = 1.428$, or equivalently $b_{101} = 1.020$, and the minimum simplified energy is 3.161 . This example is illustrated in Figure 7.2(a). The Cartesian equations of the exact and approximate splines are $x^2+y^2-1 = 0$ and $x^2-0.040xy+y^2+0.040x+0.040y-1.040 = 0$, respectively. These two curves are virtually indistinguishable. Their greatest separation, along the line $y = x$, is just 0.0012.

Changing the value of k to 1, we expect the contour to be attracted to the circle $x^2+y^2 = 3/2$, which lies outside the circle $x^2+y^2 = 1$. Indeed, the minimum energy now occurs when $b = 3.862$, or $b_{101} = 7.458$, and the minimum energy is -0.522. The negative value is a consequence of the definition of the edge energy functional as $-\|\nabla I\|^2$; this no longer represents an actual energy, but rather just compares energies of different image intensity fields. If one wants this to have a positive value, one can define the edge energy functional as $\|\nabla I(x,y)\|^2$ subtracted from the largest value

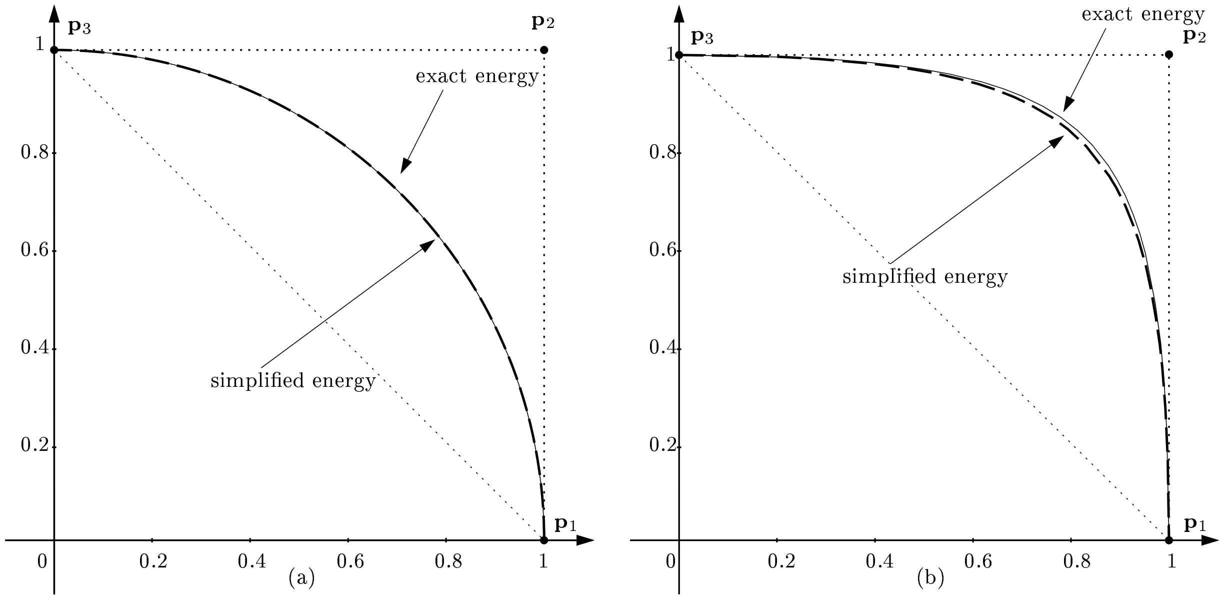

Figure 7.2: Quadratic splines for the exact energy and the simplified energy for the image $I(x, y) = \tan^{-1}[k(x^2 + y^2 - 3/2)]$ when (a) $k = 0$, (b) $k = 1$. In both figures, the exact energy is given by the thin solid line and the simplified energy is given by the thick dashed line.

$\|\nabla I\|^2$ takes on over all (x, y). The greater value of b_{101} means that the spline contour is pushed towards the corner $(1, 1)$ in Cartesian coordinates, or $(0, 1)$ in BB coordinates.

In this case the simplified energy (7.2) is minimized when $b = 3.607$, or $b_{101} = 6.505$, and the minimum simplified energy is -0.378. While these values for b and b_{101} are a bit off from the true values, the curves they represent are quite similar. Specifically, the greatest separation between the two, along the line $y = x$, is 0.0078. The Cartesian equations of the exact and approximate splines are $x^2 - 12.916xy + y^2 + 12.916x + 12.916y - 13.916 = 0$ and $x^2 - 11.010xy + y^2 + 11.010x + 11.010y - 12.010 = 0$, respectively. These are displayed in Figure 7.2(b).

Figures 7.3 and 7.4 illustrate how an energy minimizing A-spline changes shape as the ratio of β/γ changes. Figure 7.3 is an open polygon, while Figure 7.4 is a closed non-convex polygon. As the ratio β/γ increases, the A-spline turns sharper corners at the junction points and sticks more closely to the piecewise linear curve connecting these points.

The next examples are of CT scans of a vertebra column, taken at intervals of 20 millimeters(?). The initial control points are entered by hand, and the algorithm is run for all the contours in each slice. In these examples we have taken $w_{strain} = w_{image} = 1$ and $\beta = \gamma = 1$. The results are shown in Figure 7.4. The control polygons are not shown in Figure 7.4(b), (d), (f), (h) as they would clutter up the figures too much.

8 Conclusion

We have presented an algorithm for the active contouring of images which incorporates significant improvements. By using A-splines to represent contour curves, we can use polynomials of lower degree than those of traditional splines to achieve the same order of continuity. These physical