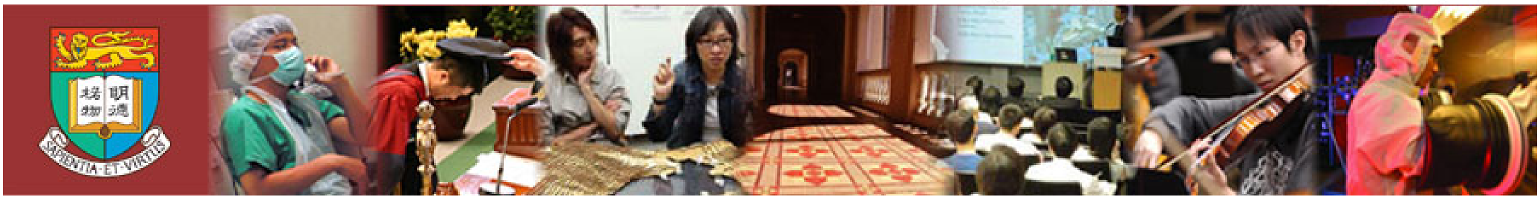

B-SPLINE SNAKES IN TWO STAGES

C.C. Leung, C.H. Chan, F.H.Y.Chan and W.K. Tsui

Department of Electrical & Electronic Engineering, The University of Hong Kong. Hong Kong, PRC.

Abstract

In using Snake algorithms, the slow convergence speed is due to the large number of control points to be selected, as well as difficulties in setting the weighting factors that comprise the internal energies of the curve. Even in using the B-Spline snakes, splines cannot be fitted into the corner of the object completely. In this paper, a novel two-stage method based on B-Spline Snakes is proposed. It is superior both in accuracy and fast convergence speed over previous B-Spline Snakes. The first stage reduces the number of control points using potential function V(x,y) minimization. Hence, it allows the spline to quickly approach the minimum energy state. The second stage is designed to refine the B-Spline snakes based on the node points of the polynomials without knots. In other words, an elasticity spline is controlled by node points where knots are fixed. Simulation and validation of results are presented. Compared to the traditional B-Spline snakes, better performance was achieved using the method proposed in this paper.

1. Introduction

Since the active contour model (snake) was first presented by Kass as an algorithm for locating the boundary of interest in images [1], many snake-based algorithms have been developed that can be used in applications of edge detection, computer vision, three-dimensional reconstruction, and medicine [2-4]. These types of snake are introduced as an energy minimization curve guided by external and internal forces toward image features that are of interest. However, the following problems have arisen when using the above algorithms.

1). Convergence is relatively slow because of the large number of coefficients to optimize;

2). Difficulty in determining the values of coefficients in the internal force term;

3). Poor resolution and accuracy of the curve due to a set of disconnected points;

4). In noisy environments, high-order derivatives on the discrete curve may not be accurate.

The B-Spline snake was proposed based on the philosophy of the snake [5-8]. It is mainly characterized by a few parameters, as well as smoothness of the curve determined by the B-Spline polynomial. Wang *et al.* introduced a multistage, optimal snake with B-Spline [9] that minimized the energy both locally and globally. Brigger *et al.* proposed a flexible tool for parametric B-Spline snakes [10]; this work extended the basic concept of B-Spline snakes in order to improve its efficiency, speed, and applicability in an interactive environment. The node points coinciding to the actual curve is the important difference to other B-Spline snakes.

Our work is based on the concept of B-Spline snakes. In order to improve the convergence speed of the spline and its efficiency, our method retains one parameter without the parameters of α, β, and C_i which are used to force the spline on local features of interest. The main contributions of this paper are as follows. First, the number of control points that we use is less than other B-Spline snakes. As we know, splines are piecewise polynomials with pieces that are smoothly connected together. However, it cannot approximate curve segments with high curvature unless a large number of control points are selected near the high curvature region. Second, we use node points $p(r\cos\theta, r\sin\theta)$ that can move to the feature of interest based on the displacement *r* and the angles *θ*. The destination of each node point depends on energy minimization. Interpolation is the last step to approximate the curve segment with the feature of interest.

The paper is organized as follows. Section II gives a brief overview of B-Spline snakes. Section III introduces our two stages approach and describes it is improvement for a good fit in a corner or any region of high curvature. The simulation results are given in Section IV, and conclusions are given in Section V.

2. Overview of B-Spline Snakes

In this section, B-Spline Snakes will be reviewed in order to justify the use of splines for solving snake problems.

A snake is a deformable model whose shape is controlled by internal forces and external forces using energy minimization. Let $s(t) = (x(t), y(t))$ represent the parametric position of a snake. We can write its energy function as shown in equation (1).

$$E_{snake} = \int_{snake} E(s(t))dt = \int_{snake} [E_{\text{int}}(s(t)) + E_{ext}(s(t))]dt \tag{1}$$

where

$$E_{\text{int}}(s(t)) = \frac{\left(\alpha(t) \left| \frac{\partial s(t)}{\partial t} \right|^2 + \beta(t) \left| \frac{\partial^2 s(t)}{\partial t^2} \right|^2 \right)}{2} \tag{2}$$

The internal force of the snake defines its physical properties, and the external force deforms the contour from the initial position to the feature of interest in an image. Then, in B-Spline snakes, $s(t)$ is partitioned into segments. Each curve segment is approximated by the piecewise polynomial that is obtained by the basic function of B-Spline, $b_m^n(t-k)$, and a set of control points ($c_k = (x_k, y_k)$), where k is the number of control points and n is the degree of polynomial. Then,

$$s(t) = \sum_{k=1}^{N} c(k) b_1^3 (t-k) \tag{3}$$

To minimize the total energy of the snake shown in equation 3, methods such as a multistage, optimal snake, and the multiresolution approach [11] were proposed so that the B-Spline snakes can fit into any region of high curvature and can improve the speed of convergence.

3. Optimal B-Spline snakes

In order to improve the accuracy of B-Spline snakes to fit into a corner feature of interest, and also to increase the convergence speed, an optimal two-stage B-Spline snake is proposed.

3.1 The first stage: Moving Knots of the B-Spline to the edge

In previous studies of B-Spline snakes, their control points (knots) correspond to the B-Spline coefficients, $\{c(k) = (x_k, y_k), k = 1,2,..., N\}$, where N is the number of control points. Fundamentally, each curve segment is defined by four control points. Each control point affects four curve segments. Moving a control point in a given direction will affect the four curve segments that move in the same direction. However, other curve segments are not influenced. The energy of the snake is minimized by iteratively adjusting the two most effective control points.

The idea of our work is not the same as described above. Individual control points will move individually to the edge according to the local energy minimization. Let the control points $\{p_0(x_0, y_0), p_1(x_1, y_1), ..., p_N(x_N, y_N)\}$ enclose the features of interest. Each point has been moved by the energy difference based on displacement, *r,* and angles, *θ*, where $x = r\cos\theta$ and $y = r\sin\theta$ (r =1, and θ =0 to 2π). For example, each control point can represent a pixel located in the center of a 3x3 image matrix which is shown in Figure 1. The energy difference between that pixel and surrounding pixels can be calculated individually. Each control point is moved toward the neighboring pixel with a higher energy difference than any of the other seven neighboring pixels. Hence, $p_0(x_0, y_0)$ is moved if:

$$E_{\min}(p_0) = \begin{cases} E_{\min}[(x_k, y_k) \mid (x_{k-1}, y_{k-1})] > 0, and \\ E_{\min}[(x_k, y_k) \mid (x_{k-1}, y_{k-1})] > E_{\min}[(x_k, y_k) \mid (x_{k-n}, y_{k-n})] \end{cases} \tag{4}$$

where n=2,3,…,8., and $E_{\min}[(x_k, y_k) \mid (x_{k-1}, y_{k-1})]$ denotes the local energy from the kth control point to its neighboring (k-1)th point.

When all control points are moved to the expected location (normally they are located on the boundary of feature of interest), B-Spline interpolation is generated based on the control points for curve rendering. To interpolate the curve from points $p_n,$ we have

$$S(x(j)) = \sum_{i=1}^{p-1} \sum_{j=1}^{q-1} \left\{ \frac{[S(y(i)) - y(i)][x(i) - x(i-1)]}{y(i) - y(i-1)} + x(i) \right\} \tag{5}$$

where p is the number of control points and q is the total number of segments $S(x,y)$.

3.2 The Second stage: Fine tuning in moving node points between polynomial Splines to the edge

The cubic B-Spline (n=3) can generate a smooth curve based on the polynomial Splines, and node points are part of the snake that correspond to the knots of the spline curve. However, control points cannot fit to the actual coordinates of the spline curve, especially in a spline of higher degree than 3 $(n \geq 3)$. The node points are another equivalent representation of the spline curve and related to the control points based on the linear equation 5. We will briefly show how to fit the node points onto the feature of interest without influencing the control points.

We impose a set of N discrete node points $n(k) = (n_x(k), n_y(k)), 0 \leq k < N$ to be part of the segments, $S(x,y)$. Then, each individual node point in each segment will move to the edge according to the local energy minimization. Now, each node point is located beyond the boundary of features of interest, and they move toward to the boundary according to two factors, the displacement *r'* and rotation angle *θ'*. Here, the coordinate of each node point $n(k)$ is represented by $n_x = r'\cos\theta'$ and $n_y = r'\sin\theta'$ (r' =1, and θ' =0 to 2π). In order to search for the best position of all node points, they are represented by a pixel located in the center of a 3x3 matrix block, as described in the previous section. The energy differences between that node point and its surrounding eight pixels can be calculated individually. The position that a node point is moved to depends on the difference in energy levels compared to those of the other seven neighboring pixels.

$$E_{\min}(n(k)) = \begin{cases} E_{\min}\left[(N_x(r),N_y(r)) \mid (N_x(r-1),N_y(r-1))\right] > 0, and \\ E_{\min}\left[(N_x(r),N_y(r)) \mid (N_x(r-1),N_y(r-1))\right] > E_{\min}\left[(N_x(r),N_y(r)) \mid (N_x(r-m),N_y(r-m))\right] \end{cases} \quad (6)$$

where

$N_x(r) = n_x(r'\cos\theta, r'\sin\theta)$,

$N_y(r) = n_y(r'\cos\theta, r'\sin\theta)$,

$N_x(r-m) = [n_x(r'-m)\cos\theta, n_y(r'-m)\sin\theta]$

and $N_y(r-m) = [n_x(r'-m)\cos\theta, n_y(r'-m)\sin\theta]$, m=2,3,…,8., and $E_{\min}\left[(x_k, y_k) \mid (x_{k-1}, y_{k-1})\right]$ denotes the local energy from the kth control point to its neighboring (k-1)th point. For curve rendering, B-Spline interpolation shown in equation 5 can be used to fit in the line segment based on node points.

4. Simulation Results and Discussion

Results of two sets of simulated data are reported in this section. The first example is a model containing rectangles, polygon, and ellipse, which are shown in figure 2a. Figure 2b shows the curve fitting results obtained using B-Spline snakes, where line segments of interconnecting the shapes of the model do not fit perfectly. Results obtained using proposed method are shown in Figure 2c, where line segments closely fit the hard corners of the model. It is clear that the proposed method demonstrates excellent performance at the corners.

The second example is an image of an aircraft that was scanned from Wang *et al.* [9]. Figure 3a shows that the aircraft image is a high curvature model. Figure 3b illustrates the result of using a B-Spline snakes, which has not been good at fitting the corners of the aircraft such as at positions marked by the arrows. However, the proposed approach has been improved the fitting at the corner shown in Figure 3c. Furthermore, only three control points were selected for each corner of the aircraft image, compared to five points used for the original B-Splines snake.

Using the proposed method, it is no need to concentrate in plotting many control points beyond the object of interest manually, and the best curve fitting of Spline is not depended on the number of control points. As a small number of control points fit into the boundary of object based on the energy minimization technique (snake function), node points have been generated by cubic Spline. Then, each node point has been fitted into the boundary of the object according to the snake function automatically. However, in order to fit to the corner by Spline, the proposed approach needs to plot three control points beyond the corner of the object. Basically, a half of control points used in this new approach, and the best fit in the corner of interest of object compared to the original B-Spline snakes are its advantage.

5. Conclusion

This paper proposes improved B-Spline snakes with two stages. Less control points are needed in our approach than the B-Spline snakes to fit the edge. Moreover, the accuracy of curve fitting, especially at sharp corners of the object, is better than the original B-Spline snakes.

References

[1] M. Kass, A. Witkin, and D. Terzopoulos, 'Snakes: Active contour model', Proc. First Int. Conf. on Computer Vision, pp.259-269, 1987.

[2] H. Eviatar and R. Somorjai, 'A fast, simple active contour algorithm for biomedical images', Pattern Recognition Letters, vol.17, pp.969-974, 1996.

[3] T. McInerney and D. Terzopoulos, 'Topologically adaptive snakes', 5th Int. Conf. Computer Vision, pp.840-845, 1995.

[4] H. Tang, and T. Zhuang, 'An improved Adaptive B-Spline Active Contour Model', Proc. 20th Annual Int. Conf. of IEEE EMBS, Vol.20, no.2, pp.990-993, 1998.

[5] F.H.Y. Chan, F.K. Lam, P.W.F. Poon, H. Zhu, and K.H. Chan, 'Object boundary location by region and contour deformation', IEE Proc. Vis. Image Signal Process, 143, no.6, pp.353-360, 1996.

[6] K.M. Lam, and H. Yan, 'Fast greedy algorithm for active contours', Electronic Letters, vol.30, no.1, 1994.

[7] S. Menet, P. Saint-Marc, and G. Medioni, 'B-snakes: Implementation and application to stereo', Image Understanding Workshop, pp.720-726, 1990.

[8] P. Saint-Marc, H. Rom, and G. Medioni, 'B-spline contour representation and symmetry detection', IEEE Trans. Pattern Anal. Machine Intell., vol.15, pp.1191-1197, 1993.

[9] M. Wang, J. Evans, L. Hassebrook, and C. Knapp, 'A Multistage, Optimal Active Contour Model', IEEE Trans. on Image Processing, vol. 5, no.11, pp.1586-1591, 1996.

[10] P. Brigger, J. Hoeg, and M. Unser, 'B-Spline Snakes: A Flexible Tool for Parametric Contour Detection', IEEE Trans. on Image Processing, vol. 9, no. 9, pp.1484-1496, 2000.

[11] A. Rosenfeld, 'Multiresolution Image Processing', New York: Academic, 1971.

x_{k-5},y_{k-5}	x_{k-6},y_{k-6}	x_{k-7},y_{k-7}
x_{k-4},y_{k-4}	x_k,y_k	x_{k-8},y_{k-8}
x_{k-3},y_{k-3}	x_{k-8},y_{k-2}	x_{k-1},y_{k-1}

Figure 1. Control point x_k,y_k with its eight surrounding pixels in 3x3 matrix.

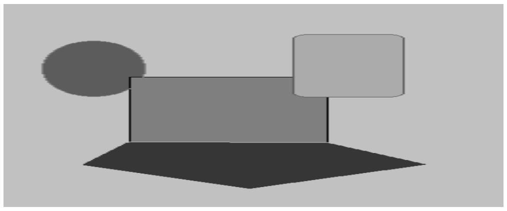

Figure 2a. The simulation model includes two rectangles, a polygon, and an ellipse.

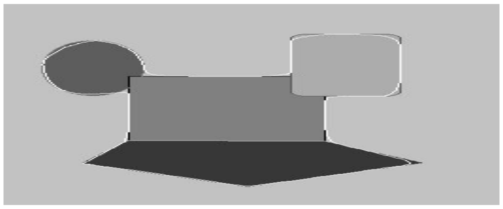

Figure 2b. The outline obtained using B-Spline snakes. Some of the hard corners are not fitted perfectly.

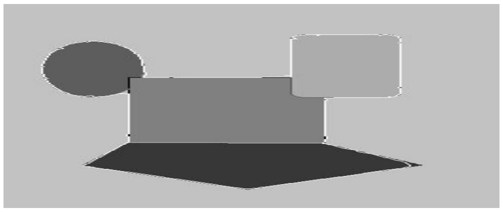

Figure 2c. The outline obtained using the two-stage B-Spline snakes approach. Many more of the hard corners have been fitted perfectly.

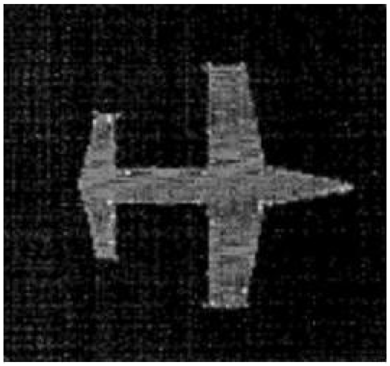

Figure 3a. The original aircraft image.

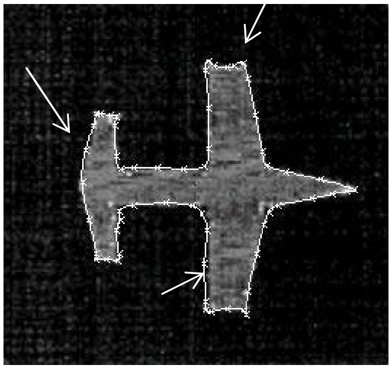

Figure 3b. The outline of the aircraft image generated using B-Spline snakes. The arrows indicate where the spline cannot be fitted perfectly.

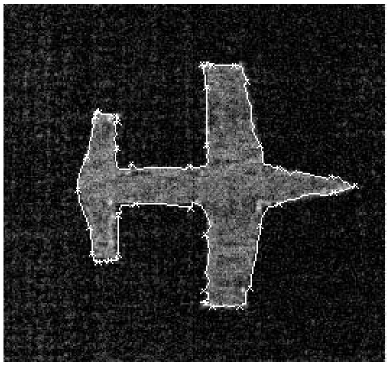

Figure 3c. The outline of the aircraft image generated using the two-stage B-Spline snakes. Spline fitting has been improved.