

Control Point Adjustment for B-Spline Curve Approximation

Huaiping Yang^a Wenping Wang^b Jianguang Sun^c

^a*The University of Hong Kong & Tsinghua University, China*

^b*The University of Hong Kong, Hong Kong*

^c*Tsinghua University, China*

Abstract

Pottmann et al propose an iterative optimization scheme for approximating a target curve with a B-spline curve based on *square distance minimization*, or SDM. The main advantage of SDM is that it does not need a parameterization of data points on the target curve. Starting with an initial B-spline curve, this scheme makes an active B-spline curve converge faster towards the target curve and produces a better approximating B-spline curve than existing methods relying on data point parameterization. However, SDM is sensitive to the initial B-spline curve due to its local nature of optimization. To address this, we integrate SDM with procedures for automatically adjusting both the number and locations of the control points of the active spline curve. This results in a method that is more robust and applicable than SDM used alone. Furthermore, it is observed that the most time consuming part of SDM is the repeated computation of the foot-point on the target curve of a sample point on the active B-spline curve. In our implementation, we speed up the foot-point computation by pre-computing the distance field of the target curve using the Fast Marching Method. Experimental examples are presented to demonstrate the effectiveness of our method. Problems for further research are discussed.

Keywords: B-spline curve, shape approximation, optimization, squared distance

1 Introduction

1.1 B-spline curve fitting problem

The B-spline curve fitting problem is to produce a B-spline curve to approximate a *target curve* within a pre-specified tolerance. We assume that the *target curve* is defined in 2D plane by a sequence of ordered dense data points or a curve given implicitly or explicitly (i.e. by a parametric equation).

There is much work in the literature on solving the curve-fitting problem using the B-spline curve. However, most of the existing methods require a parameterization of data points on the target curve, and little consideration has been given to the automatic placement of a minimal number of control points of an approximating B-spline curve satisfying a pre-specified error tolerance. It is beyond the scope of the present paper to give a detailed review of all the existing work; we will only review those most relevant results. The reader may consult [6] for a general introduction to the curve-fitting problem.

1.2 Methods based on data parameterization

A parameterization of target data points is often needed for B-spline curve approximation [4] [6]. Let P_k , $k = 1, 2, \dots, N$, denote a sequence of ordered data points on the target curve Γ . Let $X(u) = \sum_{i=1}^n B_i(u)D_i$ denote a B-spline curve to be determined to approximate the target curve Γ , where $B_i(u)$ are the B-spline basis functions of a certain order and D_i are control points. Suppose that, through some method [17], the data points P_k are made to correspond to the parameter values u_k in the domain of the curve $X(u)$. Then the usual approach to determining the control points D_i is based on a least squares formulation and computes the minimizer of an objective function

$$F = \sum_k \|X(u_k) - P_k\|^2 + \lambda F_s, \quad (1)$$

where F_s , called the *regularization term*, is quadratic in the control points D_i and is used to enforce the fairness of the final approximating curve [1] [2] [8] [23]. Therefore, the minimizer of F can be found by solving a system of linear equations.

Assigning parameter values to data points, a procedure to be referred to as *data parameterization*, is unnatural for measuring the geometric difference between a target curve and its approximating B-spline curve; a suitable measurement of the difference between two curves should be defined by a distance between two point sets, such as the Hausdorff distance [7]. Different ways of data parameterization lead to different approximation results. Fundamentally, the notion of the optimal data parameterization is theoretically elusive and computationally difficult to achieve. An improper data parameterization may considerably compromise the quality and efficiency of approximation. The following three methods for data parameterization have been proposed [15] [23] [24]: uniform parameterization, chord-length parameterization, and centripetal parameterization. Some recent research aims at correcting assigned parameter values iteratively to improve the approximation error [5] [11] [12] [13] [19]. Such correction procedures are rather time-consuming, typically taking 100-300 iterations to converge [12].

1.3 Squared distance minimization (SDM)

Pottmann et al [17] propose an optimization scheme for B-spline curve approximation that does not require data parameterization. Starting with an initial B-spline curve, this method attempts to make an active B-spline curve converge towards the target curve by minimizing an objective function defined by local approximate squared distances of the target curve. For brevity, this scheme will be referred to as *squared distance minimization*, or SDM.

The main idea of SDM is as follows. For each iteration, one computes the foot-points O_k on the target curve of a number of sample points X_k on the active B-spline curve. Here we treat the X_k as variable points expressed as linear combinations of the control points D_i . (The number of sample points per piece of the B-spline curve is a user-specified parameter; we normally use 10 sample points on each piece of a cubic B-spline curve.) Let X_k^0 denote the current location of the point X_k . Then the local approximate squared distance of the point X_k to the target curve is given by

$$F_d^+(X_k) = \frac{d}{d + |\rho|} x_1^2 + x_2^2, \quad (2)$$

where x_1 and x_2 are the coordinates of X_k with respect to the Frenet frame of the target curve at the foot-point O_k , $d = \|X_k^0 - O_k\|$, and ρ is the curvature radius of the target curve at O_k . (See Figure 1. The reader is referred to [18] for a detailed derivation and discussion of this formula.) Then the control points D_i of an approximant B-spline curve are computed by minimizing the objective function

$$F = \sum_k F_d^+(X_k) + \lambda F_s, \quad (3)$$

which is quadratic in the D_i , and thus can be solved using the quasi-Newton method.

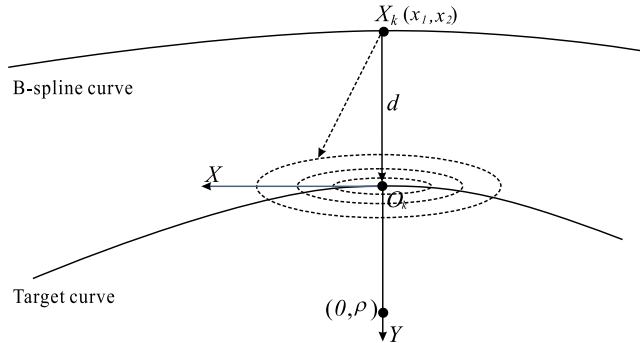


Figure 1. The local squared distance used in SDM.

SDM converges very fast. Our tests show that, with an appropriately specified initial B-spline curve, no more than twenty iterations are needed for the convergence of SDM in most cases. Furthermore, due to its independence of data parameterization, SDM produces better approximating B-spline curves than those methods relying on data parameterization, since SDM makes the sample points on the active B-spline curve move towards the target curve, rather than simply towards their corresponding foot-points.

However, because SDM is based on local optimization, its approximation results can be very sensitive to the initial B-spline curve. We explain this problem by the example in Figure 2, where the target curve is shown as the dotted curve and the initial B-spline curve as the solid curve. With the initial control points shown in Figure 2(a), SDM produces the approximating curve in Figure 2(b), in which there are redundant control points on the right part of the B-spline curve, but too few control points on the left part to warrant an acceptable approximation error. In contrast, if a different initial B-spline curve with the same number of control points as shown in Figure 2(c) is used, SDM produces the approximating curve in Figure 2(d), which is clearly more acceptable than the approximating curve in Figure 2(b).

Another example is shown in Figure 3. Here, by applying SDM alone, the two different but similar initial B-spline curves (Figure 3(a) and (c)) lead to two radically different approximation results (Figure 3(b) and (d)). These examples show clearly the importance of specifying an appropriate initial B-spline curve or providing an automatic mechanism of adjusting the control points of an active B-spline curve on-the-fly in order to obtain a satisfactory approximating curve. This provides the motivation for our work on integrating SDM with procedures for control point adjustment.

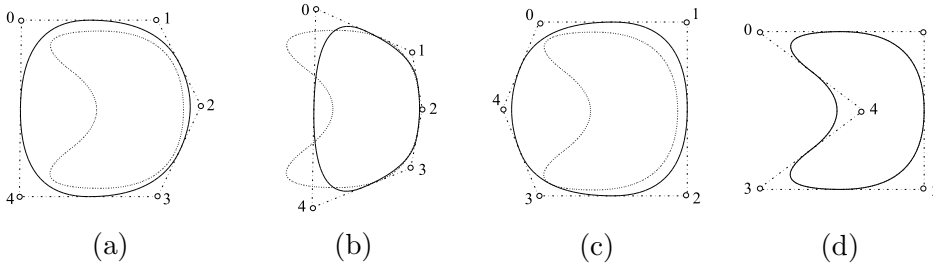


Figure 2. Effects of different initial B-spline curves. (a) An initial order-3 B-spline curve; (b) Unsatisfactory approximation generated by SDM from the control points in (a); (c) An initial B-spline curve different from that in (a); (d) Better approximation generated by SDM from the control points in (c).

1.4 Outline of our method

In practice, it is extremely difficult, if not impossible, to specify an initial B-spline curve a priori with a suitable number of control points distributed appropriately so as to yield a satisfactory approximating curve. Therefore, we

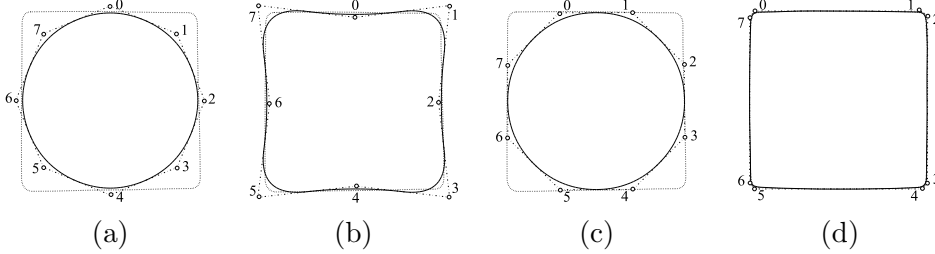


Figure 3. Two different initial B-spline curves of order-3 with 8 control points in (a) and (c) are used to approximate the target curve defined by $x^{20} + y^{20} = 1$. The initial control points in (a) and (c) are evenly distributed on a circle, and differ only in orientation. Their corresponding final approximating B-spline curves generated by SDM are shown in (b) and (d), respectively.

present a new method for adjusting the control points of the active B-spline curve. Our goal is, through the addition of new control points or removal of redundant control points, to generate a B-spline curve with as few control points as possible that approximates a given target curve to within a pre-specified error tolerance.

More specifically, given a target curve and an initial B-spline curve, which is normally specified by the user, we firstly use SDM of Pottmann et al to update the initial B-spline curve to get a B-spline curve C approximating the target curve. Then the approximation error of each segment of the B-spline curve C is evaluated. If a segment of the curve C has a large error, then this can be attributed to that the segment does not have enough control points. Therefore, we insert new control points one by one to this segment till the approximation error of the segment is reduced to within the error tolerance. On the other hand, if we detect that a segment of the curve C has redundant control points, we remove the control points of the segment one by one till the control points cannot be removed anymore without making the error larger than the error tolerance. Once the control points have been adjusted as above, SDM is applied again to produce a new approximating B-spline curve. This procedure is repeated until a satisfactory result is obtained. Figure 4 shows an example using this method.

In addition, we observe that the most time-consuming part of SDM is the repeated computation of the foot-point on the target curve of a sample point on the active B-spline curve. In our method we speed up this computation by pre-computing the discrete distance field of the target curve using the Fast Marching Method [22], which is originally proposed in the framework of the level-set method. This improvement of efficiency makes our method more applicable in practice.

The remainder of this paper is organized as follows. In Section 2 we discuss our strategy of adjusting the control points, i.e. the insertion of a new control point and the removal of redundant control points. In Section 3, we give an outline of our complete algorithm for B-spline curve approximation and discuss the

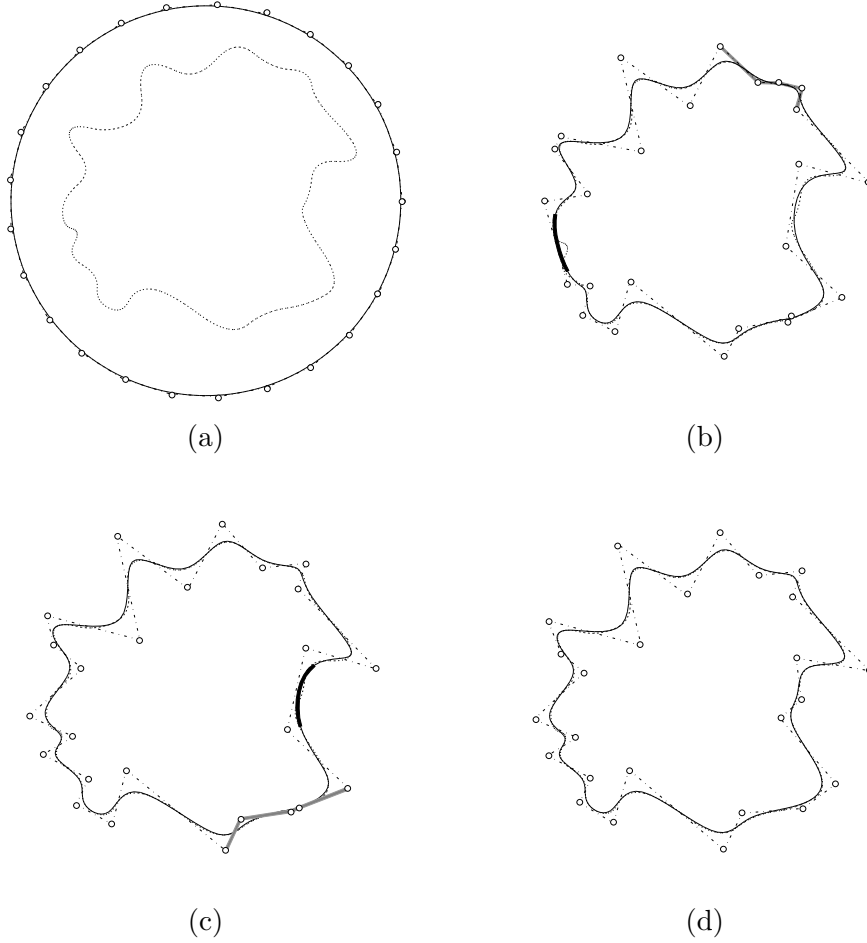


Figure 4. Adjusting the distribution of control points. (a) Initial control points of an order-4 B-spline curve (solid) and a target curve (dashed); (b) The approximation result by SDM; (c) The approximation result after control point adjustment; (d) the final result after further adjustment. The thick gray lines denote the part containing redundant control points and the thick black lines denote the B-spline piece with a large local error, i.e. where control point insertion is needed.

specification of an initial B-spline curve. In Section 4, we discuss how to use the pre-computed distance field of a target curve to speed up the foot-point computation. We present experimental results in Section 5 and conclude the paper in Section 6 with discussions and problems for further research.

2 Adjustment of Control Points

In this section we discuss how to adjust the control points of an active B-spline curve using SDM so as to obtain a B-spline curve that approximates a given target curve to within a pre-specified error tolerance ϵ_0 and has as few control points as possible. There are two parts in this adjustment strategy:

- Select and remove locally redundant control points;

- Insert new control points to a segment of the active B-spline curve whose approximation error is large and cannot be further reduced by optimization due to the lack of degree of freedom.

We assume that the knot sequence of the B-spline curve is fixed and is independent of the shape of the target curve. Uniform or uniform-periodic B-spline curves are used in all examples presented in this paper.

2.1 Removal of redundant control points

Each piece of a K -th order B-spline curve is dependent on K consecutive control points [16]. Suppose that an active B-spline curve C has n control points. Let E_j be the current approximation error of the j -th piece L_j of the piecewise B-spline curve C . Then E_j is defined by

$$E_j = \frac{1}{m} \sum_{i=1}^m d_i, \quad (4)$$

where m is the number of sample points X_i on L_j and d_i is the distance between X_i and its foot-point O_i on the target curve Γ . The approximation error of the whole curve C is defined by

$$E = \frac{1}{N_L} \sum_{j=1}^{N_L} E_j, \quad (5)$$

where N_L is the total number of pieces of the B-spline curve C .

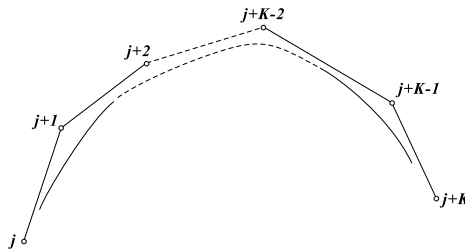


Figure 5. The set of $K + 1$ consecutive control points.

We first consider how to detect a piece of the curve C that contains redundant control points, and then how to remove redundant control points from such a piece. Let \mathcal{Q}_j denote the set of $K + 1$ consecutive control points $D_j, D_{j+1}, \dots, D_{j+K-1}, D_{j+K}$ that define two consecutive pieces of the B-spline curve C . Since the approximation error around \mathcal{Q}_j should be small if \mathcal{Q}_j contains redundant control points, the following criterion is used to detect the redundancy of control points. Define the weighted error associated with the

set \mathcal{Q}_j by

$$E_w(\mathcal{Q}_j) = \frac{1}{\sum_{L_i \in \mathcal{H}_j} \omega_i} \sum_{L_i \in \mathcal{H}_j} \omega_i E_i, \quad (6)$$

where \mathcal{H}_j denotes the set of B-spline curve pieces that are affected by the control points in \mathcal{Q}_j . Let n_i be the number of control points in \mathcal{Q}_j that affect L_i . The weight ω_i is defined by

$$\omega_i = \frac{n_i}{K}.$$

Therefore ω_i is set to be proportional to n_i , reflecting that the weight becomes smaller for a piece L_i farther away from the two pieces controlled entirely by the control points in \mathcal{Q}_j .

The error $E_w(\mathcal{Q}_j)$ is computed for each group \mathcal{Q}_j of $K + 1$ consecutive control points. Then a set \mathcal{Q}_j is regarded as *potentially redundant* if $E_w(\mathcal{Q}_j) < \epsilon_0$. If there exists no \mathcal{Q}_j whose weighted error satisfies this condition, then there are no redundant control points to remove. If there exist several groups \mathcal{Q}_j whose weighted errors satisfy the above condition, then all these groups will be further tested for possible removal of redundant control points, in a way to be explained shortly.

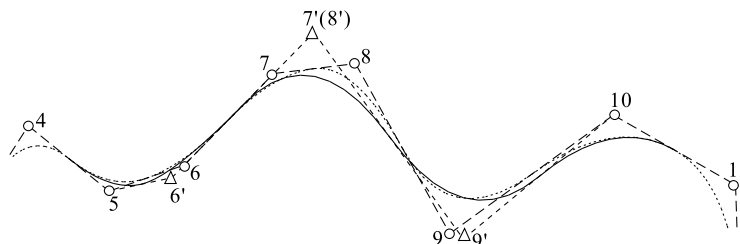


Figure 6. An example of removing a redundant control point from an order-3 B-spline curve. The group \mathcal{Q}_j is initially composed of four control points 6, 7, 8, 9, and is replaced by \mathcal{Q}'_j composed of three new control points 6', 7', 9'. The B-spline pieces in $\mathcal{Z}'_{\mathcal{Q}'_j}$ that are partially affected by \mathcal{Q}'_j include the pieces determined by control point set (4, 5, 6'), (5, 6', 7'), (7', 9', 10) and (9', 10, 11), which correspond to (4,5,6), (5,6,7), (8,9,10) and (9,10,11), respectively. The B-spline curve is dashed before the control point removal and solid after the removal. Note that the shape of the B-spline curve is well preserved by the removal of a redundant control point.

Now we consider how to remove the redundancy of a potentially redundant group \mathcal{Q}_j . The key idea is to replace the $K + 1$ control points in \mathcal{Q}_j by K properly distributed new control points through an optimization procedure. Let L_j and L_{j+1} denote the two adjacent B-spline pieces completely controlled by \mathcal{Q}_j , i.e. L_j is controlled by $D_j, D_{j+1}, \dots, D_{j+K-1}$, and L_{j+1} by $D_{j+1}, D_{j+2}, \dots, D_{j+K}$. Suppose that L_j and L_{j+1} are defined over the parameter intervals $[u_a, u_b]$ and $[u_b, u_c]$, respectively. Let \mathcal{Q}'_j denote the group of K

control points to replace the group \mathcal{Q}_j . Let L'_j be the B-spline piece controlled by \mathcal{Q}'_j . Suppose that L'_j is defined over the parameter interval $[u'_a, u'_c]$.

Let $\mathcal{Z}'_{\mathcal{Q}'_j}$ denote the set of B-spline pieces that are partially affected by control points $D'_j, D'_{j+1}, \dots, D'_{j+K-1}$ in \mathcal{Q}'_j ; there are no more than $2 \times (K - 1)$ such pieces. (See Figure 6 for an example of an order-3 B-spline curve.) Now we need to determine the distribution of the K new control points in \mathcal{Q}'_j so that the approximation error around \mathcal{Q}_j is increased as little as possible. To this end, we adopt the following two requirements:

- (1) L'_j , i.e. the piece of B-spline curve controlled by \mathcal{Q}'_j , should be close to the target curve. Note that the control points $D'_j, D'_{j+1}, \dots, D'_{j+K-1}$ in \mathcal{Q}'_j are unknown at this point and need to be determined. Suppose $2m$ points $X'(u'_i)$, $i = 1, 2, \dots, 2m$, are sampled evenly in the interval $[u'_a, u'_c]$. Then we use the following objective function defined by the approximate local squared distance formula (2) to measure the closeness between L'_j and the target curve:

$$F_{L'_j} = \sum_{i=1}^{2m} F_{d_i}^+(X'(u'_i)), \quad u'_i \in [u'_a, u'_c]. \quad (7)$$

Here, in order to define the distance functions $F_{d_i}^+(X'(u'_i))$, we get the corresponding points $X(u_i)$, $i = 1, 2, \dots, 2m$, on L_j and L_{j+1} of $X'(u'_i)$ through the following linear mapping from $[u'_a, u'_c]$ to $[u_a, u_c]$,

$$u_i = u_a + \frac{u_c - u_a}{u'_c - u'_a} \cdot (u'_i - u'_a).$$

Then we use the foot-point O_i of $X(u_i)$ and the distance $d_i = \|X(u_i) - O_i\|$ to define $F_{d_i}^+(X'(u'_i))$. (See Section 1.3.)

- (2) The B-spline pieces L'_k in $\mathcal{Z}'_{\mathcal{Q}'_j}$ should also be close to the target curve. Their difference is measured by the error term $\sum_{L'_k \in \mathcal{Z}'_{\mathcal{Q}'_j}} F_{L'_k}$. Here $F_{L'_k}$ measures the difference between L'_k and the target curve, defined by the sum of approximate local squared distances

$$F_{L'_k} = \sum_{i=1}^m F_{d_i}^+(X'(u'_i)), \quad u'_i \in [u'_{1,k}, u'_{2,k}], \quad (8)$$

where $[u'_{1,k}, u'_{2,k}]$ is the parameter interval of L'_k , and $X'(u'_i)$, $i = 1, 2, \dots, m$, are the m evenly sampled points on L'_k . Let L_k be the corresponding curve piece of L'_k before \mathcal{Q}_j is replaced by \mathcal{Q}'_j . Suppose that L_k is defined over $[u_{1,k}, u_{2,k}]$. Then we get the corresponding points $X(u_i)$, $i = 1, 2, \dots, m$ of $X'(u'_i)$ through the linear mapping

$$u_i = u_{1,k} + \frac{u_{2,k} - u_{1,k}}{u'_{2,k} - u'_{1,k}} \cdot (u'_i - u'_{1,k}), \quad u'_i \in [u'_{1,k}, u'_{2,k}].$$

Then the foot-point O_i of $X(u_i)$ and $d_i = \|X(u_i) - O_i\|$ are used to obtain the distance function $F_{d_i}^+(X'(u'_i))$.

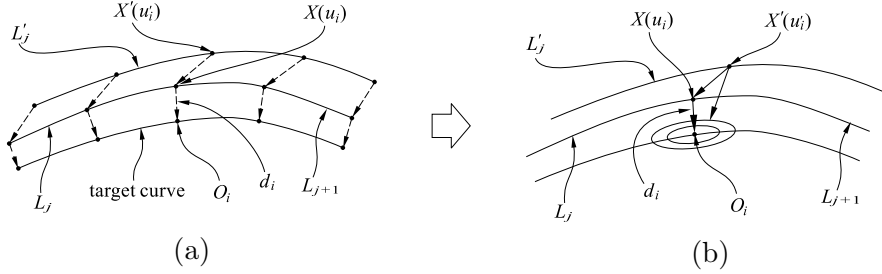


Figure 7. The relationship between $X'(u'_i)$, $X(u_i)$ and O_i .

Hence, by combining the two requirements above, the unknown control points $D'_j, D'_{j+1}, \dots, D'_{j+K-1}$ can be computed as the minimizer of the objective function

$$F_{\mathcal{Q}} = F_{L'_j} + \sum_{L'_k \in \mathcal{Z}'_{\mathcal{Q}'_j}} F_{L'_k}, \quad (9)$$

which is quadratic in $D'_j, D'_{j+1}, \dots, D'_{j+K-1}$.

When there are more than one group of control points \mathcal{Q}_j satisfying $E_w(\mathcal{Q}_j) < \epsilon_0$, the minimum value of $F_{\mathcal{Q}}$ is computed for each of these groups, and the group \mathcal{Q}_j with the smallest $F_{\mathcal{Q}}$ will be considered as the most redundant one and be replaced by \mathcal{Q}'_j , while the others are kept unchanged. Note that the removal of a potentially redundant control point is confirmed only if the approximation error still meets the error tolerance after the removal of that control point; otherwise, the removal will be reversed, i.e. no control point will be removed.

2.2 Insertion of new control points

When there are too few control points, either locally or globally, the active B-spline curves may not have enough flexibility to approximate a target curve of complex shape. In this case, new control points need to be inserted. We first need to determine if any piece of the active B-spline curve has a large approximation error that cannot be reduced by optimization, and, if such a piece of curve exists, a new control point is inserted to that piece. By inserting a control point, we mean replacing K consecutive control points by $K + 1$ appropriately distributed new control points such that the approximation error is reduced.

Again, the approximation error E_j of a piece L_j of the active B-spline curve is given by Eqn. (4). We select the segment L_{j_0} that gives the maximal error among all the pieces, i.e. $E_{j_0} = \max_j (E_j)$, and consider adding a new control

point to the piece L_{j_0} . Let $D_{j_0}, D_{j_0+1}, \dots, D_{j_0+K-1}$ be the K control points of L_{j_0} . Suppose that the K control points are replaced by $K + 1$ new control points $D'_{j_0}, D'_{j_0+1}, \dots, D'_{j_0+K-1}, D'_{j_0+K}$ whose locations are to be determined. Then the B-spline piece L_{j_0} is replaced by two new pieces L'_{j_0} and L'_{j_0+1} which have the control points $D'_{j_0}, D'_{j_0+1}, \dots, D'_{j_0+K-1}$ and $D'_{j_0+1}, D'_{j_0+2}, \dots, D'_{j_0+K}$, respectively, as shown in Figure 8.

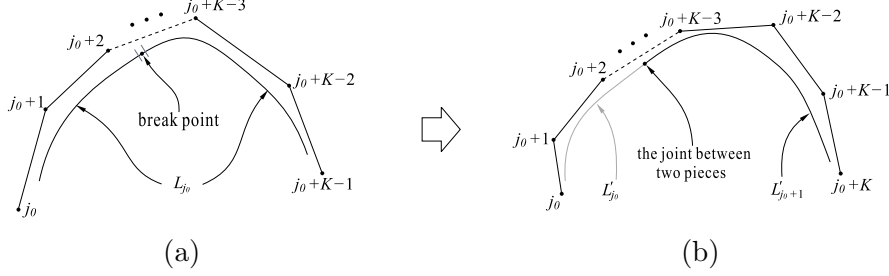


Figure 8. L_{j_0} is replaced by L'_{j_0} and L'_{j_0+1} after inserting a new control point.

In order to determine the control points $D'_{j_0}, D'_{j_0+1}, \dots, D'_{j_0+K-1}, D'_{j_0+K}$, we select the midpoint of L_{j_0} as a break point P_r to break L_{j_0} into two pieces. (see Figure 8(a)). This is similar to the Interval Midpoint Strategy [4], but with the major difference that we use the middle point as a break point, rather than as a new control point in [4]. Suppose that L_{j_0} is defined over the parameter interval $[u_{1,j_0}, u_{2,j_0}]$ and the parametric value of P_r is u_{0,j_0} . Then L'_{j_0} and L'_{j_0+1} are defined over $[u'_{1,j_0}, u'_{0,j_0}]$ and $[u'_{0,j_0}, u'_{2,j_0}]$, respectively. Let \mathcal{Z}'_{j_0} denote the set of B-spline pieces that are partially affected by the control points $D'_{j_0}, D'_{j_0+1}, \dots, D'_{j_0+K-1}, D'_{j_0+K}$. (There are $2 \times (K - 1)$ such pieces for a closed B-spline curve.) Define the following objective function,

$$F_O = (F_{L'_{j_0}} + F_{L'_{j_0+1}}) + \sum_{L'_k \in \mathcal{Z}'_{j_0}} F_{L'_k}, \quad (10)$$

where $F_{L'_{j_0}}$ is the approximation error between L'_{j_0} and the target curve, which is given by $F_{L'_{j_0}} = \sum_{i=1}^m F_{d_i}^+(X'(u'_i))$, $u'_i \in [u'_{1,j_0}, u'_{0,j_0}]$, where $F_{d_i}^+(X'(u'_i))$ is the local squared distance between $X'(u'_i)$ and the target curve, and d_i is the distance between $X(u_i)$ and O_i . The m points $X'(u'_i)$, $i = 1, 2, \dots, m$, are evenly sampled in the parameter interval $[u'_{1,j_0}, u'_{0,j_0}]$ and the point $X(u_i)$ are determined from $X'(u'_i)$ through the linear mapping from $u'_i \in [u'_{1,j_0}, u'_{0,j_0}]$ to $u_i \in [u_{1,j_0}, u_{0,j_0}]$.

The term $F_{L'_{j_0+1}} = \sum_{i=1}^m F_{d_i}^+(X'(u'_i))$ is similarly defined with the linear mapping from $u'_i \in [u'_{0,j_0}, u'_{2,j_0}]$ to $u_i \in [u_{0,j_0}, u_{2,j_0}]$, and the error term $F_{L'_k} = \sum_{i=1}^m F_{d_i}^+(X'(u'_i))$ is defined in the same way as Eqn. (8) in Section 2.1 for control point removal.

Clearly, the objective function F_O (10) is quadratic in the unknown control points $D'_{j_0}, D'_{j_0+1}, \dots, D'_{j_0+K-1}, D'_{j_0+K}$. Therefore, these control points can be

computed efficiently by the quasi-Newton optimization. Figure 9 shows an example of inserting a new control point to an order-3 B-spline curve.

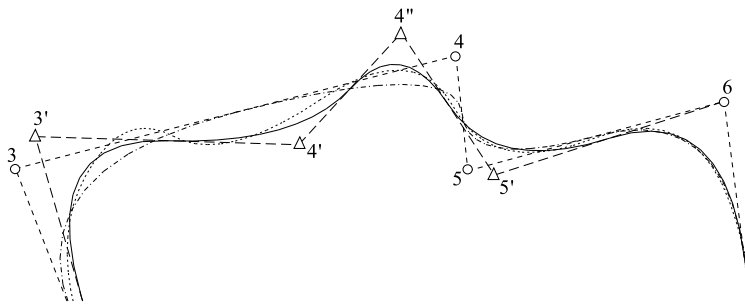


Figure 9. An example of control point insertion for an order-3 B-spline curve. The target curve is shown as the dotted curve. The B-spline curve is dash-dotted before control point insertion and solid after the insertion. The control points (3, 4, 5) are replaced by (3', 4', 4'', 5'). Clearly, a better approximation result is achieved after inserting the new control point.

Some remarks are in order about other existing methods for inserting new control points [3] [4] [12]. These methods are all for B-spline curve approximation based on data point parameterization.

- **Interval Midpoint Strategy.** Dierckx [4] proposes that a new control point be introduced in the middle of the piecewise-polynomial curve which has a large approximation error from the target curve.
- **Largest Displacement Strategy.** Lu and Milios [12] suggest that the new control point be inserted at the point on the spline which has the maximum displacement from the target curve.
- **PERM Strategy.** Cham and Cipolla [3] attempt to estimate the position on the spline to introduce a hinge such that the potential for reducing the approximation error is maximized. They call it the Potential for Energy-Reduction Maximization (PERM) method.

Cham and Cipolla compare the three methods above [3] and conclude that the first two methods are prone to be trapped in a weak local minimum while the PERM method yields near-optimal results in most cases. All these methods just add a new control point to the B-spline piece without redistributing the existing control points.

3 Complete algorithm

3.1 Algorithm

The algorithm takes as input a target curve, an initial B-spline curve and a user-specified error tolerance ϵ_0 . It outputs a B-spline curve approximating the target curve within the error tolerance ϵ_0 .

Begin

- (1) Compute a discrete distance field of the target curve on a grid using the Fast Marching Method. The distance to the target curve, together with other information, such as the curvature and normal vector of the equidistance curve, is computed for each grid point. (This part is elaborated in Section 4.)
- (2) Initialize the active B-spline curve. This entails the provision of the initial control points D_i , $i = 1, 2, \dots, n$, either manually or using the chord-length parameterization method [15]. (This is elaborated in Section 3.2.)
- (3) Use SDM of Pottmann et al to obtain an approximating curve starting with the current distribution of control point, and compute the approximation error E . If $E < \epsilon_0$, go to step 5.
- (4) Insert a new control point to a piece of the active B-spline curve where the local error is maximal and larger than ϵ_0 , following the steps in Section 2.2. Then run SDM to get the approximating active B-spline curve. Repeat this step until $E < \epsilon_0$.
- (5) Remove redundant control points while keeping $E < \epsilon_0$, following the steps described in Section 2.1.
- (6) Output the active B-spline curve.

End

3.2 Specifying initial B-spline curve

The initial shape of the active B-spline curve plays a critical role in our method. The basic requirement is that the shape of the initial B-spline curve should be close to the shape of the target curve. This requirement is formulated as a relationship between the initial B-spline curve and the medial axis of the target curve in [17]. Note that two initial curves of similar shapes may still have very different distributions of their control points (see the examples in Figure 2 and Figure 3).

A simple initial B-spline curve specified manually may suffice for a target curve of relatively simple shape. For example, the initial B-spline curve can be a line segment connecting the two ends of an open monotonic target curve as shown in Figure 14(a) or takes a circle-like shape controlled by a regular control polygon for a closed target curve as shown in Figure 4(a).

For a target curve of complicated shape or with self-intersection, such as those shown in Figure 10, a simple initial control polygon is no longer inadequate. In this case we generate the initial B-spline curve through the following steps. First we subdivide the target curve into several monotonic curve segments. (An open curve segment C is monotonic with respect to a direction v if any line perpendicular to v intersects C in at most one point. A curve segment C is said to be monotonic if it is monotonic with respect to some direction [14].) Then

local extreme points S_i on each monotonic curve segment are selected. Suppose that there are N_s such points. The middle point between two consecutive points S_i and S_{i+1} is added if the arc length between S_i and S_{i+1} is larger than L/N_s where L is the total length of the target curve. Hence, the target curve is divided into a number of curve pieces C_i by all these extreme points and additional middle points. Then the number of pieces, L_i , of the initial B-spline curve $X(u) = \sum_i B_i(u)D_i$ is set to be equal to the number of curve pieces C_i of the target curve.

Next, we sample m points $X_{k,i}$, $k = 1, 2, \dots, m$, on each piece L_i of the B-spline curve and obtain the corresponding points $P_{k,i}$ on C_i based on chord-length parametrization. Then the unknown initial control points D_i are computed by minimizing the function

$$F = \sum_i \sum_k \|X_{k,i} - P_{k,i}\|^2,$$

which is quadratic in the D_i . Note that data point parameterization by chord length is used here only for providing an initial B-spline curve. The subsequent steps of adjusting the control points and the use of SDM make our method independent of data parameterization. Examples of the initial B-spline curves generated with this method are presented in Section 5.

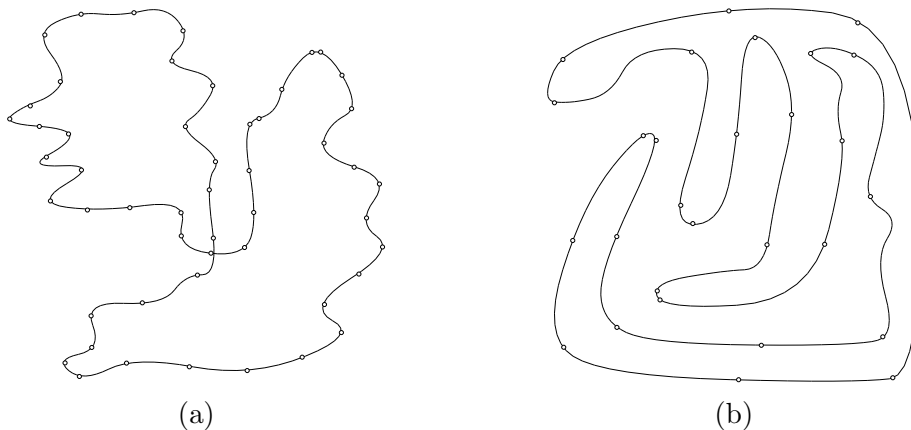


Figure 10. (a) The target curve is self-intersecting. (b) The target curve has complicated shape. The dots represent the selected extremum points and additional middle points. There are 52 and 29 such points in (a) and (b), respectively.

4 Pre-calculation of distance field

The local squared distance function $F_d^+(X_i)$ for each sample point X_i must be obtained in order to apply SDM. This in requires the computation of the foot-point O_i of X_i as well as the normal direction and curvature radius of the target curve at O_i . The foot-points of the sample points are also needed for evaluating the approximation error of an active B-spline curve. As a consequence, we need

to frequently compute the foot-points of a large number of sample points during optimization.

We compute the foot-points by pre-computing the distance field of the target curve using the Fast Marching Method. The Fast Marching Method, introduced by Sethian [22], has a wide range of applications; for instance, it has been used by Kimmel and Bruckstein [10] to compute the offset of a given shape. The Fast Marching Method is a numerical technique for solving the Eikonal equation

$$|\nabla T|F = 1, \quad \text{subject to } T|_{\Gamma} = T_0, \quad (11)$$

where F is assumed to be either always positive or always negative. The method uses an upwind, viscosity solution, and the finite difference scheme to numerically solve this equation. For $F = 1$ and $T_0 = 0$, the solution gives the signed distance from a target curve Γ . The distance field of Γ is computed on a discrete grid with the Fast Marching Method, whose time complexity is $O(M^2 \log M)$ for an $M \times M$ discrete grid in 2D plane. In our experiments it takes about 400 milliseconds to compute the distance field of a 2-D target curve with $M = 500$.



Figure 11. (a) The correspondence between X_i and O_i . (b) The distance field computed by the Fast Marching Method. The thick curve is the target curve, and the other thin curves are the equi-distance curves.

Once the distance field is available on a grid, the curvature k and the normal direction \vec{n} of the equi-distance curve can be computed for each grid point by

$$k = \nabla \cdot \frac{\nabla T}{|\nabla T|} = \frac{T_{xx}T_y^2 - 2T_xT_yT_{xy} + T_{yy}T_x^2}{(T_x^2 + T_y^2)^{3/2}}, \quad \text{and}$$

$$\vec{n} = \frac{\nabla T}{|\nabla T|}. \quad (12)$$

Then, during optimization, the distance d_i from an arbitrary sample point X_i to the target curve and the normal vector \vec{n}_i at X_i can be computed by

bi-linear interpolation from the neighboring grid points. The foot-point O_i of X_i is then computed by

$$O_i = X_i - d_i \cdot \vec{n}_i. \quad (13)$$

Note that the curvature needs to be computed only for grid points near the target curve and the curvature k of a foot-point on the target curve can be obtained by interpolation from the curvature values at the neighboring grid points.

Since the finite difference scheme used in the Fast Marching Method is a first-order approximation [20] [21], the error of the computed distance field is of the order of the grid spacing (i.e. mesh size). Because the accuracy of the distance field is critical only for sample points near the target curve, we compute a distance field with a multi-resolution approach, i.e. a refined grid is used in a narrow banded regions near the target shape, while a coarse grid is used for the remaining area. Let $d_0 = 10 \times \epsilon_0$, where ϵ_0 is the user-specified error tolerance. Then only those cells within the distance of d_0 from the target curve are computed with the refined grid. For a target curve in a unit square with $\epsilon_0 = 1 \times 10^{-3}$, we use a 500×500 coarse grid and a $5,000 \times 5,000$ refined grid, where 250,000 coarse cells and typically about 500,000 refined cells need to be computed, with the total computation taking approximately 2000 milliseconds. Furthermore, we note that it is possible to further improve the computational efficiency by employing an improved level-set method (GMM) [9], which has $O(M^2)$ time complexity versus the $O(M^2 \log M)$ time complexity of the Fast Marching Method used in our implementation.

Special treatment is needed for computing the distance field of a self-intersecting target curve. As shown in Figure 12(b), a sample point X_i on the B-spline curve can easily be misled to a wrong branch of the target curve when X_i is too close to a self-intersection point of the target curve. To circumvent this problem, we use different layers for sample points corresponding different branches of the target curve so that all sample points are made to converge to their corresponding branches. Figure 12 illustrates this procedure. The same problem also arises with a non-self-intersecting target curve when one part of the target curve is very close to another part while these two parts are not adjacent along the target curve. Although some of these cases can be handled by specifying a good initial shape by the chord-length parameterization (see Figure 15(a) and (b)), further study is needed to devise an effective solution that works in the general case.

5 Experimental results

In this section, we present some test examples to demonstrate the effectiveness of our method, as well as an example to reveal its limitation. The maximum

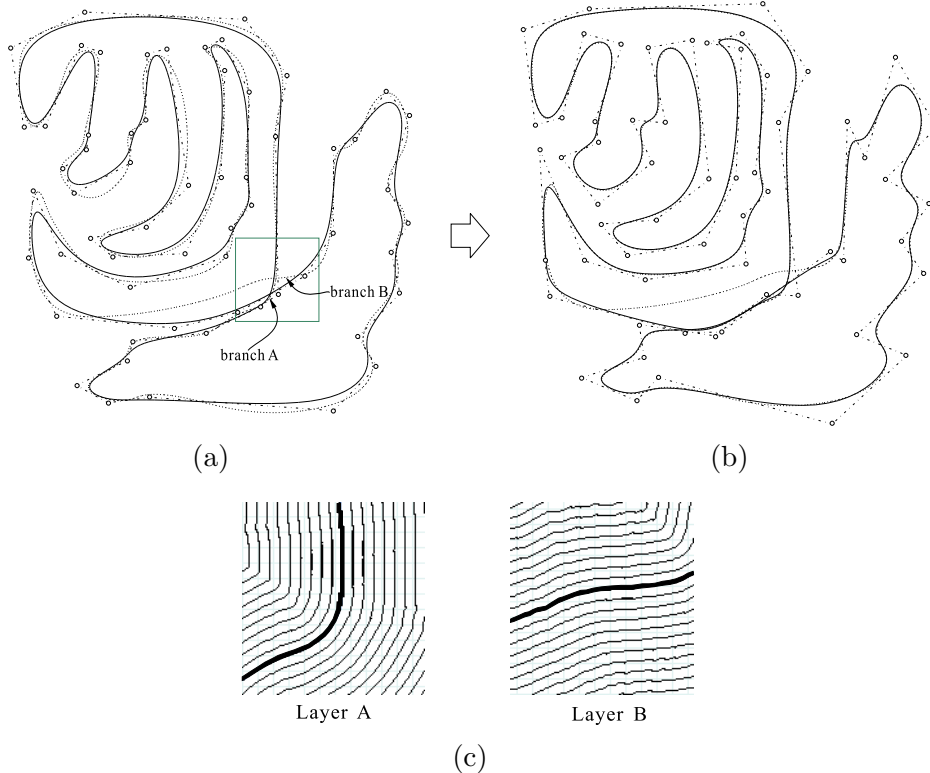


Figure 12. Two layers of the different distance field are used in a neighborhood of the self-intersection point of the target curve. The two branches A and B are indicated in (a) with their corresponding layers shown in (c). A poor approximation result obtained without using multiple layers is shown in (b). A better result obtained using multiple layers is shown in Figure 15(b) in Section 5.

error mentioned below is the maximum of the distances of the sample points on the active B-spline curve to the target curve. The average error is the approximation error defined by (5). All the experiments in this paper were run on a PC with a Pentium 4 2.4GHz CPU. The execution time shown does not include that of pre-computing the distance field. The computation of a multi-resolution distance field used in the following test examples takes about 2 seconds. The target curve lies in a unit square in all the examples.

Example 1: Control point adjustment. Figure 4(a) shows a simple enclosing curve as the initial active B-spline curve for the shown target curve of relatively simple shape. A direct application of SDM leads to the approximation shown in Figure 4(b). The active B-spline is guided by our method to produce a better approximation result, as shown progressively in Figures 4(c) and (d). The approximation errors and execution times for different cases are given in Table 1.

Example 2: Removal of redundant control points. When an initial B-spline curve has a very small approximation error and a large number of control points, our method can be used to reduce the number of the control points so as to achieve data simplification. Figure 13(a) shows a target curve which

Figure 4	(a)	(b)	(c)	(d)
No. of control points	25	25	25	25
Average error	1.680×10^{-1}	2.229×10^{-3}	1.880×10^{-3}	1.480×10^{-3}
Maximum error	3.067×10^{-1}	2.811×10^{-2}	1.132×10^{-2}	7.890×10^{-3}

Table 1. Errors of the B-spline approximations in Figure 4. The error tolerance is $\epsilon_0 = 1.5 \times 10^{-3}$. The total execution time is 79ms.

is a digital boundary curve extracted from a 2D CT image, and Figure 13(b) shows its B-spline approximation curve with 50 control points. Our method was applied to this initial active B-spline curve to reduce the number of control points, by assuming a more relaxed error tolerance. The number of control points are reduced gradually from 50 in (b) to 30 in (c), 15 in (d), 10 in (e), and, finally, 5 in (f). The errors of these coarser approximations are shown in Table 2. We see that the basic shape of the target curve is preserved even by the B-spline curve with only 5 control points. The average approximation error in Figure 13(f) is 1.063×10^{-2} and the maximum error is 5.753×10^{-2} .

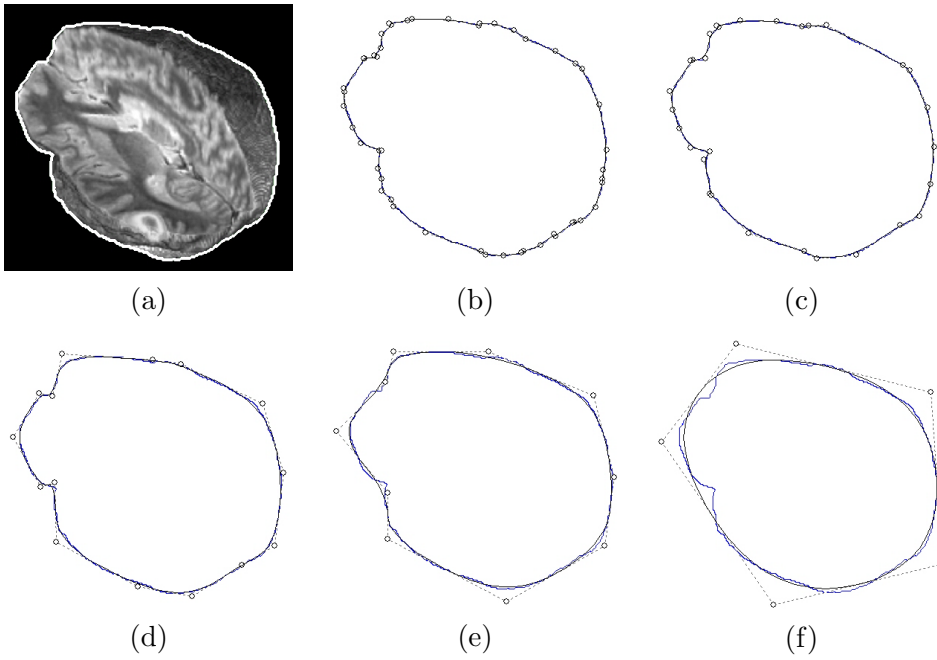


Figure 13. Removal of redundant control points from an order-3 B-spline curve. (a) The target shape is obtained from a CT image; (b) The initial B-spline curve with 50 control points; (c) The active B-spline curve with 30 control points; (d) 15 control points; (e) 10 control points; (f) 5 control points.

Example 3: Insertion of control points. Figure 14(a) shows an open target curve and an initial B-spline curve with 6 control points, comprising two triple points at the two ends of the target curve. Using our method, new control points are added one by one progressively to get the approximate results in

Figure 13	(b)	(c)	(d)	(e)
No. of control points	50	30	15	10
Average error	1.097×10^{-3}	1.352×10^{-3}	2.695×10^{-3}	4.820×10^{-3}
Maximum error	4.197×10^{-3}	5.092×10^{-3}	8.089×10^{-3}	2.104×10^{-2}

Table 2. Errors of the B-spline approximations in Figure 13. The error tolerance is $\epsilon_0 = 5.0 \times 10^{-3}$. The total execution time is 672ms.

(b), (c), and (d). The errors of the approximating curves in Figure 14(b), (c) and (d) are shown in Table 3.

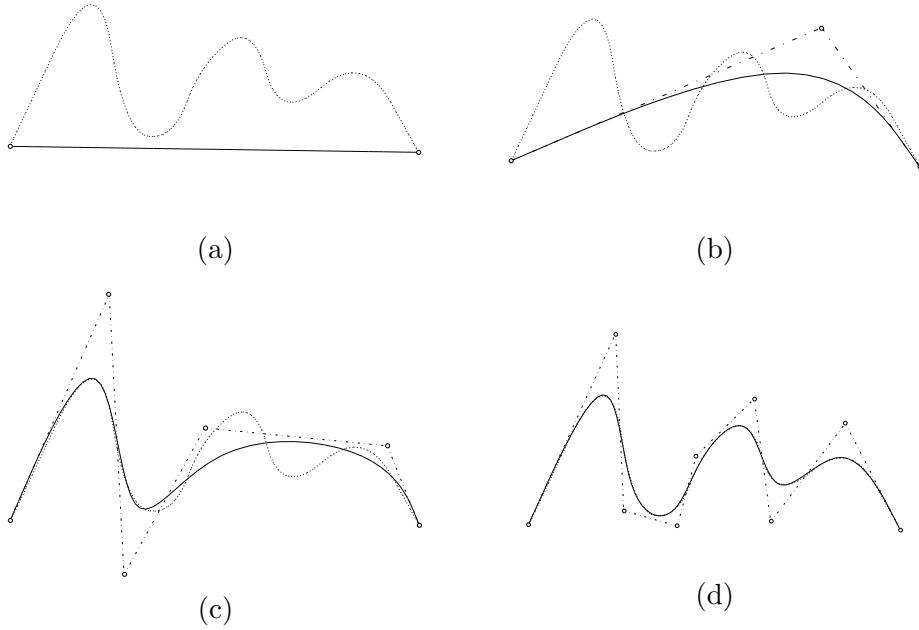


Figure 14. Inserting new control points. An open target curve and an initial order-4 B-spline curve with 6 control points is shown in (a); a triple control point is used at each end of the target curve. The active B-spline curve is shown with 7 control points in (b), 10 control points in (c), and 13 control points in (d).

Figure 14	(a)	(b)	(c)	(d)
No. of control points	6	7	10	13
Average error	9.038×10^{-2}	3.258×10^{-2}	1.394×10^{-2}	1.474×10^{-3}
Maximum error	2.507×10^{-1}	1.817×10^{-1}	1.090×10^{-1}	4.933×10^{-3}

Table 3. Errors of the B-spline approximations in Figure 14. The error tolerance is $\epsilon_0 = 1.5 \times 10^{-3}$. The total execution time is 94ms.

Example 4: Approximating a target curve of complicated shape.

Figure 15(a) shows a target curve that is both self-intersecting and highly concave, and Figure 15(b) shows a target curve which is the boundary curve extracted from the image of a palm. We first perform the monotonic curve subdivision and chord-length parameterization as described in Section 3.2 to generate their initial active B-spline curves as shown in Figure 15(a) and (c), which have 39 and 29 control points, respectively. Then our method produces the final approximating B-spline curves shown in Figure 15(b) and (d) with 60 and 57 control points, respectively. The approximations errors are shown in Table 4.

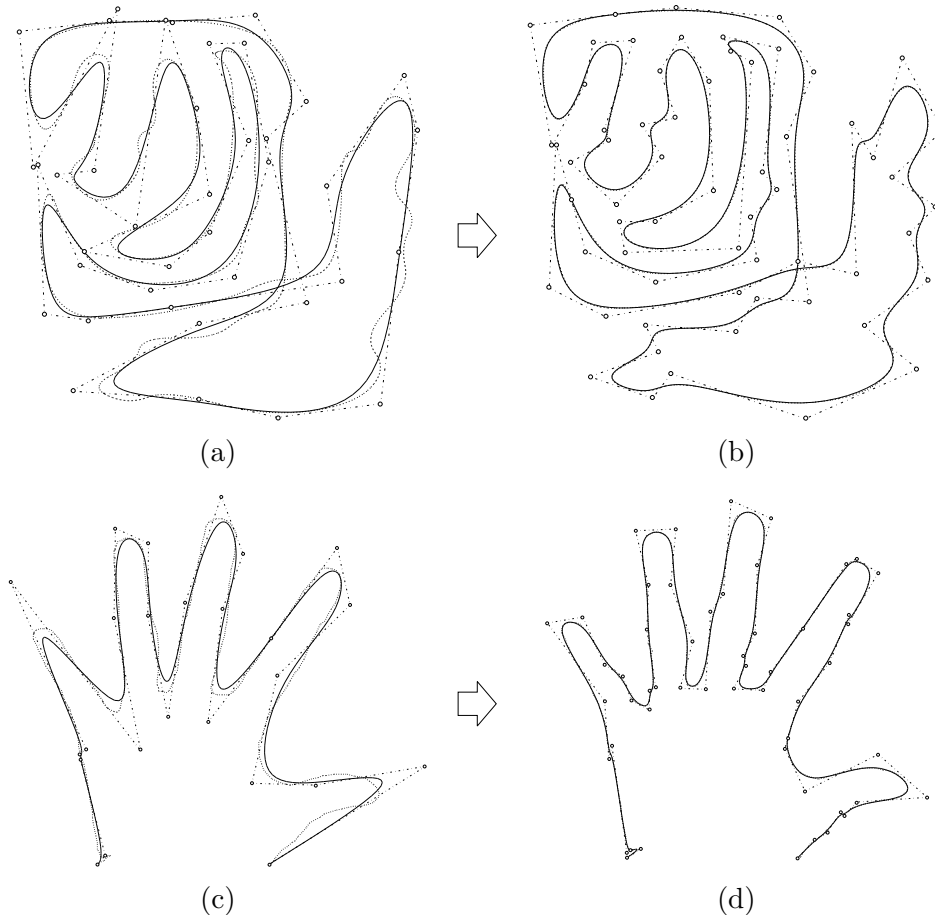


Figure 15. Approximating target curves of complex shape. An initial order-4 B-spline curve with 39 control points is shown in (a), and its corresponding final approximating B-spline curve with 60 control points is shown in (b). An initial order-4 B-spline curve with 29 control points for the “palm” curve is shown in (c), and its corresponding final approximating B-spline curve with 57 control points is shown in (d).

Example 5: A failure case. Our method will fail when two branches of the target curve are very close to each other so that a wrong matching occurs for the active B-spline curve. Figure 16 shows such an example. As mentioned in Section 4, this problem could be solved in some cases with the extra effort of computing multiple layers of the distance field for different branches that are

Figure 15	(a)	(b)	(c)	(d)
No. of control points	39	60	29	57
Average error	1.015×10^{-2}	9.931×10^{-4}	8.967×10^{-3}	9.937×10^{-4}
Maximum error	5.176×10^{-2}	4.214×10^{-3}	4.579×10^{-2}	4.911×10^{-3}

Table 4. Errors of the B-spline curves in Figure 15. The error tolerance is $\epsilon_0 = 1.0 \times 10^{-3}$. It took about 12 milliseconds to generate the initial B-spline curves in (a) and (c), respectively. It took about 893 and 983 milliseconds to produce final B-spline curves in (b) and (d), respectively.

close to each other (see Figure 12).

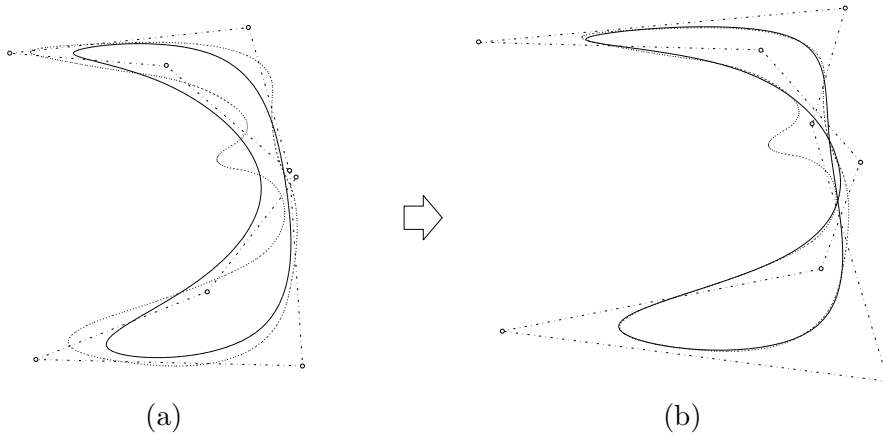


Figure 16. An example for which our method fails. (a) shows the initial order-4 B-spline curve. (b) shows the approximation result.

6 Conclusions

We have proposed new techniques for adjusting the control points of an active B-spline curve that is driven by the Pottmann et al’s optimization scheme (SDM) to converge to a given target curve in 2D plane. The resulting method produces better approximation results than SDM used alone. We have also shown that the pre-computation of a multi-resolution distance field of the target curve using the Fast Marching Method helps to improve both the efficiency and accuracy of our method.

There are two basic requirements on the initial B-spline curve for ensuring proper convergence in an active B-spline curve approximation method: 1) its shape should approximately capture the shape of the target curve; 2) its control points should be distributed appropriately. The techniques presented in this paper have proven helpful in relaxing the second requirement, i.e. improperly distributed control points can be adjusted or corrected to produce

a satisfactory result. However, the first requirement is still necessary for applying our method. We envision that a fully automatic method without both of the above requirements can be devised by integrating our techniques for control point adjustment with the framework of the active contour method (i.e. snake) [1] [8] or the level-set method [22].

An appropriate metric for approximation error is crucial to both defining the objective function and evaluating the approximation quality. The approximation error is measured in our method as the distance from the active B-spline curve to the target curve. However, a theoretically more appropriate measurement should be the Hausdorff distance between the active B-spline curve and the target curve [7]. Therefore a further research problem is to study whether the Hausdorff distance, or some form of its approximation, can be used as an error metric to obtain a more general B-spline curve approximation scheme.

Finally, further study is needed to gain a better understanding of the influence of the knot sequence of the active B-spline curve on the approximation quality.

References

- [1] A. Blake and M. Isard, *Active Contours*, Springer-Verlag, 1998.
- [2] G. Celniker and D. Gossard, Deformable curve and surface finite elements for free-form shape design, *Computer Graphics*, 25(4): 257-266, 1991.
- [3] T.J. Cham and R. Cipolla, Automated B-spline curve representation incorporating MDL and error-minimizing control point insertion strategies, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(1): 49-53, 1999.
- [4] P. Dierckx, *Curve and Surface Fitting with Splines*, Clarendon Press, 1993.
- [5] M. Eck and H. Hoppe, Automatic reconstruction of B-spline surfaces of arbitrary topological type, in *Proceedings of SIGGRAPH'96*, 325-334, 1996.
- [6] J. Hoschek and B. Jüttler, Techniques for fair and shape preserving surface fitting with tensor-product B-splines, in: J.M. Pena (Hrsg.), *Shape Preserving Representations in Computer Aided Design*, Nova Science Publishers, New York 1999, 163-185.
- [7] B. Jüttler, Bounding the Hausdorff distance of implicitly defined and/or parametric curves, *Mathematical Methods in CAGD: Oslo 2000*, 223-232, Vanderbilt University Press, Nashville 2001.
- [8] M. Kass, A. Witkins and D. Terzopoulos, Snakes – active contour models, *International Journal of Computer Vision*, 1(4): 321-330, 1987.
- [9] S. Kim, An $O(N)$ level set method for Eikonal equations, *SIAM J. Sci. Comput.* 22: 2178-2193, 2001.

- [10] R. Kimmel and A.M. Bruckstein, Shape offsets via Level Sets, *Computer Aided Design*, 25(3): 154-161, 1993.
- [11] V. Krishnamurthy and M. Levoy, Fitting smooth surfaces to dense polygon meshes, in *Proceedings of SIGGRAPH'96*, 313-324, 1996.
- [12] F. Lu and E. Milios, Optimal spline fitting to planar shape, *Signal Processing*, 37: 129-140, 1994.
- [13] W. Ma and J.P. Kruth, Parametrization of randomly measured points for the least squares fitting of B-spline curves and surfaces, *Computer Aided Design*, 27: 663-675, 1995.
- [14] K. Pankaj, Near linear time approximation algorithms for curve simplification in two and three dimensions, *Proc. of the 10th European Symposium on Algorithms (ESA '02)*: 29-41, 2002.
- [15] H. Park, Choosing nodes and knots in closed B-spline curve interpolation to point data, *Computer Aided Design*, 33(13): 967-974, 2001.
- [16] L. Piegl and W. Tiller, *The NURBS Book*, Springer-Verlag, New York, 1995.
- [17] H. Pottmann, S. Leopoldseder and S. Hofer, Approximation with active B-spline curves and surfaces, *Proc. Pacific Graphics 2002*, IEEE Press, 8-25, 2002.
- [18] H. Pottmann and M. Hofer, Geometry of the squared distance function to curves and surfaces, Technical Report 90, Institute of Geometry, Vienna Univ. of Technology, 2002.
- [19] D.F. Rogers and N.G. Fog, Constrained B-spline curve and surface fitting, *Computer Aided Geometric Design*, 21: 641-648, 1989.
- [20] J.A. Sethian, A fast marching level set method for monotonically advancing fronts, *Proc. Nat. Acad. Sci.*, 93, 4: 1591-1595, 1996.
- [21] J.A. Sethian, *Level Set Methods: Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision, and Materials Science*, Cambridge University Press, 1996.
- [22] J.A. Sethian, *Level Set Methods and Fast Marching Methods*, Cambridge University Press, 1999.
- [23] T.I. Vassilev, Fair interpolation and approximation of B-splines by energy minimization and points insertion, *Computer-Aided Design*, 28(9): 753-760, 1996.
- [24] X. Wang, F. Cheng and B.A. Barsky, Energy and B-spline interproximation, *Computer Aided Design*, 29(7): 485-496, 1997.