Optimal Spline Fitting to Planar Shape *

Feng Lu
Department of Computer Science
University of Toronto
Toronto, Ontario M5S 1A4

Evangelos E. Milios
Department of Computer Science
York University
North York, Ontario M3J 1P3

June 22, 1993

Abstract

Parametric spline models are used extensively in representing and coding planar curves. For many applications, it is desirable to be able to derive the spline representation from a set of sample points of the planar shape. The problem we address in this paper is to find a cubic spline model to optimally approximate a given planar shape. We solve this problem by treating the control points which define the spline as variables and apply an optimization technique to minimize an error norm so as to find the best locations of the control points. The error norm, which is defined as the total squared distance of the curve sample points from the spline model, reflects the discrepancy between the spline and the original curve. The objective function for the optimization process is the error norm plus a term which ensures convergence to the correct solution. The initial locations of the control points are selected heuristically. We also describe an extension of this method, which allows the addition of control points to the spline model based on local error measures if the initial set of control points fails to represent the given shape within a prespecified tolerance.

*Funding for this work was provided by an operating grant from the Natural Sciences and Engineering Research Council of Canada.

1 Introduction

Planar shape modeling has broad applications to various signal understanding and computer vision tasks, such as shape matching, feature extraction, data compression, or noise filtering [1][2][3][4]. Planar shapes are defined by their outline curves. It is desirable to be able to derive a parametric curve representation from a set of sample points which define a planar contour. This can be done by either Fourier descriptors or spatial domain models. One of the most commonly used models is the piecewise spline approximation, which has been extensively studied by the computer graphics community [5][6][7]. Local spline approximation is attractive in that it gives a smooth analytic representation of the curve. Much work has been done on representing signals or images using cubic spline models [8][9][10][11][12].

The problem we want to solve in this paper is the following. Given a planar curve represented as a sequence of sample points, find a spline with a number of control points to approximate the curve such that the approximation error is minimum. Our approach is to treat the positions of the control points of the spline as variables and use an iterative optimization algorithm to find the optimal solution in the sense that an error norm is minimum. We start with an initial set of control points, and we iteratively move them so as to improve the quality of the approximation. The approximation error norm is defined as the sum of the squared distances of the curve sample points from the spline curve. (The distance from a point to a curve is defined as the minimum distance from the point to any point in the curve.)

It appears that most of the existing work on spline approximation only deals with spline knots placement in the parametric space, while the spline control points are determined by using some least-square criteria [8]. For example, McCaughey and Andrews used variable knot splines to approximate images [9]. Paglieroni and Jain studied B-spline approximation of planar curves using a control point transformation [10], in which, first the spline knots are determined using some heuristic method, and then the control points are computed by using a transformation based on a least-square error criterion.

Our approach differs from existing work in that we treat the locations of the spline control points, rather than the knots in parametric space, as variables and we solve for the control points by minimizing an error norm which is different from the standard least-square criterion. We will compare our work with that of Paglieroni and Jain to illustrate the difference.

Paglieroni and Jain's control point transformation [10] solves the same problem as ours, namely how to use a spline model to approximate a planar curve. The control point transformation is based on a least-squared error between the given curve points and a set of selected points on the spline, obtained by uniformly (or heuristically) sampling the spline parameter. In our approach, the error norm measures the distance from each sample point to the spline itself. In other words, we effectively project each sample point onto the spline, and we claim that the distance between the point and its projection leads to a more natural error norm than using a point on the spline different from its projection.

In our approach, we fix the knots of the spline as integers from 0 to n (where n is the number of spline segments or equivalently the number of control points). Nevertheless, we can easily extend our method to work on an arbitrary set of knots. Our aim is to improve the selection of control points once the knots are determined.

Plass and Stone [11] used an iterative method which is similar to ours in that, at each step, it finds the parameter values corresponding to the projections of the sample points, and then it fits a least-square solution to the curve. Through the iterations, the method minimizes the least-square distance from the curve points to their projections on the spline. However, this method is limited to fitting a single spline segment at a time, which means the original curve must be segmented before the fitting process can be applied. Also the tangent directions at the spline end-points must be predetermined in order to obtain a smooth representation across the spline segments. Because each segment of the curve is approximated separately, a global error measure can not be easily defined or minimized. The convergence properties of their fixed-point iteration are not clear. Moreover, iterations with the Newton's formula are used to update the parameter values at each step, of which a possible divergence situation is not considered.

In our approach, the problem is well posed as an optimization problem with the objective function being the global error norm. It is guaranteed that a solution will be found. By considering the approximation error globally, we can move the control points to any location and achieve an optimal solution in terms of the global error norm. Although the optimization process sometimes only gives a local minimum, we can heuristically choose a suitable initial estimate, for which the method will converge to satisfactory results. Later we will point out that the computational cost for each step of our method is less than that of a step of Plass and Stone's method if it is extended to fit spline globally.

If we only apply the optimization process once, we get an optimal solution for a fixed number of control points. We can also add new control points to the model according to some criterion based on the error norm so as to find a best solution in the sense of a minimum number of control points with the error norm not exceeding a given threshold. Our strategy is as follows. When the error norm is not small enough, we insert a new control point in the area corresponding to the maximum term in the error norm. With the increased degree of freedom, the local error is likely to be reduced. We can repeat this process until the total error is below the given limit.

The paper is structured as follows. In section 2, we review the B-spline model. In section 3, we present the definition and computation of the error norm. In section 4, we introduce the objective function used in the optimization process. In section 5, we discuss the optimization process and explain the selection of initial estimate. We also discuss the complexity and the convergence properties of our algorithm. In section 6, we give a few examples of planar shape approximation using our method.

2 B-spline Approximation

We use the standard cubic B-spline model to approximate a given curve. B-splines are approximating splines. They do not pass through their control points, contrary to local interpolating splines. Cubic B-splines have the property that they are continuous in first and second derivatives, while interpolating cubic splines are discontinuous in the second derivative [5][6].

The basis function of the normalized rth order B-spline associated with knots $t_i, \ldots, t_{i+r}$ is defined recursively by

$$N_{i,1}(u) = \begin{cases} 1, & t_i \leq u < t_{i+1} \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

$$N_{i,r}(u) = \frac{(u - t_i)N_{i,r-1}(u)}{t_{i+r-1} - t_i} + \frac{(t_{i+r} - u)N_{i+1,r-1}(u)}{t_{i+r} - t_{i+1}}, \qquad r > 1 \tag{2}$$

The nth order normalized B-spline curve model equation is

$$\mathbf{P}(u) = \sum_{i=0}^{n-1} \mathbf{C}_i \; \phi_{i,r}(u) \tag{3}$$

where $\mathbf{P}(u) = (X(u), Y(u))^T$ is the curve model of a continuous parameter u; $\mathbf{C}_i = (x_i, y_i)^T$, $i = 0, \ldots, n-1$, are the control points; and $\phi_{i,r}(u)$ is the periodic extension of $N_{i,r}(u)$ with period $t_n - t_0$ in the case of closed curves [7].

The cubic B-spline model (periodic) with integer knots $0, 1, \ldots, n-1$ and control points $\mathbf{C}_0, \mathbf{C}_1, \ldots, \mathbf{C}_{n-1}$ is defined for $0 \leq u < n$, and it can be written in the form

$$\mathbf{P}(u) = [t^3, t^2, t, 1]\mathbf{B}[\mathbf{C}_{i-1}, \mathbf{C}_i, \mathbf{C}_{i+1}, \mathbf{C}_{i+2}]^T \qquad \text{for} \quad i \leq u < i+1, \tag{4}$$

where i is an integer from 0 to $n-1$ (the index of $\mathbf{C}$ should be modulo n), $t = u - i$, $\mathbf{B}$ is the spline matrix:

$$\mathbf{B} = \frac{1}{6}\begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{pmatrix} \tag{5}$$

The above spline model $\mathbf{P}(u)$ is a closed piecewise parametric curve consisting of n pieces of local B-splines. We will use this cubic model $\mathbf{P}(u) = (X(u), Y(u))^T$ for the curve approximation.

3 Error Norm of Spline Approximation

The idea behind the formulation of the problem as an optimization problem is the following. Given a curve $\mathbf{S}$ and a set of control points defining a spline approximation $\mathbf{P}$ of the curve $\mathbf{S}$, we compute the distance of each point $\mathbf{S}_i$ of the curve from $\mathbf{P}$. We then sum up all these distances to form the error norm E between $\mathbf{S}$ and $\mathbf{P}$. This error norm is a measure of how closely $\mathbf{P}$ approximates $\mathbf{S}$. We observe, however, that for a fixed curve $\mathbf{S}$, E is a function of the control points defining the spline approximation $\mathbf{P}$. The problem of finding the optimal approximation of curve $\mathbf{S}$ is then naturally cast as the problem of determining the values of the variables, which are the coordinates of the control points of $\mathbf{P}$, so as to minimize E. This problem is addressed in the next section.

We formally define the error norm E as the following.

Definition 1 *Let $\mathbf{P}(u)$ be the spline model defined by n control points as given in section 2, with parameter u. For any point $\mathbf{S}_i$, the distance from $\mathbf{S}_i$ to $\mathbf{P}$ is defined as the distance from*

$\mathbf{S}_i$ *to* $\mathbf{P}_i$, *where* $\mathbf{P}_i = \mathbf{P}(u_i)$ *is the point on the spline model that is closest to* $\mathbf{S}_i$, *i.e.,*

$$|\mathbf{S}_i - \mathbf{P}_i| = \min_{0 \le u < n} (|\mathbf{S}_i - \mathbf{P}(u)|). \tag{6}$$

Definition 2 *Given a sequence of* m *sample points,* $\mathbf{S}_i = (sx_i, sy_i)^T$, $i = 1, \ldots, m$, *representing a closed planar curve, and a spline model* $\mathbf{P}(u)$, *the error norm of approximation is defined by:*

$$E = \sum_{i=1}^{m} w_i |\mathbf{S}_i - \mathbf{P}_i|^2 \tag{7}$$

where w_i*'s are non-negative weights with* $\sum_{i=1}^{m} w_i = 1$.

The weights in the error norm can be used to emphasize some of the sample points, such as those with high curvature. We can select the weights according to the estimated curvature values.

To find the minimum distance from $\mathbf{S}_i$ to the spline, we need the global minimum of the distance function $F_i(u) = (X(u) - sx_i)^2 + (Y(u) - sy_i)^2$ corresponding to each sample point. We use two methods to find this global minimum. The primary method uses the minimum from the previous iteration and updates it to become the new minimum. We notice that, during the iteration, we typically only change the positions of the control points by a small amount. If we know the point $\mathbf{P}(u_i)$ on the spline that is closest to curve point $\mathbf{S}_i$ before the change, we can update the point $\mathbf{P}(u_i)$ by updating u_i to u_i'.

Let $f(u) = \frac{dF_i}{du}$, then $f(u_i) = 0$. Notice that f is the function of the coordinates of 4 control points. For a coordinate, say x_k, of one of the control points,

$$\frac{\partial f}{\partial x_k} + \frac{\partial f}{\partial u_i} \frac{\partial u_i}{\partial x_k} = 0 \tag{8}$$

then

$$\frac{\partial u_i}{\partial x_k} = -\frac{\frac{\partial f}{\partial x_k}}{\frac{\partial f}{\partial u_i}} = -\frac{\frac{\partial f}{\partial x_k}}{f'(u_i)} \tag{9}$$

where

$$\frac{\partial f}{\partial x_k} = 2B_l(u)\frac{\partial X(u)}{\partial u} + 2(X(u) - sx_i)B_l'(u) \tag{10}$$

where $B_l(u)$ is the basis function (a cubic polynomial) associated with control point $\mathbf{C}_k$.

Now if each (x_k, y_k) has changed to $(x_k + \Delta x_k, y_k + \Delta y_k)$, we can approximate u_i' by:

$$u_i' \approx u_i + \Delta u_i \approx u_i + \sum_{j=1}^{4}(\frac{\partial u_i}{\partial x_{k_j}}\Delta x_{k_j} + \frac{\partial u_i}{\partial y_{k_j}}\Delta y_{k_j}). \tag{11}$$

We can apply the Newton's method to improve $u_i + \Delta u_i$. Since $u_i + \Delta u_i$ is very close to u_i', it is a good initial value to the iteration. In case that the update gets u_i out of the current spline piece, we need to use the secondary method to find a new starting point.

The secondary method, which is used initially and when the update method fails, works by brute force. It finds all the extrema of $F_i(u)$ and chooses the minimum among them. We notice that $F_i(u)$ consists of n pieces of polynomials of order 6 as $(X(u), Y(u))$ are piecewise cubic polynomials. By letting $\frac{dF_i}{du} = 0$, we end up with n polynomial equations of order 5, although only one of them contains the minimum we want. Here it is possible to heuristicly choose only those equations which are likely to contain the global minimum. We apply Laguerre's method [13] to solve these equations. Laguerre's method finds all the roots of a function one at a time iteratively. It has the advantage that it is near-globally convergent. We efficiently implemented the method in such a way that it avoids unnecessary complex arithmetic operations.

By solving the equations, we find the parameter $u = u_i$ and the corresponding point $\mathbf{P}(u_i)$ on the spline. Then $|\mathbf{S}_i - \mathbf{P}(u_i)|$ is the required minimum distance.

4 Objective Function for Minimization

Our objective function consists of two terms. The first term is the error norm defined in the previous section, with the coordinates of all the control points as variables.

The second term $T(u_1, \ldots, u_m)$ is intended to constrain the spline to stay close to the curve $\mathbf{S}$. The reason for introducing this term is that by only minimizing the error norm it is possible that the resulting spline is self-intersecting and part of the resulting approximating spline has a large deviation from the curve being approximated. Such a solution does not violate the minimum error norm criterion since our error norm is based on the distance from a point on the curve to the spline. An appropriate error term should get rid of such self-intersection and at the same time, ensure that the objective function has continuous derivatives. We introduce a term of the form:

$$T(u_1, \ldots, u_m) = \sum_{i=1}^{m} \alpha\left(\frac{u_i - u_{i-1}}{\beta}\right)^r. \tag{12}$$

The idea behind this term is that any two successive points $\mathbf{S}_i$ and $\mathbf{S}_{i+1}$ of the curve $\mathbf{S}$ are close to each other, therefore their projections $\mathbf{P}(u_i)$ and $\mathbf{P}(u_{i+1})$ should also be close to each other, not only in space, but also in terms of the values u_i and u_{i+1} of the spline parameter u. The above error term imposes a penalty if the values u_i and u_{i+1} are not close to each other, which is the case when the approximating spline self-intersects or it has an offshoot that deviates largely from the curve $\mathbf{S}$. The parameters α, β, and r are related to the scale of the curve. However, the choice of the parameters is not very important since we only need, qualitatively in some sense, a large value when $u_i - u_{i-1} > \beta$ (indicating that a piece of the spline is off the curve) and a very small value when $u_i - u_{i-1} < \beta$ (which is normal since $n << m$). In the experiment in section 6, we choose them as $\alpha = 1$, $\beta = 0.5$, $r = 10$. Here $i - 1$ and $u_i - u_{i-1}$ should be considered as modulo n.

Formally, the objective function is:

$$\begin{aligned} F(x_0,\ldots,x_{n-1},y_0,\ldots,y_{n-1}) &= \sum_{i=1}^{m} w_i((X(u_i)-sx_i)^2+(Y(u_i)-sy_i)^2) \\ &\quad +T(u_1,\ldots,u_m) \end{aligned} \tag{13}$$

where $(X(u_i),Y(u_i))$, which is the function of $x_0,\ldots,x_{n-1},y_0,\ldots,y_{n-1}$, is the projection of the curve point $\mathbf{S}_i=(sx_i,sy_i)$ onto the approximating spline $\mathbf{P}$. The function F is continuous and so are its first derivatives.

In order to minimize F efficiently, we need to find the first derivatives of F with respect to all its variables. Without loss of generality, let us only consider $\frac{\partial F}{\partial x_k}$.

Consider one term of the sum in F (without the weight),

$$F_i(u_i)=(X(u_i)-sx_i)^2+(Y(u_i)-sy_i)^2 \tag{14}$$

and notice that u_i is a function of $x_0,\ldots,x_{n-1},y_0,\ldots,y_{n-1}$,

$$\begin{aligned} \frac{\partial F_i}{\partial x_k} &= 2(X-sx_i)(\frac{\partial X}{\partial u_i}\frac{\partial u_i}{\partial x_k}+\frac{\partial X}{\partial x_k})+2(Y-sy_i)\frac{\partial Y}{\partial u_i}\frac{\partial u_i}{\partial x_k} &(15)\\ &= 2(X-sx_i)\frac{\partial X}{\partial x_k}+2\frac{\partial u_i}{\partial x_k}((X-sx_i)\frac{\partial X}{\partial u_i}+(Y-sy_i)\frac{\partial Y}{\partial u_i}) &(16) \end{aligned}$$

The second term is 0 at $u=u_i$ because u_i is a minimizer of $F_i(u)$ and

$$\left.\frac{\partial F_i}{\partial u}\right|_{u=u_i} = \left.\left(2(X-sx_i)\frac{\partial X}{\partial u}+2(Y-sy_i)\frac{\partial Y}{\partial u}\right)\right|_{u=u_i}=0 \tag{17}$$

then

$$\left.\frac{\partial F_i}{\partial x_k}\right|_{u=u_i} = 2(X-sx_i)\left.\frac{\partial X}{\partial x_k}\right|_{u=u_i} \tag{18}$$

It can be determined on which piece of the piecewise spline model the projection $\mathbf{P}(u_i)$ is located and $\frac{\partial X}{\partial x_k}$ is the spline basis function associated with control point $\mathbf{C}_k$.

For the term T,

$$\frac{\partial T}{\partial x_k} = \sum_{i=1}^{m} \frac{\alpha r}{\beta} (\frac{u_i - u_{i-1}}{\beta})^{r-1} (\frac{\partial u_i}{\partial x_k} - \frac{\partial u_{i-1}}{\partial x_k}) \tag{19}$$

where the computation of $\frac{\partial u_i}{\partial x_k}$ is described in the previous section.

In the implementation, $2n$ derivatives corresponding to $2n$ variables (the coordinates of the n control points) are accumulated through the loop of evaluating m components of the error norm, where each component corresponds to a curve sample point.

5 Optimization Process

The algorithm we apply to minimize the objective function uses a secant method with BFGS updates to the approximated Hessian matrix [14][15]. The algorithm combines line search to ensure that it finds a point with a lower value in the objective function at every step. The rate of convergence of the secant method is super-linear. We supply a subroutine to evaluate the objective function and the gradient, and we provide an initial estimate of the minimizer.

The initial estimates of control points are selected heuristically. We first select a set of points on the original curve to be the initial spline end-points (where the spline parameter equals to a knot) by using curvature heuristics. The heuristic algorithm is as follows. We first estimate the curvature at each sample point by taking the derivative of a smoothed version of the tangent. Then we select each sample point at which the absolute value of curvature is a local maximum which also exceeds some threshold. If the curvature at a point is extremely large, indicating a sharp corner, we take two points around the corner. Finally, we add equally spaced middle points between two selected sample points if the distance between these two points is large.

Note that our optimization algorithm is quite independent of the curvature heuristics. All we need is to select a sensible set of points from the original curve. We found that the initial end-points (which determine the initial control points) need not to be chosen too carefully. The optimization process has the ability to move the points around and converge to the optimal result.

After selecting the end-points of the initial spline model, we compute the corresponding initial set of control points by solving a linear system of equations. Assume $\mathbf{C}_{j-1}, \mathbf{C}_j, \mathbf{C}_{j+1}$ are 3 successive control points. Then from the property of B-splines, the point $\mathbf{Q}_j$ where

$$\mathbf{Q}_j = \frac{1}{6}(\mathbf{C}_{j-1} + 4\mathbf{C}_j + \mathbf{C}_{j+1}) \tag{20}$$

is an end-point of the spline. With n such equations and given $\mathbf{Q}_j$'s, we can solve for the control points $\mathbf{C}_j$'s. These are the initial estimate of control points.

We also use the estimated curvature to derive the weights in the error norm. We want to put more emphasis on the high curvature regions of the curve so as to obtain an approximation which is visually pleasing. (Without the weights, the approximation tends to underweight the corner points.) In the experiments, we have set the ratio of the maximum weight and the minimum weight to be about 5 to 1.

At some point of the optimization process, we can find the largest term in the error norm, which corresponds to the maximum local error. If this error is large, it probably means that there are not enough degrees of freedom in the spline model to approximate this region of the curve. We can insert one or more control points in the vicinity and rerun the optimization process. The local error should be reduced. If we repeat this process, we can determine the spline that approximates the given curve within a specified error tolerance.

We will analyze the complexity of the algorithm. For every iteration, the cost in the secant method is $O(n^2)$. The cost involved in evaluating the objective function and gradient (using the updating method) is $O(m + n)$, where m, n are the number of sample points and the number of control points, respectively. So the total cost for one iteration is $O(m+n^2)$. Notice that this is less than the cost of solving a linear system of $2n$ variables (as would an extension of Plass and Stone's method do). Also notice that the $O(m)$ part of the computation is parallelizable as the distances from the sample points to the spline can be computed independently. The number of iterations needed to converge to an optimal solution is typically 100–300 (depending on, among other factors, the convergence criterion). Although the iterations move the control points around quite significantly, we note that, after the first 50–100 iterations, the approximation error usually can not be reduced much further. If we only want a reasonably good approximation, we can safely stop at a fixed number of iterations.

The convergence property of our algorithm is worth noting. Our process has three types of exits. The normal exit, which is reached most of the times, is at convergence (in the sense of a zero gradient). Very seldomly, the process may exit abnormally when either a given number of iterations is exhausted, or it fails to find a lower point along the search direction. (We believe that round-off errors may contribute to the last case.) Note that even at abnormal exits, we still get very good results. Typically, the process has run a few hundred iterations when it reaches an abnormal exit and the gradient is already very small; the approximation will not have much to improve anyway. We mentioned earlier that the optimization routine always finds a lower point at every iteration. Divergence should never occur.

6 Experiments

In this section, we will show a few experimental results of spline approximation for closed planar curves. All the curves used in the examples are obtained by tracing the boundaries of scanned shapes.

The first example shows the approximation to the map of Africa (Figure 1). The original curve is shown in (a) which consists of 507 sample points. The width and height of the shape are 164 pixels by 175 pixels. Part (b) shows the initial estimate of the spline which is determined by 28 end-points. These end-points (marked with a dots in the figure) are selected from the sample points using curvature heuristics. The control points of the spline, which are derived from the end-points, are also shown in the figure with small circles off the curve. Pixels of the original curve are plotted as small dots. Part (c) is the result of the optimization process using the initial estimate given in (b). Part (d) gives the optimal spline in terms of a minimum of control points (which is 43) with the maximum approximation error less then one pixel. Only the end-points are shown for clarity. In (d), we ran 50 iterations for each added new control point and finally let it run until convergence (which took 130 iterations). Table 1 shows the error measures of the above approximation results. (Maximum error is the maximum of the distances from the curve sample points to the spline. Average error is the square root of the error norm. The unit of the errors is pixel width.)

Spline	No. of CPs	Maximum Error	Average Error
Figure 1(b)	28	5.5949	1.3555
Figure 1(c)	28	1.7692	0.6644
Figure 1(d)	43	0.9894	0.3604

Table 1: Errors of the spline approximations in Figure 1.

In the second example, we show the converging process of approximating the contour of a leaf (named trifoliate) (Figure 2). The original curve has 366 sample points. The initial estimate is given in (a). The derived spline at 30, 75, and 277 iterations (convergence) are shown in (b), (c), and (d), respectively. Table 2 gives the approximation errors. The unit for the errors is pixel width.

Next, we illustrate the process of adaptively adding new control points to the model when approximating the shape of a peltate leaf. Figure 3 shows the results of the optimal spline with 9, 10, 11, and 13 control points. Table 3 gives the errors of the approximations.

Spline	Iterations	Maximum Error	Average Error
Figure 2(a)	0	3.8160	0.8334
Figure 2(b)	30	1.2091	0.3977
Figure 2(c)	75	1.0020	0.3466
Figure 2(d)	277	0.9774	0.3107

Table 2: Errors of the spline approximations in Figure 2.

Spline	No. of CPs	Maximum Error	Average Error
Figure 3(a)	9	4.3594	1.7643
Figure 3(b)	10	3.2655	1.1371
Figure 3(c)	11	2.3054	0.8246
Figure 3(d)	13	1.1079	0.3405

Table 3: Errors of the spline approximations in Figure 3.

The shapes of our last example are taken from the experiments in Paglieroni and Jain's work [10]. Figure 4(a) shows the original curves (contours of three keys); (b) shows the spline approximations to the key contours. In each of the three cases, the spline model has a minimum number of control points while its maximum approximation error is less than one pixel. By comparing the results visually, we note an improvement compared with the control point transformation method presented in [10]. Table 4 lists the number of sample points of each curve, the number of control points used in each spline, the approximation errors, and the number of iterations for the optimization process to converge.

Curve	No. of Samps	No. of CPs	Maximum Error	Average Error	Iterations
Key 1	436	30	0.8717	0.2870	233
Key 2	337	22	0.7295	0.2685	97
Key 3	385	23	0.9073	0.2928	138

Table 4: Errors of the spline approximations in Figure 4.

7 Conclusion

In this paper we have described an algorithm for optimally approximating a planar curve with a piecewise spline model, by using numerical optimization techniques. Here "optimal" means minimum error norm of approximation for a given number of control points. We define the error norm to be the sum of squared distance from each sample point to the nearest point on the spline. The coordinates of the spline control points are treated as variables, which are computed by the optimization process. By a simple extension, we can also use our method to find the spline model with a minimum number of control points while the error norm is less than a given bound.

The algorithm compares favorably with previously reported methods in terms of the quality of the approximation. Being an iterative algorithm, our method has a relatively high computational cost. However, we note that a large part of the algorithm is parallelizable. In addition, we can flexibly trade between complexity and quality of approximation by limiting the number of iterations.

References

[1] T.Pavlidis & S.L. Horowitz, "Segmentation of Plane Curves", *IEEE Trans. Comput.* **23**,1974, 860–870.

[2] E.E. Milios, "Shape Matching Using Curvature Processing", *Comp. Vision Graphics Image Process.* **47**,1989, 203–226.

[3] G.Medioni & Y.Yasumoto, "Corner Detection and Curve Representation Using Cubic B-Spline", *Comp. Vision Graphics Image Process.* **39**,1987, 267–278.

[4] D.J. Langridge, "Curve Encoding and Detection of Discontinuities", *Comp. Graphics Image Process.* **20**,1982, 58–71.

[5] R.H. Bartels, J.C. Beatty & B.A. Barsky, *An Introduction to Splines for Use in Computer Graphics and Geometric Modeling*, Morgan Kaufmann Publishers, Calif., 1987.

[6] M.E. Mortenson, *Geometric Modeling*, John Wiley & Sons, New York, 1985.

[7] L.L. Schumaker, *Spline Functions: Basic Theory*, Wiley, New York, 1981.

[8] J.R. Rice, *The Approximation of Functions*, vol.2, Mass. Addison-Wesley, 1969, p123.

[9] D.G. McCaughey & H.C. Andrews, "Image Approximation by Variable Knot Bicubic Splines", *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. **3**, No. 3, 1981, 299–310.

[10] D.W. Paglieroni & A.K. Jain, "Control Point Transforms for Shape Representation and Measurement", *Comp. Vision Graphics Image Process.* **42**,1988, 87–111.

[11] M. Plass & M. Stone, "Curve-Fitting with Piecewise Parametric Cubics", *ACM Computer Graphics*, Vol. **17**, No. 3, 1983, 229–239.

[12] P. Dierckx, "Algorithms for Smoothing Data with Periodic and Parametric Splines", *Comp. Graphics Image Process.* **20**, 1982, 171–184.

[13] R.L. Johnston, *Numerical Methods—A Software Approach*, John Wiley & Sons, New York, 1982.

[14] J.E. Dennis & R.S. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice Hall, New Jersey, 1983.

[15] D. Kahaner, C. Moler & S. Nash, *Numerical Methods and Software*, Prentice Hall, New Jersey, 1989.

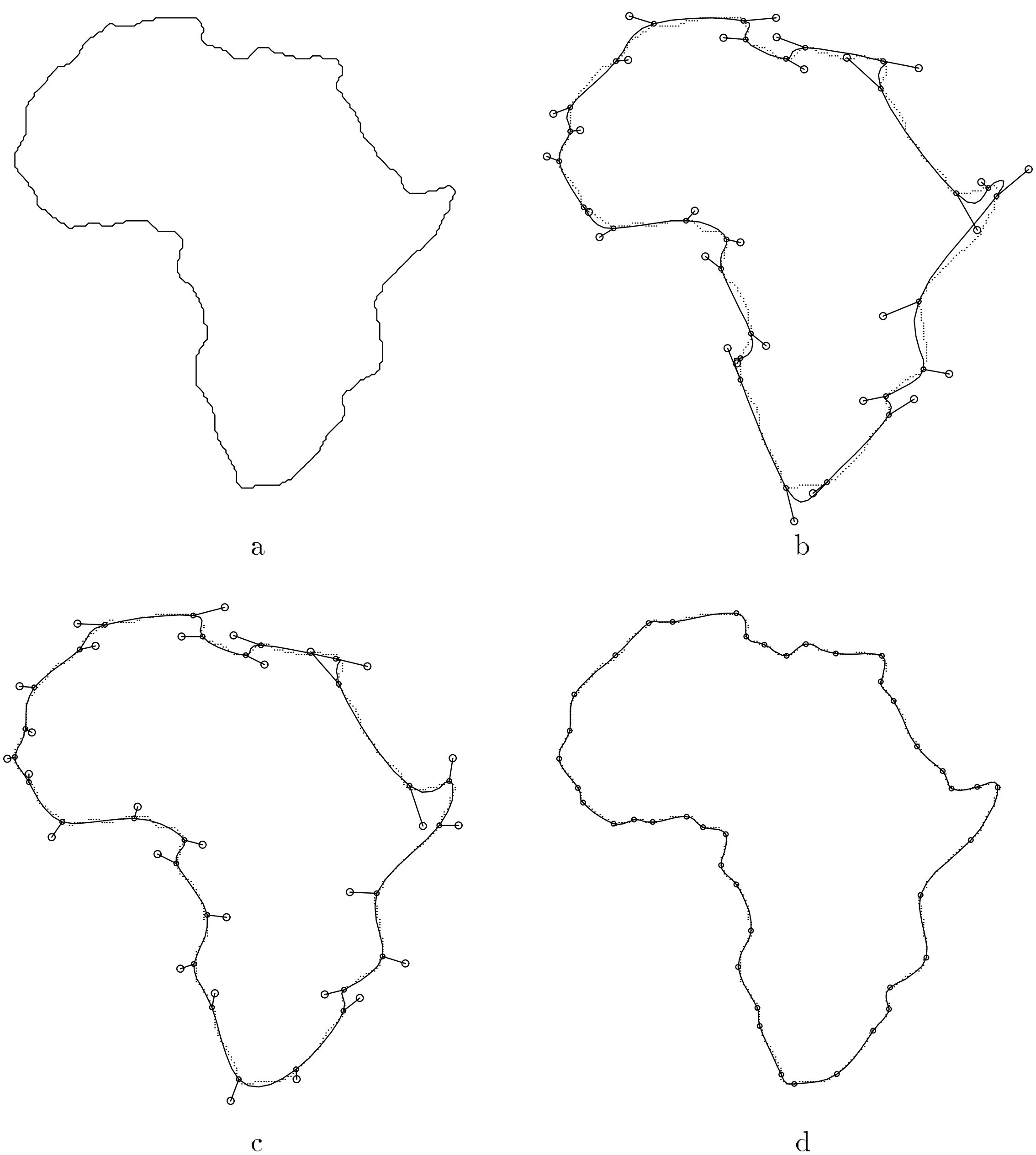

Figure 1: Spline approximations to the contour of the map of Africa: (a) original curve; (b) initial estimate of spline model with 28 control points; (c) converged optimal spline model using the initial estimate in (b); (d) optimal spline model (with 43 control points) by which the approximation error is less then one pixel.

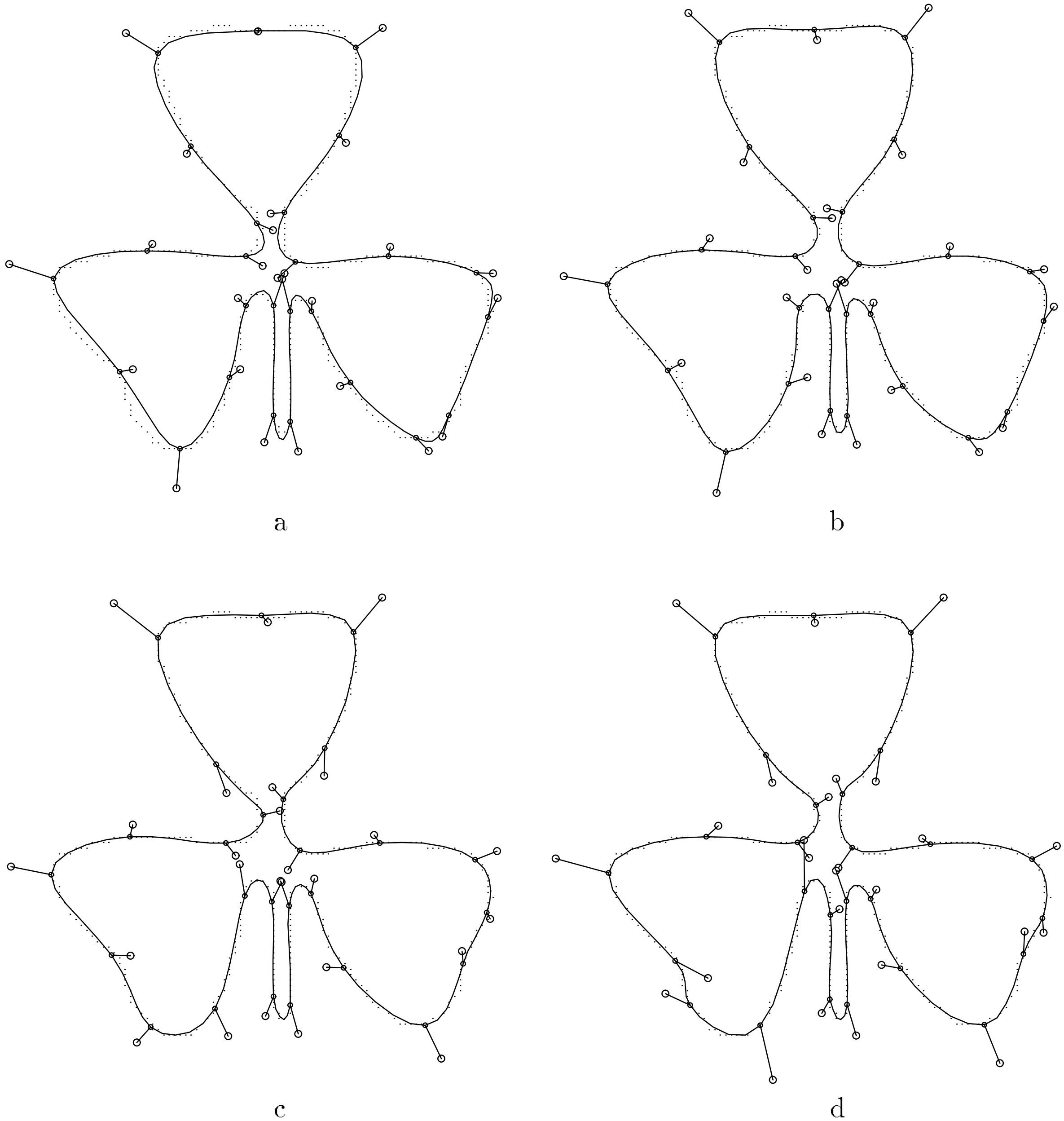

Figure 2: Converging process of the spline approximation of a leaf shape: (a) initial estimate; (b) after 30 iterations; (c) after 75 iterations; (d) at convergence (after 277 iterations).

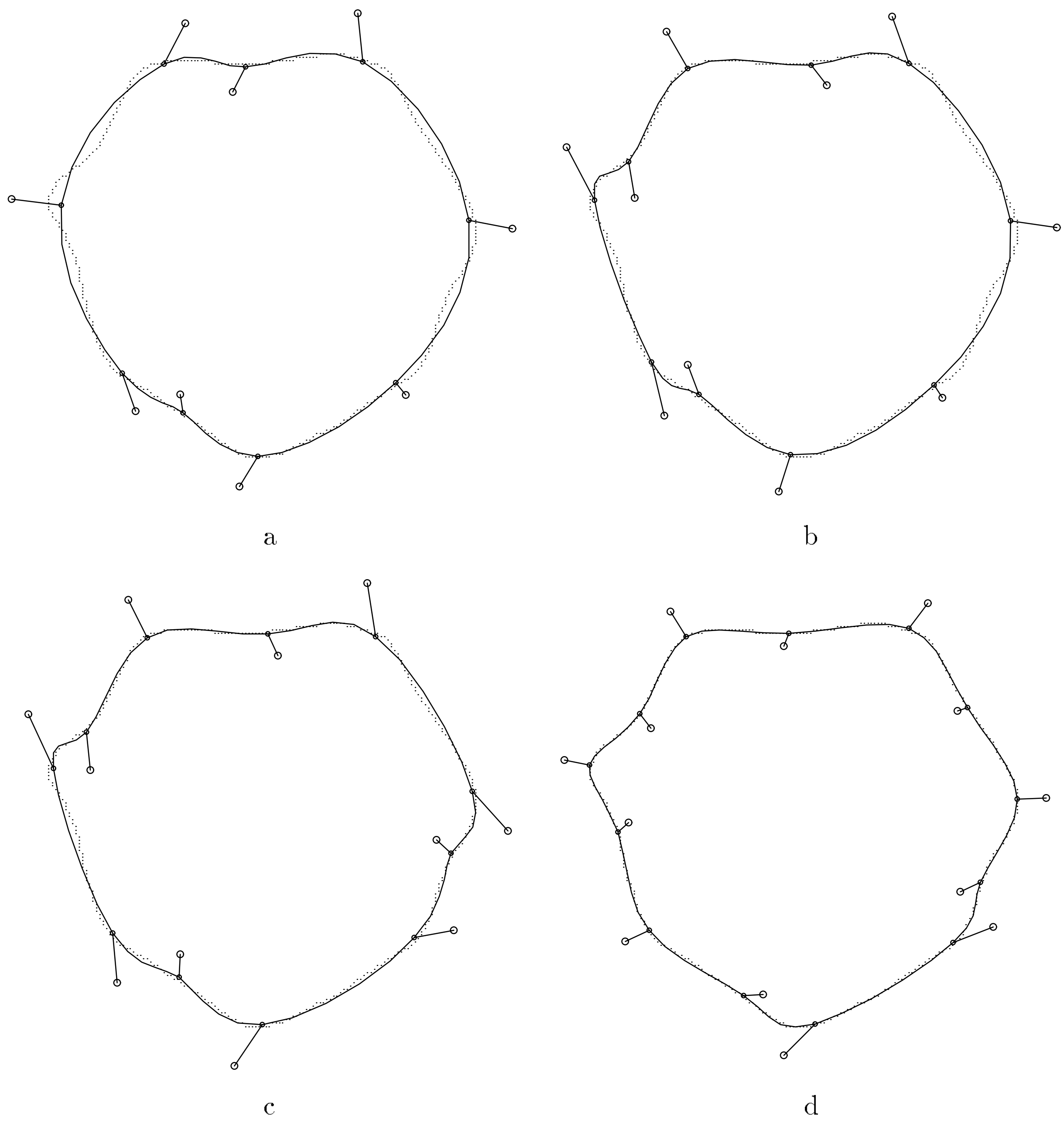

Figure 3: Adding new control points to the spline model: (a) spline with 9 control points (initial result); (b) spline with 10 control points (with 1 added CP); (c) spline with 11 control points (2 added CP's); (d) spline with 13 control points (4 added CP's).

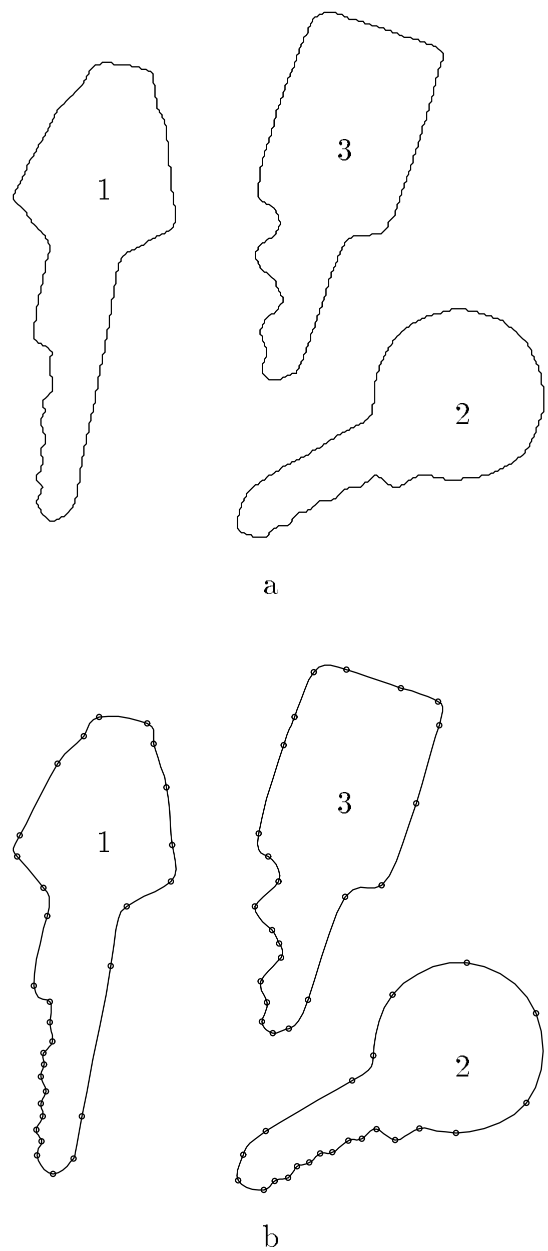

Figure 4: Approximating key contours: (a) original curves; (b) spline approximations with the spline end-points shown as dots.