Curve Recognition Using B-spline Representation†

Fernand S. Cohen, Zhaohui Huang, and Zhengwei Yang
Department of Electrical and Computer Engineering
Drexel University
Philadelphia, PA 19104

Abstract

The B-spline stands as one of the most efficient curve (surface) representation, and possesses very attractive properties such as spatial uniqueness, boundedness and continuity, local shape controllability, and invariance to affine transformations. These properties made them very attractive for curve representation in computer aided design and computer graphics. Very little work, however, has been devoted to them for recognition purpose. One possible reason might be due to the fact that the B-spline curve is not uniquely described by a single set of control points, which made the curve matching (recognition) process not a simple comparison between the respective parameters of the curves to be matched. This paper is an attempt to find matching solutions despite this limitation and addresses the problems of invariant matching and classification of 2-D closed curves with application in identification of aircraft types based on image silhouettes, and writer-identification based on hand written text.

I. Introduction

A number of different approaches have been proposed for curve modeling such as Fourier Descriptors [1]; Chain Codes [2]; Polygonal Approximation [3]; Curvature Primal Sketch [4]; Medial Axis Transform [5]; Autoregressive models [6]; moments [7-8]; parametric algebraic curves [9]; curvature invariant [10]; stochastic transformation [11]; implicit polynomial functions [12]; bounded polynomials [13]; and B-splines [14]; among others. The B-spline has good properties that made them very attractive for curve representation, and consequently they have been extensively used in computer aided design and computer graphics. Very little work, however, has been devoted to them for recognition purposes.

In this paper, we deal with the problem of matching and recognizing planar curves and which are modeled as B-splines, independently of possible transformations that the original curve has been subjected to (e.g., rotation, translation, change in scale, and orthographic or semi-perspective projection transformations), and possibly occlusion. This work is used in the development of an automated or a semi-automated system that allocates a hand-written text to one of a set of possible C writers based on the style of the written message rather than its content. This has direct applications in the banking system for automatic signature identification, and in aiding an expert in hand-writing concentrate on analyzing a manageable set of possible writers (out of a big data base of prior offenders) judiciously chosen by our system for expert witness in court cases. The second application is that of identifying airplane types given images of plane silhouettes taken at different viewing position orientation and scale than the training prototype samples.

This paper is organized as follows. The structure of the B-spline and the reasons for using it as an appropriate model for curve representation are introduced in section II. Section III deals with the problem of estimating the parameters of the B-spline from image curve data. A minimum mean square error (MMSE) procedure is adopted, and the problem for the appropriate assignments of the t parameter is addressed. Section IV introduces the curve matching problem, and sections V through VII present three different methods for matching and classifying curves. In section VIII, we introduce the two applications we have concentrated on, and presents the experimental results. Conclusions are presented in section IX.

II. Projective-invariant curve models

II.1 Uniform Cubic B-Splines

In this paper, we deal with closed curves of C^2 continuous, i.e, the closed curves which have both their slopes and curvatures continuous which are modeled by cubic B-splines. A closed cubic B-spline with n+1 parameters $C_0, C_1, ..., C_n$, (control points) consists of n + 1 connected curve segments $r_i(t) = (x_i(t), y_i(t))$, each of which is a linear combination of four cubic polynomials in the parameter t, where t is normalized for each such segment between 0 and 1 ($0 \le t \le 1$) , i.e.,

$$r_i(t) = C_{i-1}Q_0(t) + C_iQ_1(t) + C_{i+1}Q_2(t) + C_{i+2}Q_3(t) \quad (2.1)$$

† This work was supported by the Federal Bureau of Investigation under grant number J-FBI-91-352.

0-8186-2840-5/92 $03.00 © 1992 IEEE

$$Q_k(t) = a_{k0}\, t^3 + a_{k1}\, t^2 + a_{k2}\, t + a_{k3}, \quad k = 0,1,2,3 \qquad (2.2)$$

Using the continuity constraints in position, slope, and curvature on the connection points between segments, and the invariance property to coordinate transformations (the Cauchy condition [14]) leads to 16 constraint equations from which the polynomial parameters in (2.2) are found. The connection points between curve segments(see figure 1) are called the knot points $\hat{\mathbf{r}}_i$ (where $\hat{\mathbf{r}}_i = \mathbf{r}_i(0) = \mathbf{r}_{i-1}(1)$), i = 0, 1,..., n. Let t' be a variable assuming the values between 0 and n + 1, then the whole curve $\mathbf{r}(t')$ can be expressed as

$$\mathbf{r}(t') = \sum_{i=0}^{n} \mathbf{r}_i(t'- i) = \sum_{i=0}^{n+3} C_{i \bmod(n+1)}\, Q_{i,4}(t') \qquad (2.3)$$

where $\mathbf{r}_i(t'- i)$ is only nonzero for $i \le t' \le i+1$, and $Q_{i,4}(t')$, (i = 0, 1, ..., n), are called the normalized cubic B-spline bases, and are related to each other by horizontal translation (see figure 2).

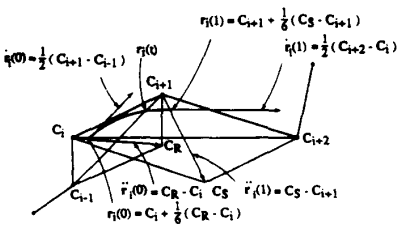


Figure 1

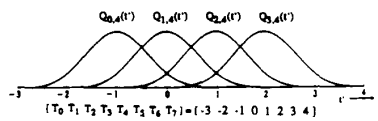


Figure 2

The amplitudes of $Q_{i,4}(t')$ are in the range of (0, 1), and the regions of support of $Q_{i,4}(t')$ are compact and nonzero for $t' \in (i, i+4)$. The values of T_i, i = 0, 1, 2, ..., n + 7, are called knots where the B-spline bases are tied together. They take the values of [-3, -2, ..., n+4]. Under uniform placement of the knots T_i, i = 0, 1, 2, ..., n + 7, given the set $\mathbf{C} = (C_0, C_1, ..., C_n)$, we can uniquely determine the knot points $\hat{\mathbf{r}} = (\hat{\mathbf{r}}_0, \hat{\mathbf{r}}_1, .. \hat{\mathbf{r}}_n)$ and vice versa. The relationship is given as

$$\hat{r}_i = \frac{1}{6}C_{i-1} + \frac{2}{3}C_i + \frac{1}{6}C_{i+1}, \qquad for\ i = 0, 1, ..., n$$

$$C_{-1} = C_n, \quad C_0 = C_{n+1} \qquad (2.4)$$

(2.3) and (2.4) combined, unfortunately, mean that a B-spline curve is not uniquely described by a single set of control points, since with each different choice for the placement of the knot points, we can induce a different set of control points that still describe the same curve. This is of great consequence in the curve matching problem as we are not able to directly compare between their respective parameters.

II.2 Why B-Splines?

In light of the above underlying structure of the B-splines, the following attractive properties make them suitable for shape representation and analysis. These are: (i) smoothness and continuity which allows any curve to consist of a concatenation of curve segments, yet be treated as a single unit; (ii) built-in boundedness, a property which lacking in both the implicit or explicit polynomial representation whose zero set can shoot to infinity [13]. The B-spline on the other hand is bounded by the polygon that join the B-spline parameters; (iii) ease of specifying the range of a multi-valued curve; (iv) the decoupling of the x and y coordinates, with each having its parametric representation, is treated separately; (v) shape invariance under transformation (affine and projective transformations), which means that the projected or transformed curve is still a B-spline whose control points are related to the object control points through the transformation; and (vi) local controllability which implies that local changes in shape are confined to the B-spline parameters local to that change.

III. B-spline parameter estimation

We are given a set of m (m > n+1) ordered curve data points $\mathbf{r} = (\mathbf{r}_1, \mathbf{r}_2, ..., \mathbf{r}_m) = ((x_1, y_1), (x_2, y_2), ..., (x_m, y_m)) = (\mathbf{x}, \mathbf{y})$. The goal is to fit a real-valued B-spline to the observed curve data. There exist two obvious ways of achieving that. The first approach uses a B-spline which interpolates between the sampled data points. The second one tries to find an approximate B-spline such that the "error" measure between the observed data and their corresponding B-spline curve values are small. This approach, unlike the previous one, does not require as many parameters, and is more resilient to noise, and is therefore the one adopted here. There are two main points that need to be specified for adopting the latter approach. These are : (i) the error distance between the curve data and their corresponding B-spline points should be specified; and (ii) appropriate values of t' should be allocated to the curve data. In this section both points are addressed.

III.1 Minimum Mean Square Error Estimation

Since the B-spline is factorizable into its different components, estimates for the control points based on the 2-D curve data can be addressed for each components separately. Let f stands for either the x-component or the y-component. Then the MMSE estimates are obtained by

minimizing the sum d^2 in (3.1) of the square residual errors between the data point $\mathbf{r}_j$ and its corresponding point on the B-spline $\mathbf{r}(t'_j)$, j = 1, 2, ..., m, with respect to the control points i.e.,

$$d^2 = \sum_{j=1}^{m} d_j^2 = \sum_{j=1}^{m} (d_j^2(x) + d_j^2(y))$$
$$= \sum_{f=x}^{y} \sum_{j=1}^{m} [f_j - \sum_{i=0}^{n+3} C_{imod(n+1),f} Q_{i,4}(t'_j)]^2 \qquad (3.1)$$

If the values of the t'_j's in (3.1) are known, the MMSE solution can be readily obtained.

III.2 Iterative B-spline Parameter Estimation

In order to arrive at the solutions, appropriate values for the t'_j's should be determined. This section deals with that problem. One possibility is to treat them as unknown parameters to be estimated along with the control points. The problem with such an approach is twofold: (i) the parameter space is bigger than the data space; and (ii) the minimization problem is highly nonlinear and computationally very intensive. To bypass these two problems we adopt an iterative procedure similar in spirit to the EM algorithm in maximum likelihood estimation. The steps in this iterative minimization process are as follows:

1. Choose an initial values for the t'_j(initial)'s;
2. Compute the MMSE for the control point parameters $\mathbf{C}^*$(initial) based on the assigned t'_j(initial) values, compute the sum of the residual errors d^2(initial) in (3.1);
3. Update the values of t'_j(new) by minimizing d^2 in (3.1) with the control point parameters fixed to $\mathbf{C}^*$(initial). Based on t'_j(new), find $\mathbf{C}^*$(new) and compute d^2(new).
4. If d^2(new) < d^2(initial) and $|d^2$(new) - d^2(initial)| > threshold then set new to initial and go back to 3, otherwise go to 5.
5. Stop, the MMSE estimates are $\mathbf{C}^*$(initial) in step 3.

This iterative minimization is guaranteed to converge to at least a local minimum of d^2 in (3.1). An initial assignment for the t'_j values can be made according to the chord length method in [16] which was shown to be a good non-iterative, simple, and fast approach for control point estimation [16]. Here, the t'_j values are assigned according to the length of the chord relative to the total chord length. The update for the values of t'_j for a fixed set of control points is achieved according to the following. Let t'_j(old) be the value of t'_j before the update. First we confine the search region to the interval I = [Integer{t'_j(old)} - 1, Integer{t'_j(old)} + 1]. This interval corresponds to the three B-spline curve segments confined by the knots k - 1, k, k + 1, and k + 2, where k = integer{t'_j(old)}. The best value t'_j(new) for t'_j that minimizes d^2 in (3.1) confined to the interval is found using a golden section search. The update for the next t' value t'_{j+1} is obtained in a similar fashion except that it is not allowed to assume a value which is smaller than t'_j(new), i.e., the search imposes the order constraint $t'_1 < t'_2 < ... < t'_m$. The improvement of this iterative optimization over the MMSE estimation with the chord length assignment is illustrated in section VIII.1.

IV. Invariant matching of curves

The goal here is to match and recognize planar curves which are modeled as B-splines, independently of possible affine transformations that the original curve has been subjected to, and possible occlusion. The scenario discussed here is one where we have a set of P prototype object planar curves modeled as B-splines, and stored in standard positions. It is assumed here that the different object curves are not related to each other through an affine transformation. We are presented with a set of sampled curve data points from the transformed object curve, and are asked to recognize (classify) the object curve from the sample curve data points. As pointed in section II.1, because of the non-uniqueness of the control points in describing any of the prototype curves, we can not directly use the estimated control points in the matching process. Three different solutions to that problem are presented.

The first method is based on differential geometry through the use of curvature κ (and torsion τ in the case of nonplanar curves) measures derived from the approximating B-spline. κ and τ are intrinsic properties of the curve shape, which by the Fundamental theorem of curves [15] completely determine the curve shape up to position. The curvature has a parametric representation in terms of t' (or arc length s), is continuous as a consequence of the C^2 continuity of the B-spline, and has a closed-form analytic expression. Moreover, when considering the parametric representation of the curvature in terms of arc length s, $\kappa(s)$ becomes invariant to translation and rotation, and could be made invariant to scale. It can also be used in the occlusion case. It is, however, variant to affine and projective transformations and sensitive to noise since the curvature involves taking second derivative of the curve.

The second method is based on relating the control points (or equivalently the knot points) of the prototype curve to the estimated control points of the sample curve after a judicious reassignment (displacement) of the knot

points associated with the sample curve in accordance with the placement of the knot points for the prototype curve. This method can be made invariant to affine transformation. Being a global comparison rather than a local one, i.e., a comparison between parameters rather than points, it is more robust than the curvature-based method as it is more resilient to noise. It, however, can not handle the occlusion case.

The third method is a residual error-based method which compare the sum of the residual error between the sample and the different prototype curves. It is the most general of the three methods. It is invariant to affine transformations, and can handle occlusion. This invariance is achieved by adjusting the control points by multiplying them with the "best" affine transformation that makes the residual error between the sample curve and each of the prototype curves the smallest.

These three methods are described in details in sections V, VI, and VII, respectively.

V. Curvature-based matching

We start the discussion by introducing the curvature measure. From now on, the dot(s) over a variable means its derivative(s) with respect to its argument. Let $\mathbf{r}(t')$ be a B-spline (a parametric regular curve) that resides on a plane in R^3 , i.e., $\mathbf{r} : (a, b) \rightarrow R^3$ The spline curve is C^2 continuous. Let $v(t') = \dot{\mathbf{r}}(t')$ be the velocity vector field, and let $\mathbf{T}(t')$ be the tangent vector $\frac{v(t')}{\|v(t')\|}$. The curvature $\kappa(t')$ which measures the degree of bending of the curve on the surface is given by [15]

$$\kappa(t') = \frac{\|v(t') \times \dot{v}(t')\|}{\|v(t')\|^3} \qquad (5.1)$$

Since the length of a curve is a geometric property, we reparametrize it in terms of arc length. We concentrate on any one of the curve segment $\mathbf{r}_i(t)$ with $t \in [0,1]$. Let $\mathbf{R}_i(s)$ be the reparametrization of $\mathbf{r}_i(t)$ in terms of arc length s, i.e., $\mathbf{R}_i(s) = \mathbf{r}_i(t = h^{-1}(s))$. If $\mathbf{r}_i(t)$ is viewed as the path of a particle moving in space, then $\|v_i(t)\| = \dot{r}_i(t')$ is its speed, and the integral of speed gives the distance s travelled by the particle in the time interval t, i.e.,

$$s = h(t) = \int_0^t \|v_i(t)\| \, dt \qquad (5.2)$$

$s = h(t)$ is the arc length along $\mathbf{r}_i(t)$. It is 1-1 mapping of the interval (0, 1) onto some interval $(0, L_i)$. The velocity field $\mathbf{T}_i(s) = \dot{\mathbf{R}}_i(s)$ is a unit vector. The curvature of $\mathbf{R}_i(s)$ is

$$\kappa_i(s) = \|\dot{T}_i(s)\| \qquad (5.3)$$

In order to evaluate $\kappa_i(s)$ in (5.3) we have to compute the integral in (5.2). The integrand in (5.2) has the form $\sqrt{at^4 + bt^3 + ct^2 + dt + e}$.

For a closed curve with n+1 curve segments, let $d_k = \sum_{i=0}^{k-1} L_i$, k = 1, ..., n, with $d_0 = 0$, be the distance travelled on the curve from the starting knot point $\hat{r}_0$ until the knot point $\hat{r}_k$, and let s' be a variable assuming the values between 0 and $d = d_{n+1} = d_n + L_n$ (total distance travelled on the curve to go back to the starting point), then the curvature function $\kappa(s')$ for the whole curve can be expressed as

$$\kappa(s') = \sum_{i=0}^{n} \kappa_i(s' - d_i) \qquad (5.4)$$

where $\kappa_i(s' - d_i)$ is nonzero only for $d_i \le s' \le d_{i+1}$. $\kappa(s')$ is continuous and periodic in s' with period d, and is invariant to shift and rotation, but variant to the choice of the starting point and to change in scale. For classification, we use a set of sampled curvature points $\{\kappa(j), j = 0, 1, 2, ..., N-1\}$ obtained by uniformly sampling the continuous curvature function $\kappa(s')$, where $\kappa(j) = \kappa(s' = j \Delta s'), j = 0, 1, 2, ..., N-1$, with $\Delta s' = \frac{d}{N}$ chosen so that the curvature is almost constant within the $\Delta s'$ interval (i.e., $|\kappa((j+\delta)\Delta s') - \kappa(j \Delta s')| << |\kappa(j \Delta s')|$, for $0 \le \delta \le 1$). For a curve which is scaled by a factor β with respect to the original curve, if we impose the same number of points N on the scaled curve as for the unscaled curve, we can achieve the invariance to scale, i.e.,

$$\kappa(j) = \kappa_\beta(j), \qquad j = 0, 1, 2, ..., N-1 \qquad (5.5)$$

where $\kappa_\beta(.)$ is the curvature of the scaled curve. Finally, to alleviate the variance of the curvature measure due to the starting point, we find the shift l that matches the observed curvature $\kappa_o(j)$ to the prototype curvature $\kappa_p(j)$. The classifier is based on the Euclidean distance and is given by

$$\min_{1 \le p \le P} \{ \min_{0 \le l \le N-1} \{ \sum_{j=0}^{N-1} [\kappa_o((l+j) \bmod N) - \kappa_p(j)]^2 \} \} \qquad (5.6)$$

where P is the total number of different prototype curves. $\kappa_o(j)$ is obtained from the sample curve by first estimating the control points from the curve as described in section III, then computing $\kappa_o(s')$ using (5.3) and uniformly sampling the range of s' $(0 \le s' \le d_o)$ to obtain $\kappa_o(j)$, j = 0, 1, ..., N-1.

V.1 Matching in the Presence of Occlusion

In the case where we are presented with only a part of the sample curve due for example to occlusion, we first estimate the control points for now the open curve, then we compute the curvature as in (5.3), and sample it using the sampling rate used for the prototype curves. Here for the observed curve we wind up with a smaller number of sample curvature points say M (M < N). We can then use the classifier in (5.6) with the upper limit in the summation sign replaced by M-1. If we also allow scaling as well as occlusion, then the classifier should be modified to

$$\min_{1\le p\le P} \{\min_{\Delta s'} \{\min_{0\le l\le N-1} \{ \sum_{j=0}^{M_{\Delta s'}-1} [\kappa_o((l+j) \bmod N) - \kappa_p(j)]^2 \}\}\} \quad (5.7)$$

Where $\Delta s'$ is sampling interval, $M_{\Delta s'}$ is its sample number.

VI. Matching using knot points

The main question that this section attempts to answer is : how can we match two curves based on comparing their control points when a curve could be described by more than a single set of control points as pointed out in section II.1? Towards that end, let us consider a set of P different curves $\mathbf{r}_p(t')$, p = 1, 2,..., P, whose centroids coincide and is taken as the origin of the curve coordinate system. For each curve, a set of n control points $C_p = (C_0(p), C_1(p), ..., C_{n-1}(p))$ is stored and used as a descriptor for the curve. Mother nature picks one of the curve, a set of sample ordered curve points $(\mathbf{r}_1, \mathbf{r}_2, ..., \mathbf{r}_M)$ then is chosen. From this data a set of control points $C = (C_0, C_1, ..., C_{n-1})$, is estimated as described in section II. Let this curve be denoted by r(t'). As pointed out in section II.1, we can not directly compare C with the different C_p's, as there are no guarantees that they correspond. In light of that fact, the problem is now of finding that set C^* (or equivalently the set of knot points $\hat{\mathbf{r}}^*$) that is guaranteed to correspond to C_p if the curve that mother nature picked up were the p^{th} curve. The set C^* is determined from the knowledge of C and C_p as follows. Since the knot points and control points are 1-1, we compute the set $\hat{\mathbf{r}}_p = (\hat{\mathbf{r}}_0(p), \hat{\mathbf{r}}_1(p), .. \hat{\mathbf{r}}_{n-1}(p))$ from the set C_p using (2.4). Let $\mathbf{r}_p(t_o)$ be the point on the p^{th} B-spline that intersects the positive x-axis, and let d(p) be the total distance traveled on the p^{th} B-spline. Moving in an anti-clockwise direction, we compute the fraction of the distance traveled from the point $\mathbf{r}_p(t_o)$ to the first knot point we encounter, relative to the total distance d(p), then to the second, and so on until we have considered the n knot points. Let $f_0(p), f_1(p), ..., f_{n-1}(p)$, be these fraction distances. We renumber the knot points in order of first encounter. Let the reordered set be denoted as $(\hat{\mathbf{r}}'_0(p), \hat{\mathbf{r}}'_1(p), .. , \hat{\mathbf{r}}'_{n-1}(p))$. Next for the curve r(t'), we compute the total distance d traveled on the curve, and find the point $\mathbf{r}(t_a)$ on the curve that intersects the positive x-axis. Moving again in an anti-clockwise direction, we place the first knot point $\hat{\mathbf{r}}^*_0(p)$ on the curve r(t') such that the fraction of the distance traveled from the point $\mathbf{r}(t_a)$ to $\hat{\mathbf{r}}^*_0(p)$ relative to the total distance d is the same as $f_0(p)$. The rests of the knots $\hat{\mathbf{r}}^*_1(p), \hat{\mathbf{r}}^*_2(p), ..., \hat{\mathbf{r}}^*_{n-1}(p)$, are placed similarly, so as the fraction of their distances from $\mathbf{r}(t_a)$ relative to the total distance d coincide with the rest of the fractions $f_1(p), ..., f_{n-1}(p)$. The classifier based on the knot points is according to

$$\min_{1\le p\le P} \{ \sum_{i=0}^{n-1} \| \hat{\mathbf{r}}'_i(p) - \hat{\mathbf{r}}^*_i(p) \|^2 \} \quad (6.1)$$

Remark:

If the curve is an affine transformation of the prototype, the affine transformation parameters are first found as explained for example in [17] using the continuous curve moments (or weighted moments). We then undo the transformation, and follow the procedure above in the absence of transformation. This can also be used in the residual-error method presented next.

VII. Residual error-based matching

For the residual error-based method, we form the sum of the residual errors of the data to to the p^{th} B-spline using the invariance property to the affine transformation

$$\begin{aligned} d_p^2 &= \sum_{j=1}^{M} \|d_p(j)\|^2 = \sum_{j=1}^{M} \|\mathbf{r}_j - \mathbf{r}_{p,T}(t'_j)\|^2 \\ &= \sum_{j=1}^{M} \|\mathbf{r}_j - \sum_{i=0}^{n+3} C_{imod(n+1)}(p, T)\, Q_{i,k}(t'_j)\|^2 \\ &= \sum_{j=1}^{M} \|\mathbf{r}(j) - \sum_{i=1}^{n} [\beta [R]\, C_{imod(n+1)}(p) + b]\, Q_{i,k}(t'_j)\|^2 \quad (7.1) \end{aligned}$$

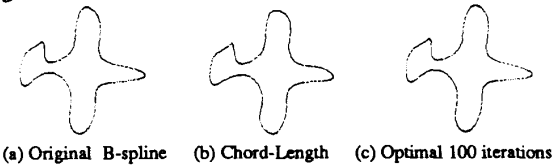
Any one of the t'_j values is computed as described in section III.2. In order to compute (7.1), the transformation parameters θ, b_x, b_y, and β have to be known. These are estimated using MMSE estimation obtained by minimizing (7.1) with respect to these parameters. The transformation parameters are computed under each class p assumption, and the different d_p^2's are computed. The sample is allocated to the class with the smallest d_p^2

value. For a more general affine transformation matrix [A], the MMSE procedure will be performed with respect to the four elements of the matrix [A].

VIII. Applications

VIII.1 B-spline parameter estimation based on the iterative t' value assignment

We use the iterative procedure described in (III.2) to assign t'j value of the cubic B-spline which approximates the given data. The data is obtained by uniformly sampling a known B-spline curve parametrized by 20 control points (see figure 3a). The first case is noiseless. The second case is when the data is contaminated by a Gaussian noise with Mean = 0.0, Variance = 0.680. The maximum squared error was found to be 5.740. As in the noiseless case, we also used a 20 control points cubic B-spline to approximate the given data. The results of noiseless case are shown in figures 3b and 3c. The average and maximum squared errors for the two methods are given in table 1.



(a) Original B-spline (b) Chord-Length (c) Optimal 100 iterations Reconstructed

Figure 3

Squared error of recovered B-spline to original B-spline data (without and with noise)

Iterations	Case1 (without noise)		Case 2 (with noise)	
	Average	Maximum	Average	Maximum
1 (Chord - length)	0.90887	11.80	1.30	19.08
10	0.03289	0.98	0.71	7.02
20	0.01963	0.73	0.65	6.87
30	0.01176	0.53	0.63	6.90
100	0.00122	0.14		

Table 1

The iterative algorithm of (III.2) improves on the error. For both cases, we recovered original B-spline with very high accruracy.

Remarks:

(i) In our experiment, we found that the error monotonically decreases as iteration grows and converges after about 100 (without noise) and 30 (with noise) iterations;

(ii) On the average the time per iteration (t' values assignments and control point estimation) was about 5 seconds.

VIII.2 Identification of Aircraft Types

Curvature-Based Method

The first application is the identification of objects like planes based on their extracted silhouettes. A set of 9 prototype curves (which are silhouettes of 9 planes) with about 500 curve points each are shown in Figure 4. A set of 100 control points is computed for each prototype curve and is stored.

We are given 6 test curves (planes) which were: rotation, translation, scale of the prototypes for the first two test curves; orthographic transformation of the prototype for the third, semi-perspective for the fourth, perspective for the fifth, and occlusion for the sixth, respectively (see figure 5).

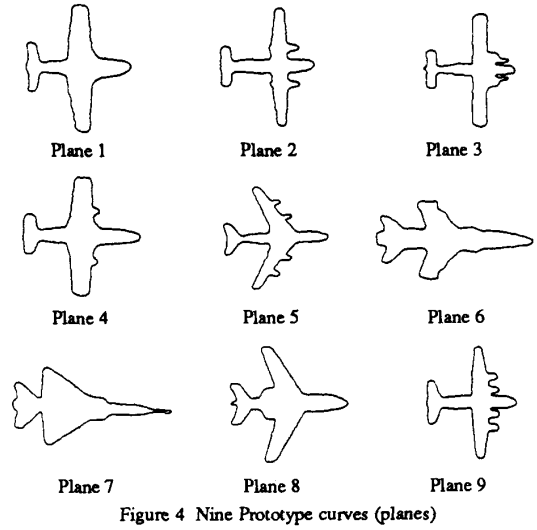


Figure 4 Nine Prototype curves (planes)

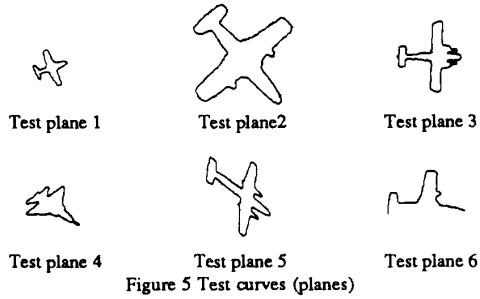


Figure 5 Test curves (planes)

We used the classifier (5.6) ((5.7) in the occlusion case) and obtained the following classification results displayed in table 2:

Average error of test to prototype planes (curvature-based method):

	pro 1	pro 2	pro 3	pro 4	pro 5	pro 6	pro 7	pro 8	pro 9
test 1	0.121	0.264	0.382	0.277	0.443	0.259	0.259	0.298	0.384
test 2	0.225	0.260	0.374	0.080	0.456	0.275	0.287	0.312	0.362
test 3	0.329	0.363	0.339	0.376	0.575	0.337	0.361	0.402	0.442
test 4	0.251	0.305	0.371	0.262	0.464	0.295	0.290	0.324	0.366
test 6	0.195	0.251	0.280	0.005	0.300	0.229	0.216	0.219	0.278

Table 2

The classifier properly allocated the first two and the last tested samples, i.e., test plane 1 to prototype 1, test plane 2 to prototype 4, and test plane 6 (which is occluded) to prototype 4. The classification for the other three tested

samples were as expected not correct, due to the variance of the curvature to orthographic, semi-perpsective and perspective projections (only the test 3 and test 4 results are listed in table 2). Finally, the computational costs are around 8 seconds.

Knot-Based Method

The experiment has been repeated using the knot point method. The classification results are shown in table 3 for the first 5 cases. For the perspective case (case 5) we have used the semi-perspective projection as if it were the true projection model. All the five cases have been properly classified.

Average error for test planes to prototype planes (knot point method):

	pro 1	pro 2	pro 3	pro 4	pro 5	pro 6	pro 7	pro 8	pro 9
test 1	4.03	16.83	18.00	14.34	34.00	46.85	61.83	18.51	18.86
test 2	60.55	19.09	24.06	1.77	41.47	33.74	61.49	42.21	16.88
test 3	17.86	11.12	4.81	15.14	27.79	43.34	52.90	36.41	9.85
test 4	33.45	35.20	32.43	28.61	37.33	38.80	2.77	26.85	34.66
test 5	19.61	3.60	9.56	20.16	32.42	35.36	43.77	27.15	12.42

Table 3

Residual Error-Based Method

Finally, the experiments are run on the six cases based on the residual error-based method and yielded the classification results shown in table 4. For the perspective case we have used the semi-perspective projection as if it were the true projection model.

Average Error of test to prototype planes (residual error-based) :

	pro 1	pro 2	pro 3	pro 4	pro 5	pro 6	pro 7	pro 8	pro 9
test 1	3.84	22.34	29.31	39.86	28.11	43.46	46.08	42.18	24.88
test 2	41.69	68.31	39.27	5.87	60.40	36.95	43.29	26.14	66.32
test 3	19.56	13.52	5.73	14.91	23.41	33.27	36.74	22.59	11.04
test 4	37.58	36.64	31.94	29.84	36.58	30.51	6.78	30.18	32.26
test 5	17.14	5.12	12.71	18.88	30.42	39.24	37.30	26.70	13.49
test 6	37.37	36.24	42.27	21.56	43.27	56.43	73.75	33.47	40.55

Table 4

The classifier properly allocated the tested samples. The reconstructed curves are almost identical to those shown in figure 4.

VIII.3 Writer-Screening Identification

The second application aims at the allocation of a hand-written text to one of a set of possible P writers based on the style of the written message rather than its content. A written text is allocated to a writer by first isolating and identifying the letters, then using each writer's characteristic shape of the letters, we allocate the message to the most likely writer. This is an intrinsic part in a system whose aims are: (i) to allocate a given handwritten message to a reduced set of writers out of a much larger set of possible writers in a data base, and (ii) to have quantitative measures that reflect the confidence level (likelihood) of the association of the given message with each writer in the reduced set. The experiment consisted of ten writers, each of which has written the letters of the alphabet ten times.

Nine of these samples were used for training. For each letter and writer, these nine training samples were used to estimate the control points which represent the particular letter and writer. The rest of the samples (260 samples) were used for classification. A set of experiments were run on single letters as well as combination of letters from the same writer (see figure 6 for an example) to test the classification method. The experimental results obtained using the residual-based decision rule outlined in section VII are summarized in the Tables 5 and 6.

	w1	w2	w3	w4	w5	w6	w7	w8	w9	w10
a	1	5/9	1	5/9	1	1	1	1	1	8/9
b	1	6/9	1	8/9	1	4/9	1	1	5/9	1
c	1	7/9	1	8/9	1	1	1	8/9	1	1
d	1	1	7/9	1/9	8/9	1	1	1	1	1
e	1	1	1	1	1	1	1	1	8/9	1
f	1	8/9	1	1	1	1	1	6/9	1	1
g	1	1	7/9	1	8/9	1	1	1	1	7/9
h	1	1	1	1	1	1	1	1	1	8/9
i	1	1	1	7/9	1	1	1	1	1	1
j	1	8/9	1	5/9	7/9	7/9	1	1	1	1
k	1	1	1	8/9	1	1	1	1	1	1
l	1	1	1	6/9	1	1	1	8/9	1	8/9
m	1	1	8/9	1/9	1	5/9	1	1	1	1
n	1	1	1	1/9	1	1	1	1	1	1
o	1	1	1	1	8/9	8/9	1	1	1	8/9
p	1	1	6/9	1	1	1	1	1	1	1
q	1	1	1	1	1	8/9	1	1	1	1
r	1	1	1	1	1	6/9	1	1	1	1
s	1	8/9	1	1	1	6/9	1	1	1	1
t	1	1	1	8/9	1	1	8/9	7/9	1	1
u	1	1	1	1	1	1	1	1	1	1
v	1	1	8/9	1	1	1	1	1	8/9	2/9
w	1	1	1	6/9	1	1	1	1	1	1
x	1	1	8/9	8/9	1	1	1	1	1	2/9
y	1	1	1	1	5/9	1	8/9	1	1	8/9
z	1	1	1	1	1	1	1	1	1	1

Table 5. Single Letter Recognition Rate

The entries in Table 5 show the recognition rate (for each letter) which indicates the number of writers (not including the correct writer) that had their residual errors higher than the one associated with the correct writer. This number is divided by 9 and is displayed in the table as a classification rate with 1 being the highest and 0 the lowest. As shown in Table 5, the recognition rate is different for the different writers and the different letters. We found that the more consistent the handwriting was, the higher was the recognition rate, e.g., writer1 and writer7. Obviously, the classification based on only single letter is neither reliable nor reasonable, and hence we consider the classification based on a number of letters jointly. The classification was based on a majority rule decision which allocates the sample to the writer for whom the single letters have been allocated to the most (i.e., the one with the highest frequency of occurrence) To illustrate this majority rule, we consider the example shown in Table 6 for a sample of combination of letters

from writer2. The tested sample from writer2 is shown in figure 6, and for comparison we also display the sample from writer9 in figure 7.

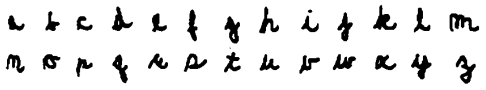


Figure 6 Tested Sample from writer2

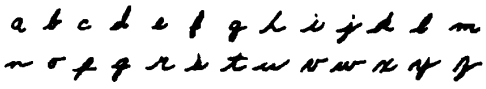


Figure 7 Sample from writer9

As shown in table 6, we can not recognize the single letter "a" correctly because the residual of writer9 is smaller than that of writer2. When "a" and "b" are considered jointly, the recognition rate is still in favor of writer9. For the letters "a", "b", and "c" jointly, we can see that the decision is still in favor of writer9 since two out of the three letters were classified as writer9. As we consider more and more letters, more and more letters are properly classified, and the decision tilts in favor of writer2.

Joint Letter(s)	Recognition rate	Classification
a	0/1	wrong
a,b	0/2	wrong
a,b,c	1/3	wrong
a,b,c,d	2/4	uncertain
a,b,c,d,e	3/5	correct
a,b,c,d,e,f	4/6	correct
a -- g	5/7	correct
a -- h	6/8	correct
a -- i	7/9	correct
a -- j	8/10	correct
a -- k	8/11	correct
a -- l	9/12	correct
...	...	...
a -- z	23/26	correct

Table 6. Joint Letters Classification

We used this majority rule for classifying ten samples each consisting of the 26 letters that were not used in the training. A 100% classification rate was achieved. The computing time for estimating the control points and evaluating the residual is around 1 second.

Finally, note that the results obtained using the curvature method were inferior to that obtained using the residual error method, and were comparable to the knot points matching method.

IX. Conclusions

Three curve matching and identification methods, which exploited different aspects of the B-spline, were presented, and were useful in shedding light on how can the nice properties associated with the B-spline facilitate and robustify the curve recognition process. By and large, each method by itself had merit, and could be used in conjunction with the other two. The approach was tried on real problems with real data, and the results were encouraging and promising.

References

[1] E. Persoon and K. S. Fu, "Shape Discrimination Using Fourier Descriptors," IEEE Trans. Pattern and Mach. Intell., vol. PAMI-8, May, 1986, pp. 388 - 397.

[2] J. A. Saghri and H. Freeman, "Analysis of the Precision of Generalized Chain Codes for the Representation of Planar Curves," IEEE Trans. Pattern and Mach. Intell., vol. PAMI-3, Sep. 1981, pp. 533 - 539.

[3] T. Pavlidis and F. Ali, "Computer Recognition of Handwriten Numerals by Polygonal Approximations," IEEE Trans. Systems, Man, Cybern., vol. SMC-6, Nov., 1975, pp. 610 - 614.

[4] H. Asada and M. Brady, "The Curvature Primal Sketch," IEEE Trans. Pattern and Mach. Intell., vol. PAMI-8, Jan. 1986, pp. 2 - 14.

[5] O. Philbrick, "Shape Description with the Medial Axis Transformation," in *Pictorial Pattern Recognition*, G. C. Cheng, Ed. Washington, D. C.: Thompson, 1968, pp. 395 - 407.

[6] R. L. Kashyap and R. Chellappa, "Stochastic Models for Closed Boundary Analysis: Representation and Reconstruction," IEEE Trans. Inform. Theory, vol. IT-27, Sep. 1981, pp. 627 - 637.

[7] M-K Hu, "Visual Pattern Recognition by Moment Invariant," IRE Trans. on Information Theory, Vol. 8, pp. 179-187, 1962.
Vol. PAMI-13, No. 11, pp. 1115- 1138, Nov. 1991.

[8] G. Taubin and D. Cooper, "2D and 3D Object Recognition and Positioning System Based on Moment Invariant," Proceedings of the DARPA-ESPRIT Workshop on Geometric Invariant, Rikjavik, Iceland, May 1992.

[9] J. Ponce and D. J. Kriegman, "On Recognizing and Positioning Curved 3D Objects from Image Contours," Proceedings of the IEEE Workshop on Interpretation of 3D Scenes, Nov. 1989.

[10] P. J. Besl and R. C. Jain, "3D Object Recognition," Computer Surveys, Vol. 17, No. 1, March 1985.

[11] U. Grenander and D. M. Keenan, "Towards Automated Image Understanding," Applied Statistics, Vol. 16, pp. 207-221, 1989.

[12] D. Forsyth, J. Mundy, A. Zisserman, and C. Brown, "Projective Invariant Representation using Implicit Algebraic Curves," Proceedings of the European Conference on Computer Vision, 1990.

[13] D. Keren, D. Cooper, and J. Subrahmonia, "Describing Complicated Objects by Implicit Polynomials," Reprint, LEMS Tech. Report, Brown University, Oct. 1991.

[14] C. de Boor, *A Practical Guide to Splines*, Springer-Verlag, 1978.

[15] R. S. Millman and G. D. Parher, *Elements of Differential Geometry*, Prentice Hall Inc., Englewood Cliffs, NJ, 1977.

[16] F. S. Cohen and J.-Y. Wang, "3-D Recognition and Shape Estimation from Image Contours," Proceedings of 1992 IEEE Conference on Computer Vision and Pattern Recognition, Urbana Champaign, Illinois, June 1992.

[17] F. S. Cohen, Z. Huang, and Z. Yang, "Invariant Matching and Identification of Curves Using B-spline Curve Representation," Under review for journal publication.