

Curve Matching by using B-spline Curves

Tet Toe

Faculty of Engineering
Assumption University
Ramkhamhaeng 24, Huamark
Bangkok 10240, Thailand

E-mail:engtto@maia.au.ac.th

Tang Van To

Faculty of Science and Technology
Assumption University
Ramkhamhaeng 24, Huamark
Bangkok 10240, Thailand

E-mail:tvto@s-t.au.ac.th

ABSTRACT

This paper presents an algorithm for estimating the control points of the B-spline and curve matching which is achieved by using the dissimilarity measure based on the knot associated with the B-spline curves. The B-splines stand as one of the most efficient curve representations and possess 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, and consequently, they have been extensively used in computer-aided design and computer graphics. The curve matching program is shown in this paper. Any input test object curve can be matched with the B-spline sample curve. The control points of sample curve are computed and stored in the program. The test object curve, a bitmap file, is thinned then converted to B-spline curve and then to match with the sample curve.

Keywords

B-spline curve, character recognition, curve matching, scaling, thinning, position vector, interdependence, dissimilarity, control points, data points, chord length, transformations, test object curve, sample curve, bitmap file.

1. INTRODUCTION

There are many techniques for curve matching. The B-spline stands as one of the most efficient curve representation, and possesses very attractive properties such as spatial uniqueness, boundedness and continuity, local shape controllability, and structure preservation under affine transformation [4]. Because of these properties they can be used for represent curves. To recognize the handwritten characters we should notice that the characters are differed for different persons, different writing ways, different sizes, different directions, different stretching ways, and so on. For the recognition of handwritten characters we deal with matching the curves which are modeled as B-splines.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

WSCG POSTERS proceedings
WSCG'2004, February 2-6, 2004, Plzen, Czech Republic.
Copyright UNION Agency - Science Press

Curve matching is achieved by choosing the best B-spline curve with invariance affine transformation. This affine transformation must be estimated. At first the control points of the B-spline are estimated. Then the best order B-spline and the best number of control points are decided. To solve the problem of curve matching the main objectives of this work are as the following :

- 1) Skeletonization
- 2) Determination of B-spline control points for the curve
- 3) Matching the curve using dissimilarity between their control points.

2. SKELETONIZATION

To make thinner form the input curve must be skeletonized according to the Thinning Algorithm [1]. THINNING is a process which transform a binary image into a line skeleton of unit thickness. The major functions of thinning in image processing are:

1. to reduce data storage,
2. to reduce transmission time,
3. to facilitate the extraction of morphological features from digitized pattern.

An input curve can be thinned by using Thinning Algorithm . A sence of scaling and thinning the curve is shown in Fig. 1. In this figure, the bitmap is scaled and thinned.

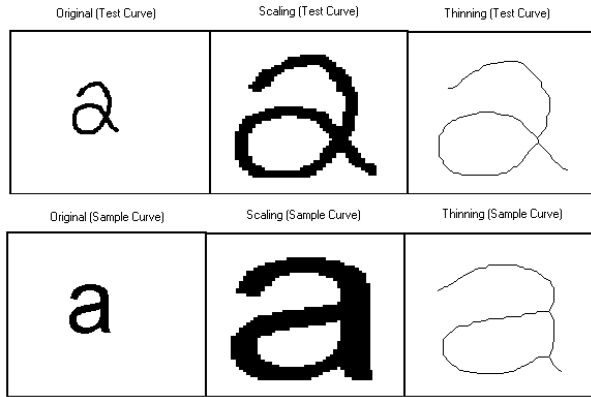


Figure 1. Scaling and Thinning

3. DETERMINATION OF B-SPLINE CONTROL POINTS FOR THE CURVE

Let $r(t)$ be the position vectors along the curve as a function of the parameter t , a B-spline curve is given by [3]

$$r(t) = \sum_{i=1}^{n+1} C_i Q_{i,k}(t), \quad t_{\min} \leq t < t_{\max}, \quad 2 \leq k \leq n+1 \quad (3-1)$$

where C_i are the position vectors of the $n+1$ control points and the $Q_{i,k}(t)$ are the normalized B-spline basis functions.

For the i^{th} normalized B-spline basis function of order k , the basis functions $Q_{i,k}(t)$ are defined by the Cox-deBoor recursion formulas. Specifically,

$$Q_{i,0}(t) = \begin{cases} 1 & \text{if } t_i \leq t \leq t_{i+1} \\ 0 & \text{otherwise} \end{cases} \quad (3-2)$$

and

$$Q_{i,k}(t) = \frac{t-t_i}{t_{i+k}-t_i} Q_{i,k-1}(t) + \frac{t_{i+k+1}-t}{t_{i+k+1}-t_{i+1}} Q_{i+1,k-1}(t) \quad (3-3)$$

In Equation (3-3), we interpret the terms

$$\frac{t-t_i}{t_{i+k}-t_i} Q_{i,k-1}(t) \text{ and } \frac{t_{i+k+1}-t}{t_{i+k+1}-t_{i+1}} Q_{i+1,k-1}(t) \text{ as } 0$$

when their denominators vanish. Equation (3-3) suggests a computational procedure for evaluating a B-spline at some point t . We observe that on any given interval $[t_i, t_{i+1}]$ there are only $k+1$ B-spline basic functions of degree k that are nonzero. On that interval, $Q_{i,k}(t)$ depends only on $Q_{i,k-1}(t)$ because $Q_{i+1,k-1}(t)$ is zero there while $Q_{i-l,k}(t)$ ($0 < l \leq k$) depends on both $Q_{i-l+1,k-1}(t)$ and $Q_{i-l,k-1}(t)$. The interdependence of the B-splines is shown in Fig.2.

In order to find the splines of degree k , we must find $k-1$ previous levels in the diagram shown in that figure and at each level we must find the B-splins $Q_{i,j}(t)$ to $Q_{i-l,j}(t)$, where j is the degree at that level and l ranges from 0 to j . This leads to Algorithm 3.1.

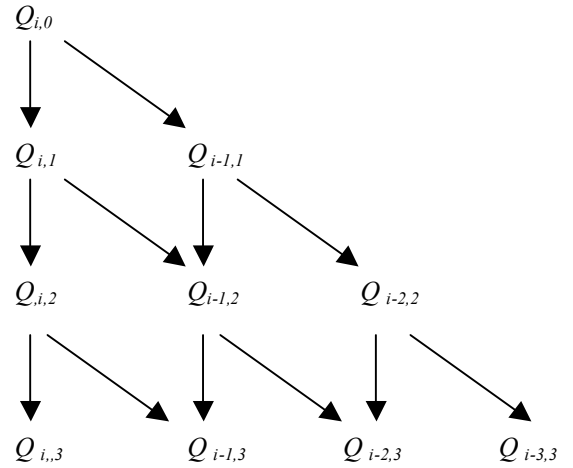


Figure 2. Interdependence of the values of the B-splines at a point t

In Fig. 2 each term is weighted sum of one or two terms in the line above it. The lines with arrows indicate the flow of the computation. Vertical lines denote multiplication by the first factor in Equation (3-3) and diagonals by the second factor in that equation.

Algorithm 3.1

Procedure BSPLINE(i, t, k):

Evaluation of all the B-splines at a point t belonging to the interval $[t_i, t_{i+1}]$.

Notation: k is the degree of the spline. t is the point where the splines are evaluated. The array $Q(I, J)$ contains the values $Q_{i,j}(t)$. a and b auxiliary variables.

Initalize $Q(i, 0)$ to 1.

For $j=l$ to k do:

Begin.

 For $l=0$ to j do:

Begin.

 {Compute $Q(i-l, j)$.}

$a = ((t-t_{i-l}) / (t_{i-l+j} - t_{i-l})) \times Q(i-l, j-1)$.

$b = ((t_{i-l+j+1} - t) / (t_{i-l+j+1} - t_{i-l+1})) \times Q(i-l+1, j-1)$.

$Q(i-l, j) = a + b$.

End.

End.

End of Algorithm.

We are given a set of data points r_1, r_2, \dots, r_j . We have to determine a set of control points that generates a B-spline curve for a set of given data points. If a data point lies on the B-spline curve, then it must satisfy Equation (3-1). Writing Equation (3-1) for each of j data points yields

$$r_1(t_1) = Q_{1,k}(t_1)C_1 + Q_{2,k}(t_1)C_2 + \dots + Q_{i,k}(t_1)C_i$$

$$r_2(t_2) = Q_{1,k}(t_2)C_1 + Q_{2,k}(t_2)C_2 + \dots + Q_{i,k}(t_2)C_i$$

.

.

.

$$r_j(t_j) = Q_{1,k}(t_j)C_1 + Q_{2,k}(t_j)C_2 + \dots + Q_{i,k}(t_j)C_i$$

where $2 \leq k \leq n+1 \leq j$. This system of equations is more compactly written in matrix form as

$$[r] = [Q][C] \quad (3-4)$$

where

$$[r]^T = [r_1(t_1) \ r_2(t_2) \ \dots \ r_j(t_j)]$$

$$[C]^T = [C_1 \ C_2 \ \dots \ C_i]$$

$$[Q] = \begin{bmatrix} Q_{1,k}(t_1) & Q_{2,k}(t_1) & \dots & Q_{i,k}(t_1) \\ Q_{1,k}(t_2) & Q_{2,k}(t_2) & \dots & Q_{i,k}(t_2) \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ Q_{1,k}(t_j) & Q_{2,k}(t_j) & \dots & Q_{i,k}(t_j) \end{bmatrix}$$

If $i = j$, then the matrix $[Q]$ is square and the set of control points is directly obtained by matrix inversion, i.e.,

$$[C] = [Q]^{-1}[r], \quad 2 \leq k \leq n+1 = j \quad (3-5)$$

If $i < j$, $[Q]$ is no longer square, this problem can be solved according to the rules of matrix. Recalling that a matrix times its transpose is always square. Thus, multiplying both sides of Equation (3-4) by $[Q]^T$ yields

$$[Q]^T[r] = [Q]^T[Q][C]$$

and

$$[C] = [[Q]^T[Q]]^{-1}[Q]^T[r] \quad (3-6)$$

In both cases, resulting B-spline curve passes through each data point, i.e., a curve fit is obtained. This technique assumes that the matrix $[Q]$ is known. Provided that the order of the B-spline basis k , the number of control points $n+1$, and the parameter value along the curve are known, then the basis functions $Q_{i,k}(t_j)$ and hence the matrix $[Q]$ can be obtained.

The parameter value t_j for each data point is a measure of the data point's distance along the B-spline curve. We can use the Chord length method for the estimation of t_j . For this method the chord length l of the curve is first computed according to

$$l = \sum_{j=1}^n \|r_j - r_{j-1}\| \quad (3-7)$$

t_j associated with the point r_j is computed as

$$t_j = t_{j-1} + \|r_j - r_{j-1}\| \frac{t_{\max}}{l}, \quad (3-8)$$

$j = 1, 2, 3, \dots, n+1$ where $t_1 = 0$ and $t_{\max} = n+1$.

When the sampled curve points on a B-spline are dense, the B-spline curve between two consecutive sampled curve points is well approximated by a line segment, and the distance between two points is well approximated by the chord length, which is a linear approximation to the B-spline between points.

4. MATCHING THE CURVE USING DISSIMILARITY BETWEEN THEIR CONTROL POINTS

We are presented with a set of sample curve data points from a transformed object curve, and are asked to recognize the test curve from the B-spline best fitted to the sample curve data points. Given two B-

splines $r(t)$ and $r'(t)$, let S and S' be their total arc length respectively. Assume that their centers of mass coincide with the origin of the coordinate system. The control points assignment of curve $r'(t)$ is as follows.

Let $\phi = (r_1, r_2, \dots, r_{n+1})$ be the control points set of curve $r(t)$ and let s_1, s_2, \dots, s_{n+1} be the distances in arc length travelled counterclockwise from the point of intersection of the curve $r(t)$ with the positive x-axis (from the starting point if the curve is open) to r_1, r_2, \dots, r_{n+1} respectively. Denote $\phi' = (r'_1, r'_2, \dots, r'_{n+1})$ as the assigned control points set for the curve $r'(t)$, and let $s'_1, s'_2, \dots, s'_{n+1}$ be the distances in arc length travelled counterclockwise from the point of intersection of the curve $r'(t)$ with the positive x-axis (from the starting point if the curve is open) to $r'_1, r'_2, \dots, r'_{n+1}$ respectively. The ϕ' set is determined such that :

$$\frac{s'_i}{S'} = \frac{s_i}{S}, \quad i = 0, 1, 2, \dots, n \quad (4-1)$$

This control points assignment guarantees that the ratio of the distances in arc length between the control points of $r'(t)$ relative to its total length is the same as that for $r(t)$.

The error (dissimilarity) E between $r(t)$ and $r'(t)$ is then defined as:

$$E = \frac{\sum_{i=0}^{i=n} \sqrt{\|r_i - r'_i\|^2}}{n} \quad (4-2)$$

Eqn.(4-2) should theoretically be zero when $r(t)$ and $r'(t)$ are the same, $r'(t)$ is recognized as sample curve for which the error (dissimilarity) E is the smallest. According to the curve matching algorithm the error (dissimilarity) E between the test object curve and sample curve is calculated. If the value of E is equal to zero we can say that the test curve and sample curve coincide. Otherwise the minimum value of E which is less than the threshold value is chosen as the best match.

5. CONCLUSION

This curve matching program can be used to recognize the curves and character recognition. In this study the experiment is made to test the curve matching program. According to the result of the experiment

this curve matching program can provide the recognizing the handwritten characters which are differed for different persons, different writing ways, different sizes, different directions and different stretching ways. In this paper, the curve matching program covers only the curve matching method involving uniform B-spline curves of degree 3 and limited control points. The uniform B-spline with equal chord length does not work well with the curve with the curvature change very fast, or in another words, the curve has more details. This can be done by using nonuniform B-spline with inverse chord lengths. If the test object curve and sample curve are not in the same orientation, or the test object curve is stretching in some direction the program developed also fails to recognize. This can be improved by determine the best affine between the test object curve and sample curve. The number of segments as well as degree of B-spline depends significantly to the (sample/test object) curves. However, in this study we have not covered yet. What is the best degree as well as the best number of segments, how to divide a curve to segments are very interesting.

REFERENCES

- [Car85a] Carlo, A.; and Cabriella, S. A width-independent fast thinning algorithm, IEEE Trans. Patt. Anal. Machine Intell, PAMI-7, no. 4, pp. 463-474, 1985.
- [Coh94a] Cohen, F.S.; and Wang, J.Y. Modeling image curves using invariant 3-D object curve model. IEEE Trans. 16: 1-12, 1994.
- [Coh95a] Cohen, F.S.; Huang, Z.; and Yang, Z. Invariant matching and identification of curves using B-spline curve representation. IEEE Trans. on Image Process. 4: 1-10, 1995.
- [deB72a] de Boor, C. On Calculation with B-splines, J. Approx. Theory 6: 50-62, 1972.
- [deB78a] de Boor, C. A Practical Guide to Splines, Springer-Verlag, New York, 1978.
- [Mur94a] Murray, J.D.; and van Ryper, W.. Encyclopedia of Graphics File Formats. O'Reilly & Associates, Sebastopol, CA, USA, 1994.
- [Pav82a] Pavlidis, T. Algorithms for Graphics and Image Processing. Computer Science Press, Rockville, MD, USA, 1982.
- [Rim93a] Rimmer, S. Windows Bitmapped Graphics. McGraw-Hill, New York, 1993.
- [Rog90a] Rogers, F.D.; and Adams, J.A. Mathematical Elements for Computer Graphics. McGraw-Hill, New York, 1990.
- [Yam88a] Yamaguchi, F. Curves and Surfaces in Computer Aided Geometric Design. Springer – Verlag, New York, 1988.