Curve Matching and Character Recognition by Using B-Spline Curves

H. Tirandaz, A. Nasrabadi, J. Haddadnia, *Member, IACSIT*

***Abstract*—This paper is devoted to presenting a novel algorithm for curve matching and character recognition. This algorithm is based on constructing and comparing B-spline curves of object boundaries. At first, the dominant points are calculated on the objects boundary by using Local Curvature Maximum (LCM), then, control points are obtained by using B-spline least square fitting technique. Curve matching and character recognition is done by comparing result B-spline curves. The result from proposed method shows good accuracy besides low computational complexity of proposed method for curve matching and character recognition.**

***Index Terms*—B-Spline curves, curve matching, character recognition, control points, parametric values.**

I. Introduction

There are many techniques for curve matching and character recognition. 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 [1]. Because of these properties they can be used for represent curves, and consequently they have been extensively used in computer-aided design and computer graphics [4,6], and object segmentation and tracking [7].

In this paper, curve matching and character recognition is done by using of B-spline curves for contour construction. At first, dominant points are calculated on the boundary of objects by using Local Curvature Maximum (LCM), and then, control points are obtained by using B-spline least square fitting technique. At the end, dissimilarity between sample and test character control points are calculated by least-square method and then determine most similar character, from sample characters data set, as the result.

The rest of this paper is organized as follows: In section 2, scaling and contour calculations and in section 3, details on the proposed method for calculations of dominant point are described. In section 4, calculations of control points based on least-square B-spline curves fitting method are described. Finally, In Section 5 proposed method for Dissimilarity calculations between curves method are described and experimental results are given to demonstrate the usefulness and quality of the approach. Steps of Our proposed method are explicitly presented in fig 1.

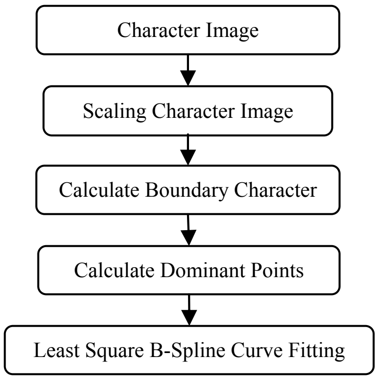


Fig. 1 B-spline curve matching steps

II. Scaling and Contour Calculation

At this stage, scaling is used to avoid from error created in calculation of B-spline curves and elucidating results. Then, we calculate boundary image points by using of gradient filter. The results are depicted in fig. 6.

III. Dominant Points

Since there is not need to use all data points for curve contour construction, besides some points properly selected from the given points play an important role in yielding better approximations;

Fitting curve problem by using of dominant points, thus, will present better approximation. The dominant points are obtained based on a method described in [2].

For the time being , consider that the B-spline curve fitting problem of character boundary is basically converted to select dominant points d_j $j= 0,\ldots,n$ from data points p_i $i= 0,\ldots,m$. Once dominant points are chosen, a B-spline curve can be generated with ease by performing the knot placement in Eq. (6) and the least-square curve fitting to all the points. Eventually, the quality of a resulting curve depends on how to select dominant points. Various kinds of feature points characterizing the shape of the given point set can be used for selecting dominant points. Compared to the others, the LCM points can be determined more easily and robustly. Since, In this work, LCM (Local Curvature Maximum) method that was described in [11], is used. From the curvatures k_i estimated at the given points, we would select n points, as LCM points, that they have greater curvature.

A. *Curvature*

Curvatures of the given points provide useful information

Manuscript received March 2011.

Hamed Tirandaz is with the Mechatronics Engineering Department, Sabzevar Tarbiat Moallem University, Sabzevar, Iran,(e-mail: hamedtirandaz@gmail.com.)

Abbas Nasrabadi is with the Mechatronics Engineering Department, Sabzevar Tarbiat Moallem University, Sabzevar, Iran,(e-mail: abbas.nasrabadi@gmail.com.)

Javad Haddadnia is with the Electrical Engineering Department, Sabzevar Tarbiat Moallem University, Sabzevar, Iran,(e-mail: haddadnia@sttu.ac.ir).

to describe the shape of the points [3], [6]. When a curve r(t) is given, the curvature k_i at the points p_i can easily obtain as follows [6]:

$$K_i = \frac{\| r'(\bar{t}_i) \times r''(\bar{t}_i) \|}{\| r'(\bar{t}_i) \|} \quad (1)$$

Where $r'(\bar{t}_i)$ and $r''(\bar{t}_i)$ are the first and the second derivatives of the curve r(t) at the parameter $\bar{t}_i$, respectively. Eq. (1) is used to estimate the initial curvatures of the given points for the base curve and to update the curvatures as the iteration proceeds. The result shown at fig. 3 and 4.

IV. Control Points of B-Spline Curves

We have assumed that the reader is familiar with the concepts of B-spline curves [3], [6], [8].Let r(t) be the position vector along the curve as a function of the parameter t, B-spline curve of this function can be represented as follow [3]:

$$r(t) = \sum_{i=0}^{n} B_{i,p}(t) C_j \qquad t_{p-1} \le t \le t_{n+1} \quad (2)$$

where C_j , j = 0, ... , n are the position of n+1 control points and $\{B_{i,p} \mid i=0,\ldots,n\}$ are the normalized B-spline basis functions of order p. The knot vector $T=\{t_0,t_1,\ldots,t_{n+p},t_{n+p+1}\}$ consists of non-decreasing real-valued knots and mostly its first and last knots are repeated with multiplicity equal to order p as follows:

$$t_0 = \cdots = t_{p-1} = 0 \quad , \quad t_{n+1} = \cdots = t_{n+p} = 1 \quad (3)$$

A. Least-Square B-Spline Curve Fitting Method

As a widely used tool for computing a B-spline curve r(t), least-squares B-spline curve fitting problem from data points p_i , as shown in fig. 2 takes three steps: parameterization, knot placement, and least-squares minimization. In parameterization, we select the parametric values $\bar{u}_i$ of the points p_i . In knot placement, we determine a knot vector T after specifying the order p and the number of points n. In least-square minimization, we determine the control points C_j (j = 0, . . . , n) of a B-spline curve C(t). this process is described as follows:

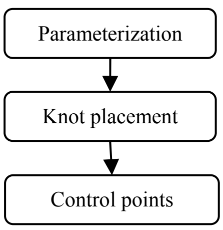


Fig. 2 Least-Square B-Spline Curve Fitting Method

1) Parameterization

In parameterization, we select the parametric values $\bar{u}_i$, from the dominant points d_i , that have described in section 3. Mostly the parameter values are computed using the chord length or centripetal methods [1], [2], Accordingly, we use the chord length for computing parameter values. The chord length technique is defined as follows:

$$u_0 = 0 , u_n = 1 \text{ and Then}$$
$$\bar{u}_k = \bar{u}_{k-1} + \frac{| P_k - P_{k-1} |}{d} \qquad \mathbf{k} = 1,\cdots,n-1 \quad (4)$$

Where d_i , the approximated length of the curve is:

$$d = \sum_{k=1}^{n} | P_k - P_{k-1} | \quad (5)$$

2) Knot Placement

In knot placement after specifying the order p, the number of dominant points n, the parameter values $\bar{u}_i$ and the number n of control points, we determine a knot vector T. Mostly the knots are determined to reflect the distribution of the parameter values and to guarantee that every knot span contains at least one parametric value. The interior knots t_i of the knot vector T can be spaced as follows [3]:

$$u_0 = \ldots = u_p = 0 \qquad u_{n+1} = \ldots = u_{n+p+1} = 1$$
$$u_{j+p} = \frac{j}{n-p+1} \qquad j = 1,\cdots,n-p \quad (6)$$

Where n+p+1 is the size of knot vector, p is degree of curve and n+1 is the number of control points .

For the time being, we can compute control point values by using least-square B-spline curve fitting by using Eq. (6), as the follow.

3) Least-Square Fitting Method

In least-square minimization process, we determine the control points C_j , j=0,…, n of a B-spline curve r(t) by minimizing the least-squares error defined as[3], [4]:

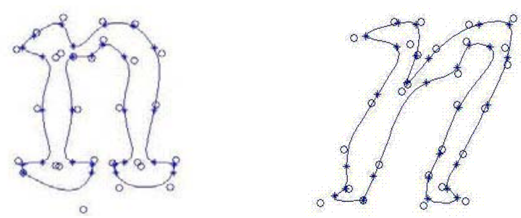
$$E(C_1,\cdots,C_n) = \sum_{i=1}^{m} \| r(\bar{u}_i) - C_i \| \quad (7)$$

Where m is the number of data points.

This method is described in [2], in details.

V. Dissimilarity Between Curves

So far, we describe how to compute B-spline curves for a set of data points from sample and test characters. For example, B-spline curve of the sample and test character are been shown in Fig. 3. For the time being, we will compute dissimilarity between each sample character and test character and then find the best matched character among the samples character to the test character. Therefore, we let r(t) and r´(t) to be the B-spline curves of test and sample character. In this section, we will calculate dissimilarity between test character and all sample character in our codebook, and then select a character that has most matching degree with test character as the matching process result.



(a) Sample character (b) Test character

Fig. 1 comparison between the B-spline curves of test and sample character.

Let $Q = (q_1, q_2, \dots, q_{n+1})$ be the control vector of test curve r(t) and $Q' = (q'_1, q'_2 \dots, q'_{n+1})$ be the control vector of sample curve r′(t), and let l be an index that minimizes distances between two control vectors. Then, similarity between two control vector is determined as:

$$\min_{0 \le l \le n} \sum_{j=1}^{n} \| q_j - q'_{(j+l) \bmod n} \| \tag{8}$$

At the time being, error rate E between the test curve r(t) and a sample curve r′(t), is defined as:

$$E = \sum_{t} \frac{\| r(t) - r'(t) \|}{m} \tag{9}$$

Such that:

$$\| r(t) - r'(t) \|$$

$$= \sum_{i=0}^{n} | B_{i,p}(t) C_i - B_{i+l \bmod n, p}(t - t_l) C_{i+l \bmod n} \tag{10}$$

Where m is the number of B-spline curve points and t_l is distance between two B-spline curve.

In this paper, we have considered the number of control points equal to n+1 as the number of dominant points selected from m data points.

The best character matching from sample codebook with the test character, then, can be determined with iterating this comparison process for all sample characters against the test character.

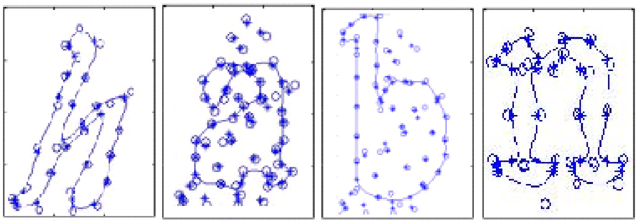


Fig. 4- comparison between the B-spline curves.

VI. Conclusion

B-splines stand as one of the most efficient curve representations. A method for solving curve matching problem by using of B-spline curves is presented, in this paper. Proposed method can effectively detect characters with different sizes, formats, especially can effectively applied to recognize handwritten. Some advantages of dissimilarity comparison between test character and sample characters have obtained from the proposed method are shown in Table 1. Corresponding B-spline curves of their character boundary images are shown in fig. 5. As the result are shown in Table 1, proposed method can effectively recognize characters.

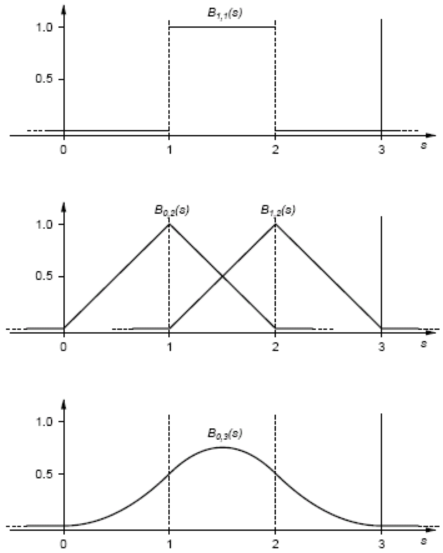


Fig. 5 - A spline basis function $B_{n,d}$ of order d is built up recursively from basis functions of lower order.

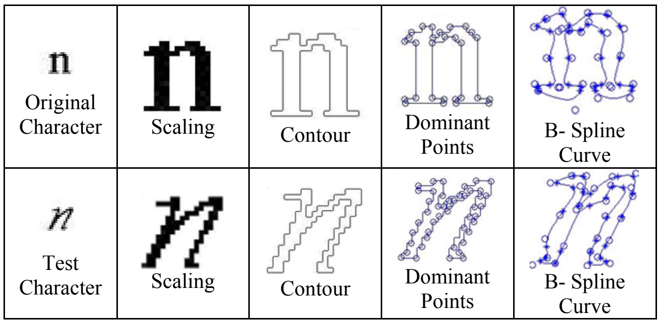


Fig. 6 steps of proposed Algorithm

TABLE 1 Result of Comparison Between Test Character H and Sample Characters (A, B, N, H, D)

Text / Sample	a	b	c	d	e	f	g	h	k	L	m	n
a	**0.894**	1.252	1.014	1.607	1.476	2.012	1.115	1.341	1.917	1.413	1.682	1.117
b	1.178	**1.005**	1.101	1.128	1.392	1.607	1.952	1.150	0.931	1.018	1.227	0.804
c	1.154	1.901	**0.452**	1.234	0.931	1.763	1.416	1.544	1.481	1.449	1.604	1.737
d	1.583	1.154	1.005	1.121	0.981	1.357	1.505	1.171	1.456	1.132	1.739	1.457
e	1.127	1.529	0.394	1.670	**0.516**	2.112	2.192	1.803	2.005	1.913	1.346	1.178
f	1.870	1.165	1.282	1.175	1.647	**1.175**	1.692	1.564	1.341	1.159	2.325	2.070
g	0.824	1.307	1.156	1.663	1.173	1.874	**1.026**	1.550	1.712	1.789	1.221	1.761
h	1.178	1.337	1.442	1.153	2.176	2.201	2.475	**1.099**	1.180	1.148	1.294	**0.283**
k	2.127	0.828	2.011	1.933	1.464	1.724	2.296	1.159	**0.715**	1.098	1.443	1.217
l	1.758	1.139	1.306	**0.988**	1.412	1.290	2.100	1.117	0.808	**0.342**	1.470	1.127
m	1.237	1.455	1.168	1.413	1.742	2.478	2.203	1.206	1.270	1.362	**0.766**	1.104
n	0.950	1.141	1.963	1.632	1.183	1.802	1.351	1.219	1.054	1.177	0.920	0.780

Appendix

B-Spline Curves

Letting *r(t)* be the position vector along the curve as a function of the parameter *t*, a B-spline curve is given by

$$r(t) = \sum_{i=0}^{n} B_{i,k}(t) C_i \tag{11}$$

where $a \le t \le b$, the C_i are the position vectors of the n+1 control polygon vertices, that they can be calculated from least-square algorithm introduced in Eq. 7 and $B_{i,k}$ are normalized basis functions for a spline of order *k* that they can be built up recursively from basis functions of lower order as follows:

$B_{i,k}$

$B_{i,k-1}$ $B_{i+1,k-1}$

$B_{i,k-2}$ $B_{i+1,k-2}$ $B_{i+2,k-2}$

. .

. . .

$B_{i,1}$ $B_{i+1,1}$ $B_{i+2,1}$ $B_{i+3,1}$ . . . $B_{i+k-1,1}$

Let $B_{i,k}$, be the *i*th basis function for a spline of order *k*. Then for a spline with non-uniform knots spacing, the following recursive rule applies:

Ground instance

$$B_{n,1}(s) = \begin{cases} 1 & if\ n \le s < n+1 \\ 0 & otherwise \end{cases} \tag{12}$$

Inductive step

$$B_{n,d}(S) = \frac{(s - k_n)B_{n,d-1}(s)}{k_{n+d-1} - k_n} + \frac{(k_{n+d} - s)B_{n+1,d-1}(s)}{k_{n+d} - k_{n+1}}. \tag{13}$$

These curves are described in references[3], [4], [6], in details.

REFERENCES

[1] 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.

[2] Hyung Park, Joo Haeng Lee, "B-spline curve fitting based on adaptive curve refinement using dominant points" Computer-Aided Design, Elsevier Lts,2007

[3] Piegl, L., and Tiller, W., *The NURBS Book,* 2nd ed., Springer, Berlin, 1997

[4] Davie. F. Rogers, "An Introduction to NURBS With Historical Perspective" ,2001.

[5] Cohen, E., Some mathematical tools for a modeler's bench, *IEEE Comput. Graph. and Appl.,* Vol. 3, No. 7, pp. 63-66, October 1983.

[6] Blake, A., Isard, M., 1998. Active Contours. Springer, London, England

[7] Sukmarg, O., Rao, K. R., B-Spline Curve Representation of Segmented Object in MPEG Compressed Domain, 2000

[8] De Boor C. A practical guide to splines. Berlin: Springer; 1978.

[9] Park H. An error-bounded approximate method for representing planar curves in B-splines. Computer Aided Geometric Design 2004;21(5):479–97.

[10] Cohen, Huang, Yang, Invariant matching and identification of curves using B-spline curve representation, IEEE Trans , 1995

[11] Razdan A. Knot placement for B-spline curve approximation. Report 1999. Arizona State University. http://citeseer.ist.psu.edu/398077.html.

Hamed Tirandaz received the B.Eng. Degree in Applied Mathematics from the University of Sabzevar Tarbiat Moallem University, Sabzevar, Iran, in 2006, and M.Sc. degree in Mechatronics Engineering from Semnan University, Semnan, Iran, in 2009.

His research interests include mainly deformable models applied to segmentation and tracking, OCR, computer vision image and video processing. He has published several papers in the above area. Email: hamedtirandaz@gmail.com

Abbas Nasrabadi was born in Sabzevar, Iran. He received M.Sc. in Mechatronics Engineering in January 2008 from Semnan University, Semnan, Iran. He works in Mechatronics Engineering Department, Computer and Electrical Engineering Faculty, in Sabzevar Tarbiat Moallem University, Sabzevar, Iran. He has published several papers in the above area of image processing. He is a member of the IACSIT. Email: a.nasrabadi@stmu.ac.ir. Personal Homepage: http://profsite.sttu.ac.ir/nasrabadi

Javad Hddadnia Javad Haddadnia received his M.Sc. degree in Electrical and Electronic Engineering with the first rank in 1995 and PhD in Electronic Engineering in 2002, from Amirkabir University of Technology, Tehran, Iran, respectively. He is currently an associate professor in the Electrical and Computer Engineering Department of Sabzevar Tarbiat Moallem University, Sabzevar, Iran. His research interests include digital image processing, computer vision, neural network and face detection and recognition. He has published several papers in the above area of image processing. He is a member of SPIE and CIPPR. Email: haddadnia@sttu.ac.ir