# A NOVEL 2D SHAPE MATCHING ALGORITHM BASED ON B-SPLINE MODELING

*Yue Wang and Eam Khwang Teoh*

School of Electrical and Electronic Engineering, Nanyang Technological University
Nanyang Avenue, Singapore 639798
Email: s2633175g@ntu.edu.sg, eekteoh@ntu.edu.sg

## ABSTRACT

This paper presents a novel algorithm for 2D planar curve recognition based on B-Spline modeling. It combines the advantages of the B-Spline that are continuous curve representation and affine invariant, and the robustness of the CSS matching with respect to noise and affine transformation. It solves the problem of the non-uniqueness of B-Spline in curve matching. A new algorithm, which contains the degree increasing and reduction, is proposed for smoothing B-Spline and constructing the CSS image. The proposed algorithm has been tested and good performance has been obtained.

## 1. INTRODUCTION

The object matching is the ultimate purpose in many application domains of image processing and computer vision. Many methods have been developed for object matching. These methods can be further classified as boundary, region and model-based methods [1].

A direct approach is to match object boundary curves extracted from images. A number of approaches have been proposed for curve modeling, such as Fourier descriptor, parametric algebraic curve, shape axis tree, shape space and B-Spline [2]-[7]. The B-Splines stand as one of the most efficient curve representation and possess very attractive properties such as compactness and continuity, local shape controllability, and invariance to affine transformations. However, there is very little work has been devoted to B-Spline for curve matching (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 process difficult when comparing the respective control points of the curves to be matched. Current existing methods for B-Spline curve matching are based on either the calculation of sum error of the corner points [2], the discrete re-sampled points [3] on the B-Spline curve, or a re-construction of B-Spline with a fix number of control points [4]. These methods are noise sensitive and do not employ the property of continuous curve description of the B-Spline.

The Curvature Scale Space (CSS) image was introduced in [10] as a shape representation for digital planar curves. The representation is computed by convolving the curve with a Gaussian function at different levels of scale, and extracting the location of inflection points of the resulting curves for matching. However, it suffered from the noise sensitivity due to curvature computation on the digital contour.

This paper is an attempt to find a novel matching solution to overcome this limitation and combine the robustness of B-Spline and CSS matching. The full proposed algorithm is given in Section 2. Section 3 shows some experimental results and this paper is concluded in Section 4.

## 2. B-SPLINE CURVE MATCHING ALGORITHM

The main goal of our algorithm is to combine the advantages of the B-Spline that are continuous curve representation and affine invariant, and the robustness of the CSS matching with respect to noise and affine transformation. In addition, it should also avoid the conventional CSS matching that has to be using re-sampled points on the curve, thus reduces the error.

Our method consists of two steps: (i) Smooth B-Spline curve and construct the CSS image; (ii) Extract the maxima of CSS image and perform matching. The details are discussed in the following Sections. This method has three mainly novelties:

1. It relies on the whole continuous B-Spline curve rather than the discrete re-sampled points used in almost all other methods that would introduce error and lead to imprecise results.
2. The CSS has been proved to be a good method for shape matching. More importantly, the B-Spline curve is particularly easy and accurate in calculating the curvature of the curve and generating the CSS image due to its equation expression.

0-7803-8554-3/04/$20.00 ©2004 IEEE.

3. Our algorithm can be used for affine invariant matching without estimation of affine transformation.

### 2.1 The Algorithm for B-Spline Curve Smoothing

In order to generate the CSS image, the input shape must be smoothed. We present a new smoothing scheme: the B-Spline curve is smoothed first by increasing the degree, and then a Least Square Error method is used to approach the smoothed B-Spline curve with the same degree as the original B-Spline curve. As the smoothed B-Spline curve is represented by the same degree of B-Spline, it allows straightforward computation of geometric properties of the smoothed curve, such as affine length and curvature. Therefore, it is particular suitable for forming the CSS image of B-Spline curve.

#### *2.1.1 Increase of Degree for Smoothing of B-Spline Curve*

It is well known that for the same set of control points $\{G_0, G_1, ..., G_n\}$, increasing the degree of the B-Spline will result at the smoother curve it may form [8]. Moreover, since B-Splines are affine invariant, this means that the B-Spline representations of two sets of control points related by an affine transformation, are related by the same affine motion.

#### *2.1.2 Reduction of Degree for Smoothed B-Spline Curve*

Assume for a B-spline $g_i(s)$ of degree $k$, assume its control points are $Q = \{Q_0, Q_1, ..., Q_n\}$, to reduce a B-Spline $h_i(s)$ the degree to $k$-1, suppose its control points are $D = \{D_0, D_1, ..., D_n\}$. The objective is to minimize,

$$f = \sum_{i=0}^{n} \int_0^1 \left(g_i(s) - h_i(s)\right)^2 ds \qquad (1)$$

In order to get the control point $D$ of B-Spline $h_i(s)$, we should get partial derivatives of $D$ from equation (1), that is:

$$\frac{\partial f}{\partial D_i} = \sum_{i=0}^{n} \int_0^1 \frac{\partial}{\partial D_i} \left(g_i(s) - h_i(s)\right)^2 ds \qquad (2)$$

By solving the equation (2), the control points $D$ of lower degree B-Spline can be obtained. Here we consider the case of approaching the B-Spline with degree 5 by B-Spline of degree 4. First we assume for the B-Spline with degree 5, we have:

$$g_i(s) = \begin{bmatrix} s^4 & s^3 & s^2 & s & 1 \end{bmatrix} M_5 \begin{bmatrix} Q_i \\ Q_{(i+1)\bmod(n+1)} \\ Q_{(i+2)\bmod(n+1)} \\ Q_{(i+3)\bmod(n+1)} \\ Q_{(i+4)\bmod(n+1)} \end{bmatrix}, i = 0,1,2,...,n. \qquad (3)$$

where

$$M_5 = \frac{1}{24} \begin{bmatrix} 1 & -4 & 6 & -4 & 1 \\ -4 & 12 & -12 & 4 & 0 \\ 6 & -6 & -6 & 6 & 0 \\ -4 & -12 & 12 & 4 & 0 \\ 1 & 11 & 11 & 1 & 0 \end{bmatrix} \qquad (4)$$

For the B-Spline with degree 4, we have

$$h_i(s) = \begin{bmatrix} s^3 & s^2 & s & 1 \end{bmatrix} M_4 \begin{bmatrix} D_i \\ D_{(i+1)\bmod(n+1)} \\ D_{(i+2)\bmod(n+1)} \\ D_{(i+2)\bmod(n+1)} \end{bmatrix}, i = 0,1,2,...,n. \qquad (5)$$

where

$$M_4 = \frac{1}{6} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} \qquad (6)$$

the equation (2) can be converted to below equations [9]:

$$B \cdot Q^T = A \cdot D^T \qquad (7)$$

where

$$B = \begin{bmatrix}
\frac{15619}{20160} & \frac{15619}{20160} & \frac{477}{2249} & \frac{247}{20160} & \frac{1}{20160} & 0 & 0 & \cdots & \cdots & 0 & 0 & \frac{1}{20160} & \frac{247}{20160} & \frac{477}{2249} \\
\frac{477}{2249} & \frac{15619}{20160} & \frac{15619}{20160} & \frac{477}{2249} & \frac{247}{20160} & \frac{1}{20160} & 0 & 0 & \cdots & \cdots & 0 & 0 & \frac{1}{20160} & \frac{247}{20160} \\
\frac{247}{20160} & \frac{477}{2249} & \frac{15619}{20160} & \frac{15619}{20160} & \frac{477}{2249} & \frac{247}{20160} & \frac{1}{20160} & 0 & 0 & \cdots & \cdots & 0 & 0 & \frac{1}{20160} \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
\cdots & 0 & \frac{1}{20160} & \frac{247}{20160} & \frac{477}{2249} & \frac{15619}{20160} & \frac{15619}{20160} & \frac{477}{2249} & \frac{247}{20160} & \frac{1}{20160} & 0 & \cdots & \cdots & \cdots \\
\cdots & 0 & 0 & \frac{1}{20160} & \frac{247}{20160} & \frac{477}{2249} & \frac{15619}{20160} & \frac{15619}{20160} & \frac{477}{2249} & \frac{247}{20160} & \frac{1}{20160} & 0 & \cdots & \cdots \\
\cdots & \cdots & 0 & 0 & \frac{1}{20160} & \frac{247}{20160} & \frac{477}{2249} & \frac{15619}{20160} & \frac{15619}{20160} & \frac{477}{2249} & \frac{247}{20160} & \frac{1}{20160} & 0 & \cdots \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
\frac{247}{20160} & \frac{1}{20160} & 0 & 0 & \cdots & \cdots & 0 & 0 & \frac{1}{20160} & \frac{247}{20160} & \frac{477}{2249} & \frac{15619}{20160} & \frac{15619}{20160} & \frac{477}{2249} \\
\frac{477}{2249} & \frac{247}{20160} & \frac{1}{20160} & 0 & 0 & \cdots & \cdots & 0 & 0 & \frac{1}{20160} & \frac{247}{20160} & \frac{477}{2249} & \frac{15619}{20160} & \frac{15619}{20160} \\
\frac{15619}{20160} & \frac{477}{2249} & \frac{247}{20160} & \frac{1}{20160} & 0 & 0 & \cdots & \cdots & 0 & 0 & \frac{1}{20160} & \frac{247}{20160} & \frac{477}{2249} & \frac{15619}{20160}
\end{bmatrix}$$

$$A = \begin{bmatrix}
\frac{302}{315} & \frac{397}{840} & \frac{1}{21} & \frac{1}{2520} & 0 & 0 & \cdots & \cdots & 0 & 0 & \frac{1}{2520} & \frac{1}{21} & \frac{397}{840} \\
\frac{397}{840} & \frac{302}{315} & \frac{397}{840} & \frac{1}{21} & \frac{1}{2520} & 0 & 0 & \cdots & \cdots & 0 & 0 & \frac{1}{2520} & \frac{1}{21} \\
\frac{1}{21} & \frac{397}{840} & \frac{302}{315} & \frac{397}{840} & \frac{1}{21} & \frac{1}{2520} & 0 & 0 & \cdots & \cdots & 0 & 0 & \frac{1}{2520} \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
\cdots & 0 & \frac{1}{2520} & \frac{1}{21} & \frac{397}{840} & \frac{302}{315} & \frac{397}{840} & \frac{1}{21} & \frac{1}{2520} & 0 & 0 & \cdots & \cdots \\
\cdots & 0 & 0 & \frac{1}{2520} & \frac{1}{21} & \frac{397}{840} & \frac{302}{315} & \frac{397}{840} & \frac{1}{21} & \frac{1}{2520} & 0 & 0 & \cdots \\
\cdots & \cdots & 0 & 0 & \frac{1}{2520} & \frac{1}{21} & \frac{397}{840} & \frac{302}{315} & \frac{397}{840} & \frac{1}{21} & \frac{1}{2520} & 0 & \cdots \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
\frac{1}{2520} & 0 & 0 & \cdots & \cdots & 0 & 0 & \frac{1}{2520} & \frac{1}{21} & \frac{397}{840} & \frac{302}{315} & \frac{397}{840} & \frac{1}{21} \\
\frac{1}{21} & \frac{1}{2520} & 0 & 0 & \cdots & \cdots & 0 & 0 & \frac{1}{2520} & \frac{1}{21} & \frac{397}{840} & \frac{302}{315} & \frac{397}{840} \\
\frac{397}{840} & \frac{1}{21} & \frac{1}{2520} & 0 & 0 & \cdots & \cdots & 0 & 0 & \frac{1}{2520} & \frac{1}{21} & \frac{397}{840} & \frac{302}{315}
\end{bmatrix}$$

Therefore the control points D of degree reduced B-Spline are:

$$D = A^{-1} B \cdot Q \qquad (8)$$

Figure 1 and Figure 2(a) show two examples of smoothing B-Spline based on our proposed method.

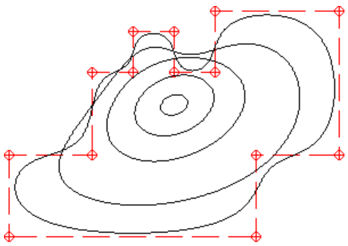

**Figure 1** An example on smoothing the B-Spline curve based on the proposed method. ⊕ indicates the initial control points.

### 2.2 CSS Image Construction

The CSS image of a planar shape curve represents the location of inflection point on its normalized affine arc-length. It can achieve an affine invariant parameterization. The normalized affine arc-length can be defined as:

$$L(s)=\frac{\int_0^s\left|\dot g(s)\times\ddot g(s)\right|^{\frac{1}{3}}ds}{\int_0^1\left|\dot g(s)\times\ddot g(s)\right|^{\frac{1}{3}}ds} \tag{9}$$

In the conventional CSS image construction, the vertical of CSS image is defined by the width $\sigma$ of a 1-D Gaussian function. Obviously, in our method, this cannot be used. A new definition using the ratio of the affine arc-lengths of original shape and current smoothed shape has been developed as the vertical of CSS image [9]:

$$Y(m)=1-\frac{\sum_{i=0}^{n}\int_0^s\left|\dot g_i^m(s)\times\ddot g_i^m(s)\right|^{\frac{1}{3}}ds}{L_{original}} \tag{10}$$

where $L_{original}$ is the affine arc-length of the original shape:

$$L_{original}=\sum_{i=0}^{n}\int_0^s\left|\dot g_i(s)\times\ddot g_i(s)\right|^{\frac{1}{3}}ds \tag{11}$$

If we keep smoothing any shape, it will result in the shrink to its centroid. Therefore, the definition of (10) will also normalize the vertical of CSS image to [0,1). The locations of inflection points of every $Y$ on the normalized affine length of whole smoothed curve during B-Spline smoothing procedure can be displayed in the plane $(L,Y)$, where $L$ is the normalized affine length of current smoothed curve and is from 0 to 1. $Y$ is defined in equation (10). One example on construction of CSS image that follows above definition is shown in Figure 2.

### 2.3 Extraction of Maxima of CSS Contour

The locations of CSS contour maxima are extracted as the shape descriptors for CSS curve matching. Note that usually the small contours of the CSS image represent the noise or small ripples of the input curve. If a maximum is less than 0.3 of the largest maximum of the same CSS image, it will be removed from matching. As a result, only significant concavities and convexities of a shape will contribute to the representation and matching. In Figure 2(b), the '+' marks denote the extracted maxima.

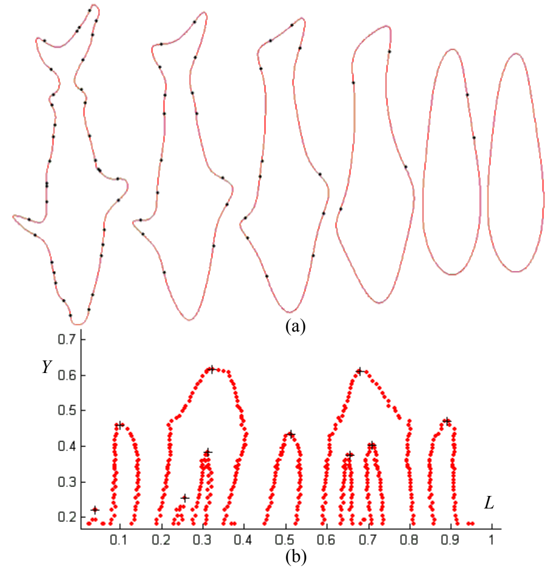


**Figure 2** CSS image construction for a fish shape. (a) the smoothing procedures for B-Spline curve by using proposed method, the first one is the original shape ; (b) shows the constructed CSS image.

### 2.4 Curvature Scale Space Matching

The complete matching algorithm which compares the two sets of maxima, one from the input image and the other from the model, is mainly followed the conventional CSS matching algorithm [11]. The algorithm first finds any possible changes in orientation which may have occurred in one of the two shapes. A circular horizontal shift then is applied to one of the two sets of CSS maxima to compensate for the effects of change in orientation. The summation of the Euclidean distances between the relevant pairs of maxima is then defined to be the matching value between the two CSS images. The mirror of the input shape is also to be performed the matching. Using the maxima of input shape we can easily calculate a new set of maxima which belongs to the CSS image of the mirror image of the input. We can then repeat matching algorithm for the new set and consider the lowest matching cost between the two.

## 3. EXPERIMENTAL RESULTS

The object matching algorithm presented in this paper has been simulated and tested on various images. A database, which contains 1100 shapes of marine creatures, has been set up for this experiment. The maxima of CSS image of every shape have been extracted as well. The algorithm presented above is used to find the similar shapes of the input shapes from this prototype database. We first verify the affine invariant of CSS image based on our method.

Figure 3(a) is the affine transformed shape of the original shape shown in Figure 2(a), its CSS image is shown in Figure 3(b). Figure 3(c) shows the maxima of both CSS images, they are overlapped quite well. Quantitative comparison[9] confirms the affine invariant of our method.

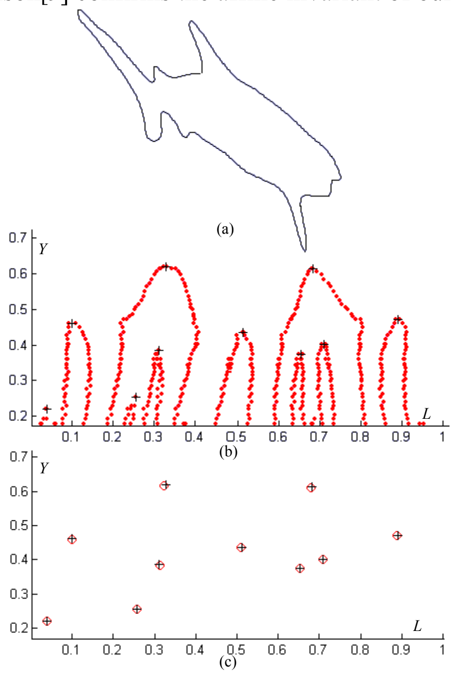


**Figure 3** (a) Affine transformed shape of Figure 2 (a); (b) CSS image; (c) Overlap of maxima of (b) and Figure 2(b).

Some of the matching results are shown in Figure 4. The input shapes are in the first row of three columns. The similar contours that have been matched by the proposed algorithm are shown in the rest columns. Please note that some of the matching results are different in rotation and scale of input shape, or are the mirror-of input shape.

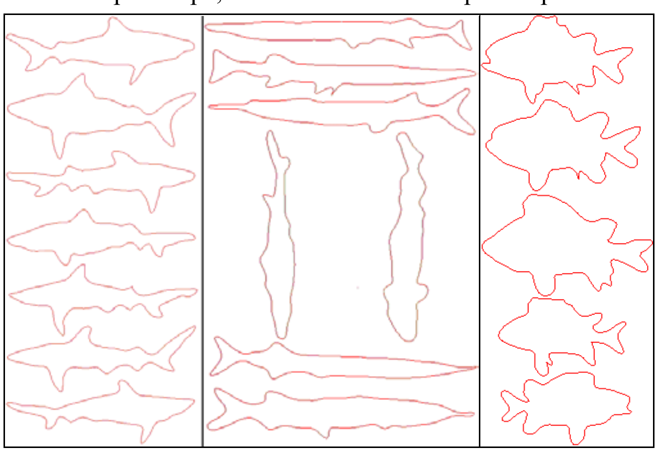

**Figure 4** More matching results.

## 4. CONCLUSION

A two-step object matching algorithm using B-Spline modeling is presented. A new method for smoothing B-Spline and construction of CSS image is presented. The proposed method is based on the continuous B-Spline representation for describing the whole curve rather than the discrete re-sampled points of the curve, thus avoids the errors and achieves a good accuracy. Furthermore, the proposed algorithm can be used for affine invariant matching without estimation of affine transformation. The proposed method has been tested for finding similar shapes from a prototype of marine animals with a vast variety of shapes. The experimental results show the good performance of the proposed algorithm.

## REFERENCES


[1] L. Torres and M. Kunt, *Video Coding: The 2nd Generation Approach*, Kluwer Academic Publisher, Dordrecht, 1996.

[2] Y. Gu and T. Tjahjadi, "Coarse-to-Fine Planar Object Identification Using Invariant Curve Features and B-Spline Modeling", *Pattern Recognition*, vol: 33, pp. 1411-1422, 2000.

[3] F. S. Cohen and J. Wang, "Modeling Image Curves Using Invariant 3D Object Curve Models - A Path to 3D Recognition and Shape Estimation from Image Contours", *IEEE Transactions on PAMI*, vol. 16, pp. 1-23, 1994.

[4] F. S. Cohen, Z. Huang and Z. Yang, "Invariant Matching and Identification of Curves Using B-Splines Curve Representation", *IEEE Transactions on Image Processing, v*ol. 4, no. 1, pp. 1-10, January 1995.

[5] Z. Huang and F. S. Cohen, "Affine-Invariant B-Spline Moments for Curve Matching", *IEEE Transactions on PAMI,* vol. 5, no. 10, October 1996.

[6] Y. Wang, E. K. Teoh and D. Shen, "Structure-Adaptive B-Snake for Segmenting Complex Objects", *ICIP 2001*, pp. 769-772, October 2001.

[7] Y. Wang, E. K. Teoh and D. Shen, "Lane Detection and Tracking Using B-Snake", *Image and Vision Computing*, vol. 22, Issue 4, pp. 269-280, April 2004.

[8] R. H. Bartels, J. C. Beatty, and B. A. Barsky, *An Introduction to Splines for Use in Computer Graphics and Geometric Modeling*. Morgan Kaufmann: Los Altos, CA, 1987.

[9] Y. Wang, *Object Segmentation and Matching Using B-Spline Model*, Ph.D Thesis submitted to Nanyang Technological University, Singapore, May 2004.

[10] F. Mokhtarian and A. K. Mackworth, "Scale-Based Description and Recognition of Planar Curves and Two-dimensional Shapes", *IEEE Trans. PAMI.*, vol. PAMI-8, pp. 3443, 1986.

[11] F. Mokhtarian, S. Abbasi and J. Kittler, "Robust and Efficient Shape Indexing Through Curvature Scale Space", *BMVC'96*, vol. 1, pp. 53-62, 1996.