Image Shape Representation Using Curve Fitting

PORNCHAI MONGKOLNAM[a], THANEE DECHSAKULTHORN, and
CHAKARIDA NUKOOLKIT[a]
[a]School of Information Technology
[a]King Mongkut's University of Technology Thonburi
Bangkok, THAILAND

Abstract: - We present an approach for representing digital image by using B-spline curve fittings to the segmented digital image. The image is first segmented to form isolated contours and regions. Each region's boundary (contour) is thereafter approximated by a B-spline curve. A cubic B-spline curve is used instead of a far higher degree Bezier curve to approximate the boundary because it has a local control property and is less wiggly. The obtained B-spline curves are collectively composed and used to represent the original image. One of the many practical applications lies in image retrieval, clustering and classification where general shapes, arrangement and composition of control points of those obtained B-spline curves are used to compare and find similarity between images in a collection.

Key-Words: - Image shape representation, curve fitting, curve approximation, B-spline, image segmentation, and contour-based image retrieval

1 Introduction

There are several existing work that use B-spline curves to represent shape of objects. A profile of 3D object is approximated by the curve and is used for curve matching in a data retrieval application. The object matching is an integral part for many applications of image processing and computer vision. The B-spline curves are widely used for curve representation of the object contours or boundaries because they possess some attractive properties such as smoothness, compactness, local shape controllability, and affine transformation invariance.

However, to the best of our knowledge few have applied the B-spline representation to a digital image for both image structural composition and comparison. Therefore, our work focuses on using the B-spline curves to represent structural shapes or contents of the image. By knowing the shapes, components and their relationships of the image, a good comparison could be made against other images, resulting in some interesting researches in various fields such as image retrieval, clustering, classification, and shape compression for a fast retrieval and display. A good overview of using shapes for image retrieval can be found in [9, 11].

This paper is organized as follows. First, some related work that are closely linked to ours on image shapes, B-splines and their applications are presented. Next, our work and its results are provided in the following section. It begins with image segmentation to get image contours and ends with B-spline basics and its approximation to those contours. Last, a conclusion and future work is given.

2 Related Work

Some prior work in 2D shape matching and curve representation has inspired our work in using B-spline curves to approximate boundaries of segmented image. Wang et al. [10] present a novel 2D shape-matching algorithm based on B-spline modeling. The algorithm is based on the B-spline curves and the Curvature Scale Space (CSS) image. The CSS image introduced by Mokhtarian et al. [5] is robust with respect to noise and affine transformation. It is used as a shape representation for a digital planar curve. The representation is computed by convolving the curve with a Gaussian function at different scaling levels. The CSS is suitable because the B-splines have advantages of being continuous curve representation and affine invariant. The algorithm first smoothens the B-spline curve of an input shape and constructs the CSS image. It then extracts the maxima of CSS image and performs matching.

Cohen et al. [1] use B-spline curves for matching 2D objects such as aircrafts and handwriting. Due to the B-splines' attractive properties, they choose the B-splines for curve modeling over other approaches such as the Fourier descriptors, the chain code, the polygonal approximation, the curvature primal sketch, the medial axis transform, the autoregressive models, the moments, the parametric algebraic curves, the curvature invariant, the stochastic transformation, the implicit polynomial functions, the bounded polynomials, and the reaction diffusion. The goal of their algorithm is to match and classify planar curves that are modeled as B-splines, independent of any affine transformations. Two methods are presented. First, the control points of the prototype curves are globally related to the knot points of a given sample curve and are compared. Second, a sum of the residual error between each prototype curve and the given sample curve is compared.

Vailaya et al. [8] propose a new image database retrieval method based on shape information. The following two features are used to represent shape of an image: a histogram of the edge directions and the invariant moments. Euclidean distance between the edge direction histograms is used as a matching score. The shape of the image is also represented in terms of seven second-order and third-order invariant moments.

In addition to a curve matching, shape matching is also achieved by matching skeletal graphs (medial axis graphs) as done by Sebastian et al. [6]. Because outline curves typically do not meaningfully represent the interior of the shapes; therefore, the medial axis has been effectively used for matching shapes. In their work, the shock graph, which is the medial axis endowed with geometric and dynamics information, is used because it gives a richer description of shapes.

3 Our Work

Our work begins with image segmentation, which enables us to globally identify structures of the image. The JSEG algorithm [2, 3] is used. It involves two independent steps: color quantization and spatial segmentation. First, the image pixel colors are quantized to several classes called class maps, which are used to differentiate regions in the image. Second, a region-growing method is used to segment the image based on the property of the class maps. A good segmentation method would certainly help us obtain good image contours and their relationships, which eventually influence later steps in image retrieval and clustering. Therefore, improvement steps for cleaning contours either by synthesis or analysis can be performed afterward.

A B-spline curve [4, 7] is more widely and suitably used to represent a complex curve than a far higher degree Bezier curve because of its local control property and its ability to interpolate or approximate a curve with lower degree. The B-spline curve is a generalization of the Bezier curve and has more desired properties than the Bezier curves. The B-spline has the following important properties. First, it is a piecewise polynomial curve with a given degree. This property allows for designing a complex curve with lower degree polynomials, using multiple segments joined with certain continuity constraints. Second, the B-spline curve is contained in the convex hull of its control polyline. The polyline is defined by the B-spline control points; therefore, it can be used to represent the shape of the curve. Third, a change of the position of the control points only locally affects the curve. Thus the controllability is more flexible than the Bezier counterpart, and that is important for a curve design. Fourth, no straight line intersects the B-spline curve more times than it intersects the curve's control polyline. This results in a variation diminishing property. Last, affine transformation such as rotation, translation, and scaling can be applied to the B-spline control points quite easily instead of to the curve itself. This results in the affine invariance property.

All those properties hold for the B-spline curves and play an integral part in image shape modeling for our work based heavily on them. The B-spline curve, *C(u)*, is defined as:

$$C(u) = \sum_{i=0}^{h} N_{i,p}(u) P_i$$

where P_i is a control point, p is a degree, u is parameter, and $N_{i,p}$ is a B-spline basis function and is defined recursively as follows:

$$N_{i,0}(u) = \begin{cases} 1 & \text{if } u_i \le u < u_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u)$$

where u_i is known as a knot, where the two curve segments join with certain continuity. Fig. 1 shows a cubic B-spline curve obtained from eight control points. Its control points are shown as dark circles. Gray lines connecting control points are a polyline (or polygon legs), and they can captures the overall shape of the curve.

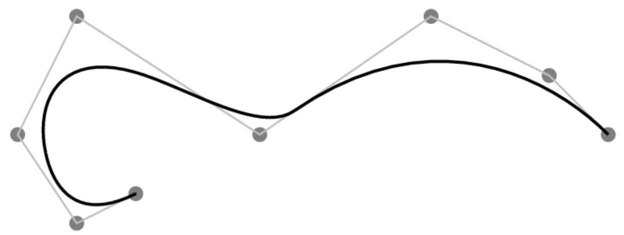


Fig. 1: A cubic B-spline curve with 8 control points that are shown as dark circles.

Given data points, the B-spline curve can either interpolate or approximate those points. In our work interpolation is not practical because of a large number of data points from the image contours. With such number the resulting curve could be wiggly and with a lot of control points if the curve would have passed through all points. Therefore, the B-spline curve approximation is chosen for our work. In the approximation the B-spline curve does not have to pass through all data points except the first and last data points. A number of the B-spline control points would reflect the goodness of the approximation. For each data point, an error distance is computed as the square distance between a data point and a corresponding point on the B-spline curve. A sum of all square error distances is used to measure how well the B-spline curve approximates the data points. An objective is to minimize the sum of the error distance in order to get a good approximation of the data points.

Therefore, a problem statement can be posed as:

Input: Given a set of n+1 data points, $D_0, \ldots, D_n$, in a given order.

Output: A B-spline curve of degree p with h+1 control points, $P_0, \ldots, P_h$, which satisfies the following two conditions:

1. The curve interpolates the first and last data points, D_0 and D_n and
2. The curve approximates the data points in the sense of a least square error distance.

Because the curve contains the first and last data points, we have $D_0 = P_0$ and $D_n = P_h$. The curve equation is now written as:

$$C'(u) = N_{0,p}(u)D_0 + \left(\sum_{i=1}^{h-1} N_{i,p}(u)P_i \right) + N_{h,p}(u)D_n$$

Let parameters be $t_0, \ldots, t_n$. The number of parameters is equal to the number of the data points because we want to find the corresponding point on the curve for each data point. The centripetal parametrization is used and computed as:

$$\frac{\Delta_i}{\Delta_{i+1}} = \left[\frac{\|\Delta x_i\|}{\|\Delta x_{i+1}\|} \right]^{1/2}$$

where $\Delta_i = t_{i+1} - t_i$ and $\Delta x_i = D_{i+1} - D_i$.

And the sum of all square error distances is computed as:

$$f(P_1, \ldots, P_{h-1}) = \sum_{k=1}^{n-1} \left| D_k - C(t_k) \right|^2$$

The control points, $P_1, \ldots, P_{h-1}$, are sought such that the objective function $f(\ )$ is minimized. Extensive coverage of the B-spline approximation is given in [4, 7].

Fig. 2 shows the B-spline approximation of a given 225 data points. Various numbers of control points are used, from 6, 10, 15, to 20. More control points offer higher flexibility and result in the curve closer to the data points. To the extreme if the number of control points equals the number of data points, the resulting curve is simply an interpolating curve. So, how many control points should be used to reasonably approximate the data points? A universal answer to this question does not exist; it depends on each application at hand. The data points used in our work are obtained from contours of the segmented regions after performing image segmentation on the image. The segmentation technique used is based on the unsupervised segmentation algorithm, JSEG [2, 3].

Fig. 3, Fig. 4, and Fig. 5 show examples of the B-spline curve fittings to given contours of extracted regions shown in shaded areas. Fig. 3(d) shows the B-spline curve approximation of 674 data points of one contour. The obtained approximating curve is smooth, visually pleasing, and represented by just 30 control points instead of the original 674 data points. Approximation is also needed for other contours of the image and is stored for later use. After obtaining a complete approximation to all structures of the image, a location, arrangement and composition of those structures (also called primitives) are analyzed and used as one key feature for the image retrieval, clustering and classification. Similar argument can be made for the Fig. 4 and Fig. 5.

4 Conclusion

The proposed approach extracts structural shapes of the image and uses well-defined B-spline curves to approximate the boundary of each segmented shape. The B-spline curves are used in our work due to their compactness, continuity, local shape control property, and affine invariance. Most prior work has used B-spline curves of profile boundaries to match a shape of one object as a whole to others in object database.

Our work extends the curve fitting to each segmented region of the image/object, in finer details instead of the merely global aspect of the image/object. The composition of the approximating curves of all structures is used to represent the original image. Arrangement of the curves' control points and geometrical relationship among those structures can collectively be used to compare similarity between images/objects.

5 Future Work

Our future work is to extend the idea of image shapes' curve representations to image clustering and classification. By using arrangement, both locally and globally, and composition of the B-splines' control points, the similarity of related images can be computed and used to compare, retrieve, cluster, and classify images from a collection. Furthermore, medial axes [6] computed from closed B-spline curves could be used in finding similarity as well. In addition to the B-spline representation, other features such as color histogram from different color spaces, wavelet coefficients, texture, or even textual information could additionally be used.

Having good results of image retrieval, clustering and classification would be somewhat meaningless without a good user interface to understand and further explore the obtained data. Future work in visualization in both a two-dimensional and three-dimensional setting is to be done to help us better view and understand those related images and their relationships in a more meaningful way.

6 Acknowledgments

We would like to thank Y. Deng and B.S. Manjunath, and L. Vandevenne for making their segmentation code and flood-fill code available, respectively. Special thank is also to C. K. Shene for his helpful B-spline tutorial notes. This work would not be possible without a kindly support from the School of Information Technology at KMUTT, which provides us a good working and critically thinking environment.

This work is supported in part by the National Science and Technology Development Agency (NSTDA) under grant F-31-205-20-01 and the KMUTT's Research and Intellectual Property Promotion Center (RIPPC).

References:

[1] Cohen, F.S., Huang, Z. and Yang, Z., Invariant matching and identification of curves using B-splines curve representation, *Image Processing, IEEE Transactions on*, Vol. 4, Issue 1, Jan. 1995, pp. 1-10.

[2] Deng, Y. and Manjunath, B. S., Unsupervised segmentation of color-texture regions in images and video, *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI '01)*

[3] Deng, Y., Manjunath, B. S. and Shin, H., Color image segmentation, *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, CVPR '99, Fort Collins, CO, Vol. 2, Jun. 1999, pp. 446-51.

[4] Farin, G. E., *Curves and Surfaces for CAGD: A Practical Guide*, Morgan Kaufmann; 5th edition, Oct. 2001.

[5] Mokhtarian, F. and Mackworth, A. K., A theory of multiscale, curvature-based shape representation for planar curves, *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, Vol. 14, Issue 8, Aug. 1992, pp. 789-805.

[6] Sebastian, T. B. and Kimia, B. B., Curves vs. skeletons in object recognition, *International Conference on Image Processing*, 2001. Proceedings, Vol. 3, Oct. 2001, pp. 22-25.

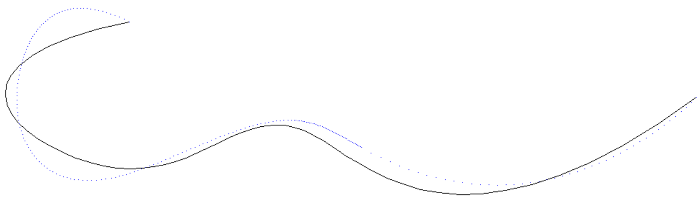
[7] Shene, C. K., Introduction to computing with geometry, *http://www.cs.mtu.edu/~shene*

[8] Vailaya, A., Zhong, Y. and Jain, A. K., A hierarchical system for efficient image retrieval, *IEEE Proceedings of ICPR '96*.

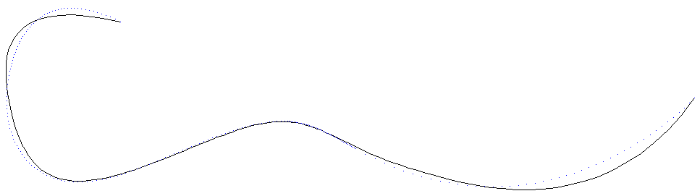
[9] Wang, J. Z., Li, J. and Wiederhold, G., SIMPLIcity: Semantics Sensitive Integrated Matching for Picture Libraries, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 23, No. 9, Sep. 2001, pp. 947-963.

[10] Wang, Y. and Teoh, E. K., A novel 2D shape matching algorithm based on B-spline modeling, *Image Processing, ICIP '04. 2004 International Conference on*, Vol. 1, 24-27, Oct. 2004, pp. 409-412.

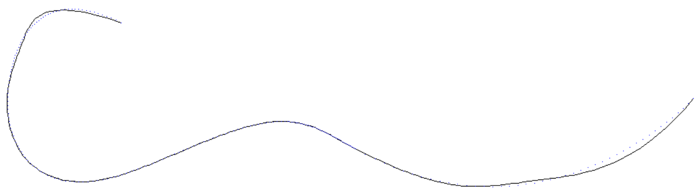
[11] Zhang, D., Image Retrieval Based on Shape, *Ph. D. Dissertation*, Mar. 2002, Monash University, Australia.



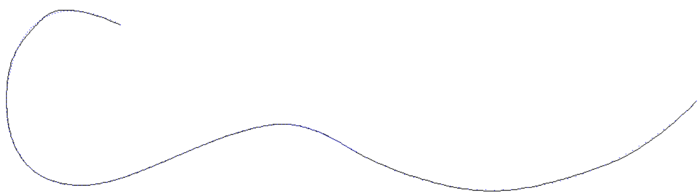
(a)



(b)



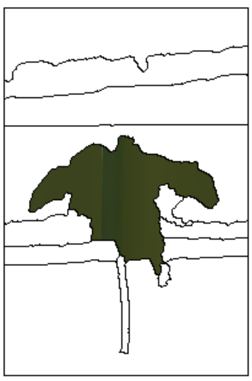
(c)



(d)

Fig. 2: B-spline curve approximation of the 225 test data points with (a) 6 control points, (b) 10 control points, (c) 15 control points, and (d) 20 control points.

(a)



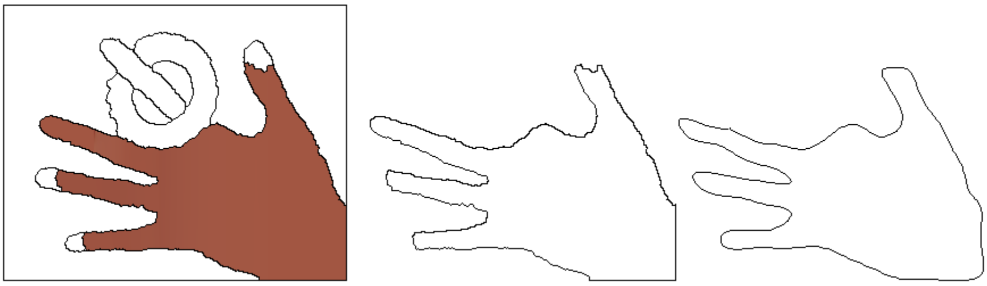
(b)

(c)

(d)

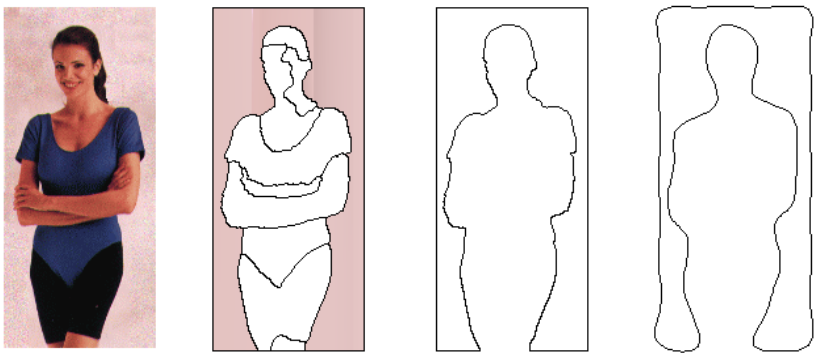
Fig. 3: (a) Original image. (b) A segmented contour and filled region to be approximated by B-spline curve. (c) Extracted boundary with 674 data points. (d) B-spline curve with 30 control points

(a)



(b) (c) (d)

Fig. 4: (a) Original Hand image. (b) A segmented contour and filled region to be approximated by B-spline curve. (c) Extracted boundary with 1535 data points. (d) B-spline curve with 50 control points.



(a) (b) (c) (d)

Fig. 5: (a) Original Woman image. (b) A segmented contour and filled region to be approximated by B-spline curve. (c) Extracted boundary with 1533 data points. (d) B-spline curve with 50 control points.