Boundary Approximation Using B-Spline Curves and Genetic Algorithms

NORTHERN ARIZONA UNIVERSITY
FOUNDED 1899

College of Engineering and Natural Sciences

Michael D. Thomson, Dr. Phillip A. Mlsna

Department of Electrical Engineering, Northern Arizona University, Flagstaff, AZ 86011

Abstract

Our goal is to produce an efficient method of accurately approximating the boundaries of regions extracted from image data. This will enable the efficient comparison of region shapes in a content-based image retrieval application. We use B-splines to represent shape information because 1) their control points are much more compact in storage and 2) translation, rotation, and scaling are easily accomplished in B-spline form. This capability can be greatly utilized in areas such as medical imaging, target recognition, or internet search engines.

We are currently developing software to approximate the closed boundaries of shapes extracted from an image. This approximation uses a set of 2nd-order B-spline curves, defined by a set of control points. We have found that the number of control points used is consistently less than 1/3 the number of boundary pixels. This decreases search and computation time, and also reduces the storage required. This approximation has three main stages:

1) Initial boundary approximation using matrix multiplication with the pixel-defined boundary's maximum curvature points to produce B-spline control points;
2) Error refinement using a gradient-descent algorithm to find local minimum mean square errors (which usually are also global minima);
3) Use of a genetic algorithm to further refine error and to reduce the number of control points used.

Preliminary results and current work are described.

Introduction

Each B-spline curve is made from three control points (see Figure 1). Since the boundary we are approximating is a closed loop, the approximation's B-spline segments also form a closed loop.

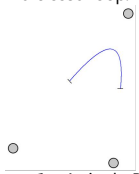

Figure 1 – A single B-spline segment and control points.

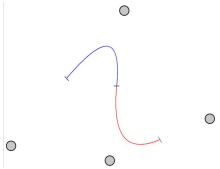

Figure 2 – Two adjacent B-spline segments and their control points.

Some other benefits of using B-splines are:

1) The location where adjacent b-splines meet (called a knot) has a continuous derivative (there is a smooth transition between adjacent splines – see Figure 2.)
2) Manipulation of a control point only affects the adjacent three segments, so local error-reducing adjustments are possible.

Implementation

To calculate the control points for the B-splines, we use

$$\mathbf{C} = \mathbf{R}^{-1}\mathbf{M}$$

where $\mathbf{C}$ is a vector containing the calculated control points, $\mathbf{M}$ a vector of the maximum curvature points in the boundary, and $\mathbf{R}$ is the following nXn matrix (where n is the length of $\mathbf{M}$):

$$\mathbf{R} = \begin{bmatrix} 3/4 & 1/8 & 0 & 0 & \cdots & 0 & 1/8 \\ 1/8 & 3/4 & 1/8 & 0 & \cdots & 0 & 0 \\ 0 & 1/8 & 3/4 & 1/8 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1/8 & 0 & 0 & 0 & \cdots & 1/8 & 3/4 \end{bmatrix}$$

After finding the control points for the B-splines, we calculate the curve produced by using the following equation:

$$P_i(t) = \frac{1}{2}\begin{bmatrix} t^2 & t & 1 \end{bmatrix}\begin{bmatrix} 1 & -2 & 1 \\ -2 & 2 & 0 \\ 1 & 1 & 0 \end{bmatrix}\begin{bmatrix} C_{i-1} \\ C_i \\ C_{i+1} \end{bmatrix}$$

where P is a point on the curve, C indicates a control point from vector $\mathbf{C}$, and t steps from 0 to 1. Higher resolution B-spline curves can be obtained by using smaller steps as t increments from 0 to 1.

Below are some results of the initial boundary approximation (Figure 3) and the gradient-descent error refinement (Figure 4). Gradient-descent refinement is accomplished by finding the largest error in the approximation and nudging the three control points associated with the violating segment. This is repeated until nudging the control points stops yielding improvements.

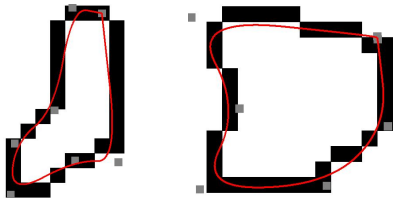

Figure 3 – Some results of the initial approximation.

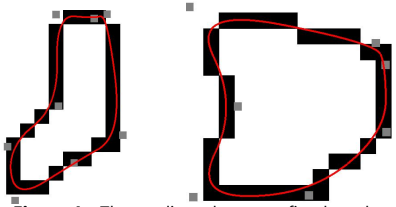

Figure 4 – The gradient-descent refined results.

Results & Analysis

The approximations below (Figures 5, 6, and 7) are typical results from our program. We are currently developing the genetic algorithm aspect of the project. The results shown below were achieved using only the initial approximation and gradient-descent error refinement. Note the versatility of our program by the range of boundary characteristics handled.

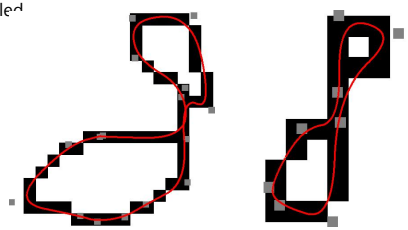

Figure 5 – Even when pixels are repeated in the boundary, the resulting approximation is not adversely affected.

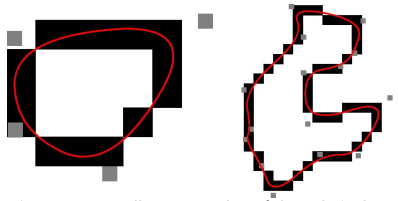

Figure 6 – Excellent examples of the relatively few control points needed to approximate a boundary, for simple or complex boundary shapes.

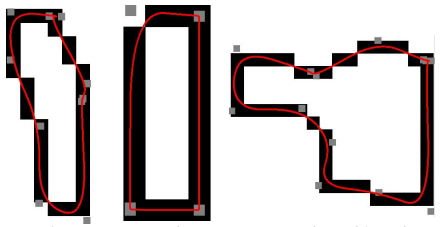

Figure 7 – How sharp corners can be achieved in an approximation (by using two control points at the same location).

Conclusions & Future Work

Improvements can be made to the number of control points used in several of the results shown above (similar approximations could be achieved with fewer control points). It is also likely that many approximation errors that cannot be reduced any further with gradient-descent can be reduced through some other method. This is a prime example of where a genetic algorithm will be valuable. Using a genetic algorithm is key to minimizing the storage size and error of the B-spline approximations.

Genetic algorithms are ideal for reducing error beyond local minima. This is done by introducing quasi-random values into the data being used that can "bump" the error out of a local minimum and into either a better local minimum or a global minimum. The steps to accomplish this are:

1) Make a population of the data to be optimized and randomly alter and mutate (within constraints) data of members in the population.
2) Find the best members of the population. These best combinations of data values are kept, and the worst combinations discarded.
3) Of the population remaining, individual members have values mixed and switched between each other, with a few random mutations applied as well.
4) This newly arranged population once again keeps the best combinations and discards the worst combinations.
5) Repeat steps (3) and (4) until no more significant improvements are observed. After only a few iterations, the error values tend to be closer to global minima instead of local minima.

For our purposes, this algorithm can be used for refining the location of control points (to reduce error), adding control points where it will significantly decrease error, and removing control points where it will not greatly increase error (which also reduces storage size).

References

Anand, Vera B. *Computer Graphics and Geometric Modeling for Engineers*. New York: John Wiley & Sons, 1993. 246-277.

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

El Doker, T. A., and Mlsna, P. A. "Efficient Unsupervised Estimation of Second-Order B-Spline Contour Descriptors." *Proc. of IEEE Midwest Symposium on Circuits and Systems*. Tulsa, OK, Aug. 4-7, 2002.

Acknowledgements

Erica Liszewski and other students have contributed to this work in earlier phases.