AN EFFICIENT BOUNDARY ENCODING SCHEME WHICH IS OPTIMAL IN THE RATE DISTORTION SENSE

*Fabian W. Meier, *Guido M. Schuster and Aggelos K. Katsaggelos*

Northwestern University, Department of Electrical and Computer Engineering,
Evanston, IL 60208-3118, USA, email: fabian@ece.nwu.edu, aggk@ece.nwu.edu

*U.S. Robotics, Advanced Technologies Research Center, Mount Prospect, IL 60065, USA
email: gschuste@usr.com

ABSTRACT

A major problem in object oriented video coding is the efficient encoding of the shape information of arbitrarily shaped objects. Efficient shape coding schemes are also needed in encoding the shape information of Video Object Planes (VOP) in the MPEG-4 standard. In this paper, we present an efficient method for the lossy encoding of object shapes which are given as 8-connect chain codes [1]. We approximate the object shape by a second order B-spline curve and consider the problem of finding the curve with the lowest bit rate for a given distortion. The presented scheme is optimal, efficient and offers complete control over the trade-off between bit-rate and distortion. We present results with the proposed scheme using objects shapes of different sizes.

1. INTRODUCTION

This research is motivated by the importance of shape coding within the MPEG-4 standard [2], and in object oriented video coding [3]. In this paper we refer to the shape information of a single object as a *boundary* (sometimes also referred as contour). We measure the performance or rate of a shape coding scheme with the relative measure e in bits per boundary point (bbp). Rate e is calculated by dividing the total rate needed to encode the boundary approximation by the number of boundary points. For lossy encoding, using a coding performance measure is only meaningful if the distortion measure is also known.

A simple way to represent object boundaries is with the use of a chain code. Freeman [4] originally proposed the use of chain coding for boundary quantization and lossless boundary encoding. The 8-connected chain code encodes one of the 8 possible steps to get from a pixel to one of its closest neighboring pixels with a rate of 3 bbp.

In [5, 6] vertices were found in an optimal way to approximate boundaries with polygons. In this paper we extend this lossy boundary encoding approach and approximate boundaries with *quadratic uniform B-splines.* An iterative encoding approach employing third order B-spline curves was proposed in [7]. The results, however, are not unique nor optimal and depend on the initial curve.

In the following the problem to be solved is formulated in Sec. 2 and the proposed algorithm is developed in Sec. 3. Experimental results are described in Sec. 4 and conclusions in Sec. 5.

2. PROBLEM FORMULATION

The goal of the proposed algorithm is to find a second order B-spline curve that approximates a given boundary using the smallest number of bits, without exceeding an allowable distortion. In this constrained optimization problem we have to find a set of control points - defining the B-spline curve - that can be encoded with the lowest possible rate and the approximation error (distortion) must be below a certain limit. Once we find an optimal solution to this problem we are able to find a solution to the dual problem, that of finding a B-spline curve approximation with the lowest possible distortion given a maximum rate R_{max}, iteratively.

Definition of a B-spline Curve: A B-spline is a specific curve type from the family of parametric curves. It consists of one ore more *curve segments.* Each curve segment is completely defined by $(n+1)$ *control points* where n defines the order of the curve. The second order B-spline curve segment Q_u with control points (p_{u-1}, p_u, p_{u+1}) and the constant base matrix M is defined as follows:

$$\begin{aligned} & Q_u(p_{u-1}, p_u, p_{u+1}, t) = T \cdot M \cdot P \\ & = \begin{bmatrix} t^2 & t & 1 \end{bmatrix} \cdot \begin{bmatrix} 0.5 & -1.0 & 0.5 \\ -1.0 & 1.0 & 0.0 \\ 0.5 & 0.5 & 0.0 \end{bmatrix} \\ & \cdot \begin{bmatrix} p_{u-1,x} & p_{u-1,y} \\ p_{u,x} & p_{u,y} \\ p_{u+1,x} & p_{u+1,y} \end{bmatrix} = \begin{bmatrix} x(t) & y(t) \end{bmatrix}, \end{aligned} \tag{1}$$

for $0 \leq t \leq 1$, and 0 otherwise

$p_{u,x}$ and $p_{u,y}$ are the horizontal and vertical coordinates of control point P_u, respectively. Every point of the curve segment can be calculated with Eq. (1) by letting t vary from 0 to 1. Every curve segment starts ($t = 0$) exactly midway between the first and the second control point, and ends ($t = 1$) exactly midway between the second and the third control point. Note that every curve segment shares control points with its neighboring curve segments; control points p_{u-1} and p_u are used by the previous curve segment Q_{u-1}, and p_u and p_{u+1} are used by the next curve segment Q_{u+1}. When we use a double point, such as $p_{u-1} = p_u$, the curve segment Q_u will begin exactly at the double control point. We apply this property at the beginning and the end of the curve. The reason for choosing the B-spline with the lowest possible order (n=2) is to keep the complexity of the curve, and the proposed algorithm, small. Note that a first order B-spline is a polygon.

Notation: The following notation will be used. Let $B = \{b_0, \ldots, b_{N_B-1}\}$ denote the boundary which is an ordered set. b_i is the i-th point of B and N_B is the total number of points in B. Let $P = \{p_0, \ldots, p_{N_P+1}\}$ denote the set of control points of the B-spline curve, which is also an ordered set, with N_P the total number of curve segments. Every B-spline curve segment is defined by three control points p_{u-1}, p_u and p_{u+1}, henceforth denoted by $Q_u(p_{u-1}, p_u, p_{u+1})$ without the use of t as in Eq. (1), or simply by Q_u. We assume that the locations of the control points of the curve are encoded using a predictive scheme where $r(p_{u-1}, p_u, p_{u+1})$ is set equal to the number of bits needed to encode the relative location of control point p_{u+1} if the locations of p_{u-1} and p_u are known.

Distortion Measure: In general we are interested in a curve distortion measure which can be used to determine the approximation quality of an entire curve. We chose the *maximum absolute distance* between the original boundary and its approximation as distortion measure. The distortion function measures the absolute distance between every boundary point and the closest point of its approximated representation. If we imagine a *distortion-band* with width $2 \cdot D_{max}$ along the boundary B, a B-spline approximation must therefore always be inside the band in order to satisfy the maximum absolute distance distortion requirement. We define the distortion function d for a single curve segment as follows:

$$d(p_{u-1}, p_u, p_{u+1}) = \begin{cases} 0: & Q_u \text{ inside band} \\ \infty: & \text{any point of } Q_u \text{ outside band} \end{cases} \quad (2)$$

A distortion band can be defined by assigning all pixels to the band that are within a certain distance D_{max} from every boundary pixel. For our experiments we chose a distortion band with a sub-pixel resolution of 1/3 pixel (see Figure 1). Eq. (2) is implemented by quantize the curve segment Q_u to 1/3 pixel resolution in a first step. In a second step every curve pixel is tested whether it is located inside or outside the distortion band in order to determine the output value of d.

Admissible Control Point Set: From a theoretical point of view, the set of admissible control points for a B-spline boundary approximation should contain all pixels in the image plane. In order to keep the algorithm efficient, we restrict the control points to a set of relevant locations. We call this the *set of admissible control points* A and define it as a band along the boundary B, where the band is determined by W_{max}. W_{max} is measured from the center of the boundary pixel to the center of the admissible control point pixel. Set A must also be ordered to employ the presented boundary approximation algorithm. We therefore propose to order set A by assigning all points of A to their nearest boundary point and then imposing the order of the boundary onto the set A. Details on the assigning algorithm can be found in [6].

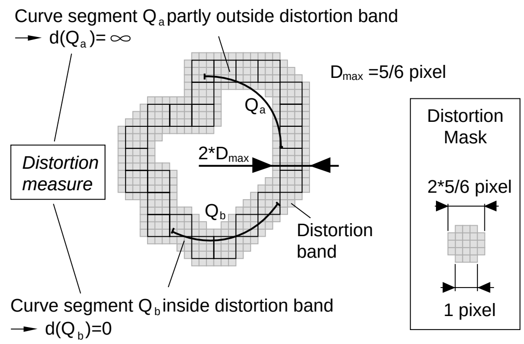

Figure 1: *Implementation of Distortion Measure* $d(\cdot)$: The distortion band of width $2 \cdot D_{max}$ along the object boundary B consists of sub-pixels with 1/3 pixel resolution.

3. THE SHAPE CODING ALGORITHM

Our approach for finding an optimal B-spline approximation for a given boundary is to model the set of all possible B-spline curves with a weighted directed acyclic graph (DAG). Once we have defined a graph, we find the best boundary approximation with a shortest path algorithm. In this paper we use the terms *state* and *vector* instead of the terms *vertex* and *edge* commonly used in graph theory.

Figure 2 illustrates in the form of an example how a DAG is derived from a boundary ($N_B = 7$) and how the shortest path solution leads to a lossy shape approximation. In Figure 2.A the admissible control point set A is equal to the set of boundary points B; in this case the only valid control point locations are boundary pixels. The reason of A being very small is to keep the complexity of this example as low as possible. The DAG in Figure 2.B consists of states and vectors. Several states are associated with a single admissible control point a_i. Every state is uniquely described by two admissible control points (a_j, a_i), where a_j refers to the state's connecting previous state, with the condition $j < i$ for the indices[1]. A vector $\vec{E}$ starts at control point $p_u = a_j$ and ends at control point $p_{u+1} = a_i$.

The curve segment distortion $d(\cdot)$ of Eq. (2) can be combined with the segment rate $r(\cdot)$ by defining a weight function w for the vectors as follows,

$$w(p_{u-1}, p_u, p_{u+1}) = r(p_{u-1}, p_u, p_{u+1}) + d(p_{u-1}, p_u, p_{u+1}) \quad (3)$$

Note that w is equal to the rate for all the curve segments which satisfy the distortion constraint of Eq. (2), but infinite for those which do not. Eq. (3) has three input variables; the first two variables p_{u-1} and p_u represent the two admissible control points associated with the state where the vector begins and the third variable p_{u+1} is the admissible control point associated with the state where the vector ends. Every state together with a vector represents a

[1] The following exception is necessary to allow double control points at the beginning and the end of the curve : $i = j$ if $\{i = 0, i = N_B - 1\}$

B-spline curve segment, so that any path in the DAG from state (a_0, a_0) to state (a_6, a_6) is a possible curve approximation. Let $R^*_{(a_i,a_j)}$ be the best total rate of the path from the first state (a_0, a_0) to state (a_i, a_j) and $R^*_{(a_i,a_j)}$ is the sum of all the weights of that path. $Ptr(a_i, a_j)$ is a back pointer that is used to remember that path.

The task of the shortest path algorithm is to find a path from state (a_0, a_0) to state (a_6, a_6) with the lowest total weight, which is clearly $R^*_{(a_6,a_6)}$. Because we interpret the length of a vector as the number of bits necessary to encode that vector, the shortest path is the path with the lowest total bit-rate. Once a shortest path has been found (Figure 2.C), all admissible control points assigned to the states of this path (Figure 2.D) define completely the control points for the B-spline approximation. We are using the existing single source DAG shortest-path algorithm [8] which is even faster than Dijkstra's algorithm because of the acyclic nature of the DAG.

Again, note that the definition of the weight function w leads to a length of infinity for every path that includes a curve segment with an approximation error larger than D_{max}. Therefore a shortest path algorithm will not select a path with one or more distorted curve segments.

Control Point Encoding Scheme: So far, any control point encoding scheme which satisfies the assumption that the control points are encoded differentially, i.e., the rate to encode point p_{u+1} depends only on the previous two points, p_{u-1} and p_u, could have been used. In this paragraph we present a specific control point encoding scheme to encode the vector $\vec{E}_u$ between the control points p_u and p_{u+1}. We encode the vector between two control points by an angle α and a run β, which form the symbol (α,β). We employ a logarithmic code [6] for encoding the runs β. In this scheme the run of one pixel length has a codeword length of 2 bits and the longest encodable length of 15 pixels requires 5 bits to encode. In natural boundaries, the arrival direction of a vector is highly correlated with the departure direction of the following vector. This implies that the arrival direction should be used to predict the departure direction. We predict that the absolute angle of the departure angle is the same as the absolute angle of the arrival angle. We propose to encode only the four most probable difference angles $\{-90^o, -45^o, +45^o, +90^o\}$, where 0^o is the direction of the previous vector. Clearly we need only 2 bits for the angle information α. The rate function $r(p_{u-1}, p_u, p_{u+1})$ must consider the case when a vector cannot be encoded; that is, either when the vector is longer than 15 pixels or the difference angle is not one of the valid angle values. If this happens, the rate r is set equal to infinity.

4. EXPERIMENTAL RESULTS

To demonstrate the proposed shape coding scheme we encoded three different objects boundaries (Shapes 1, 2 and 3) with 70, 158 and 257 boundary points. For the encoding simulations we varied the maximum distortion D_{max} from 0.4 to 3.0 pixels. We set the width W_{max} of the admissible control point band A equal to 1.0 for all our experiments. The rate to encode the absolute position of the first control point is neglected since it depends on the size of the image. Figure 3 shows the performance of the shape coding algorithm in form of a rate-distortion curve. Average encoding rates e in the range of 0.70 ...0.84 bbp were achieved in our experiments with distortion values of D_{max}=1.0, and e=0.57 ...0.64 with D_{max}=2.0. Figure 4 shows the original object shape of Shape 2 and three approximations with distortion values of 0.8, 1.4 and 3.0. In Figure 5 the B-spline curve approximation as well as the distortion band (D_{max}=1.0) for Shape 1 are shown.

5. CONCLUSIONS

The contribution of this paper is a general and clear mathematical description how to approximate a given object boundaries by a B-spline curve. Based on the mathematical model we find a optimal solution in terms of the bit-rate for the stated problem. Existing and future shape coding algorithms can be compared with the described method. For example the bit-rate efficiency of a low complexity shape-coder can be assessed if the optimal solution is known. As was also mentioned earlier higher order curves as well as distortion bands of variable width can be incorporated into the proposed algorithm in a straightforward way.

6. REFERENCES

[1] F. W. Meier, G. M. Schuster, and A. K. Katsaggelos, "An efficient boundary encoding scheme using B-spline curves which is optimal in the rate-distortion sense," in *2nd Erlangen Symposium, Advances in Digital Image Communication*, (Erlangen, Germany), pp. 75–84, Apr. 1997.

[2] M. R. Banham and J. C. Brailean, "An overview of the MPEG-4 standard: Enabling digital multimedia compression," in *International conference on advanced science and technology*, (Schaumburg, IL), pp. 2–19, 1997.

[3] H. Musmann, M. Hötter, and J. Ostermann, "Object-oriented analysis-synthesis coding of moving images," *Signal Processing: Image Communication*, vol. 1, pp. 117–138, Oct. 1989.

[4] H. Freeman, "On the encoding of arbitrary geometric configurations," *IRE Trans. Electron. Comput.*, vol. EC-10, pp. 260–268, June 1961.

[5] G. M. Schuster and A. K. Katsaggelos, *Rate-Distortion Based Video Compression, Optimal Video frame compression and Object boundary encoding.* Kluwer Academic Press, 1997.

[6] G. M. Schuster and A. K. Katsaggelos, "An efficient boundary encoding scheme which is optimal in the rate distortion sense," in *Proceedings of the International Conference on Image Processing*, vol. II, (Lausanne, Switzerland), pp. 77–80, Sept. 1996.

[7] R. Lagendijk and J. Biemond, "Low bit-rate coding for mobile multimedia communications," in *Proceedings of the European Signal Processing Conference*, pp. 435–438, 1996.

[8] T. Cormen, C. Leiserson, and R. Rivest, *Introduction to algorithms.* McGraw-Hill Book Company, 1991.

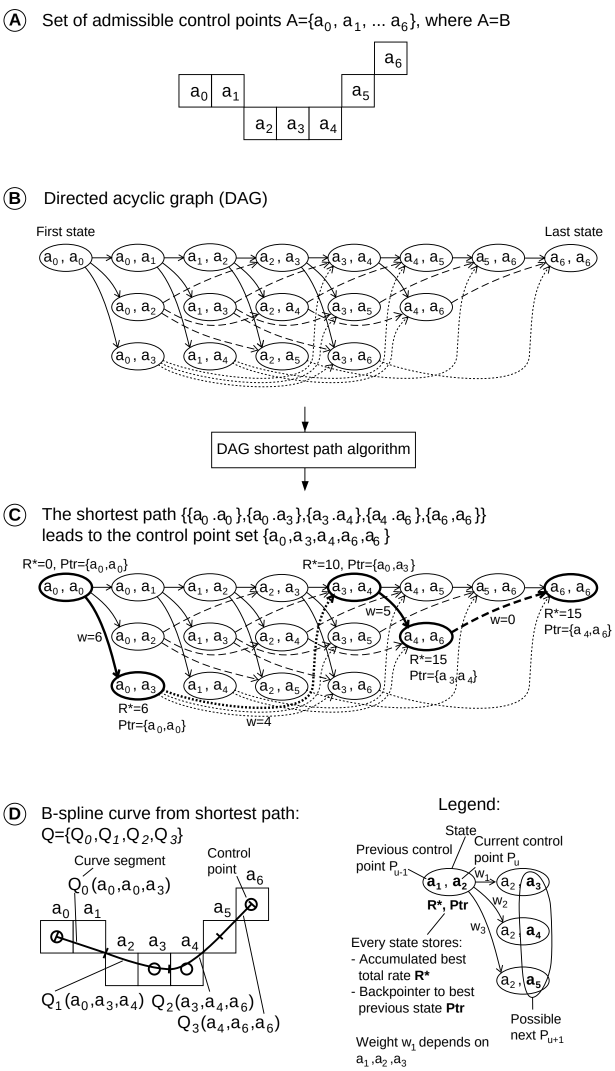

Figure 2: *Approximation of boundary B through a B-spline curve:* Once a set of admissible control points A is defined (A), a DAG can be defined (B). The shortest path algorithm finds a set of control points of the shortest path (C) from state (a_0, a_0) to state (a_6, a_6). The control point set defines the B-spline curve approximation (D) of the original boundary.

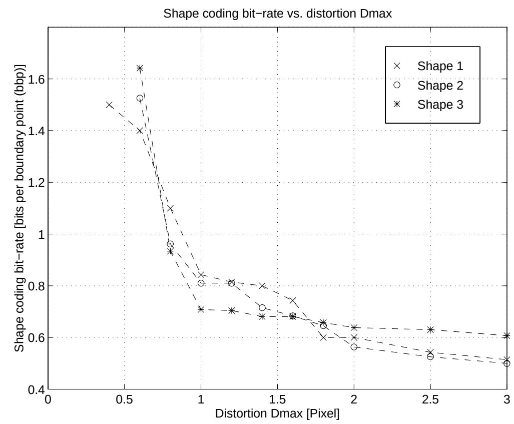

Figure 3: *Rate-Distortion Curve:* Shape encoding bit-rate e in bits per boundary point (bbp) vs. the maximal distortion D_{max}.

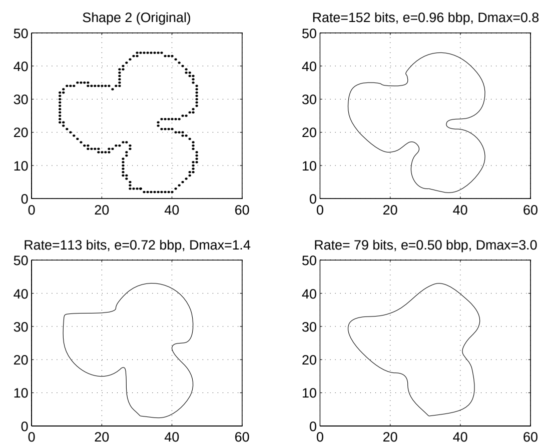

Figure 4: Object shape approximations of shape 2 with different distortion values.

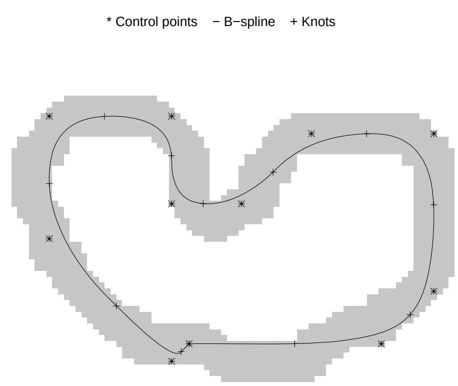

Figure 5: B-spline approximation and distortion band (width $= 2 \cdot D_{max} = 2 \cdot 1.0$) of Shape 1. Bit-rate e=0.84 bbp, 59 bits, N_B=70. Resolution of the distortion band: 1/3 pixel, resolution of control points: 1 pixel.