EXPLOITING TEMPORAL CORRELATION IN SHAPE CODING

Gerry Melnikov, Guido M. Schuster and Aggelos K. Katsaggelos*

Northwestern University
Electrical and Computer Engineering Dept
Evanston, Illinois 60208, USA
Email: {gerrym,aggk}@ece.nwu.edu

*3COM
Advanced Technologies Research Center
Mount Prospect, Illinois 60056, USA
Email: Guido_Schuster@3com.com

Abstract

This paper investigates ways to explore the between frame correlation of shape information within the framework of an operationally rate-distortion (ORD) optimal coder. Contours are approximated both by connected second-order spline segments, each defined by three consecutive control points, and by segments of the motion-compensated reference contours. Consecutive control points are then encoded predictively using angle and run temporal contexts. We utilize a novel criterion for selecting global object motion vectors, which further improves efficiency. Formulating this problem as Lagrangian minimization, we employ an iterative technique to remove dependency on a particular VLC and jointly arrive at the ORD optimal solution and its underlying conditional parameter distribution.

1. INTRODUCTION

In the process of evaluating competing techniques for the MPEG-4 standard, several binary coders were considered. These coders, however, lack optimality in their both intra and inter modes of operation. The context-based (CAE) coder [1] capitalizes on temporal redundancy by performing object-based motion compensation and extending the context template into the neighboring pixels of the reference frame. Similarly, the MMR coder [10] differs from its intra mode counterpart in the choice of pixels serving as context. In the baseline and the vertex-based polynomial approaches (inter mode) [3, 6] a contour in the current frame is approximated through motion compensation by a contour in the previous frame, with segments exceeding a certain error threshold coded in their respective intra mode. All of these coders are ad-hoc in the intra mode, and, therefore, also in the inter mode. They fail to achieve operational optimality since they neither take the tradeoff between the rate and the distortion into account nor do they use the distortion metric used for their evaluation in the encoding process.

We have previously proposed optimal approximations of a given boundary based on curves of different orders and for various distortion metrics [2]. Recently, operationally optimal vertex-based coders were proposed for the intra mode [9, 4]. In [5] this problem was solved optimally and jointly with the variable-length code selection. In this work we extend this ORD optimal framework to take into account the temporal contour redundancies present in typical video sequences. We employ a novel criterion for global object-based motion vector selection which fits naturally into the chosen code structure. We adaptively switch between context and tracking modes to better capitalize on temporal redundancies.

In addition to arriving at the inter mode ORD optimal representation of a sequence for a particular coding framework, characterized by fixed VLC tables, we employ an iterative procedure to find the underlying parameter probability distribution resulting in the most efficient ORD curve.

This paper is organized as follows. The algorithm structure is presented in Sec. 2. The framework for taking advantage of frame to frame correlation between contours is addressed in Sec. 2.1. Section 2.2 deals with the context-based control point encoding scheme, while the additive distortion metric is discussed in Sec. 2.3. Section 2.4 describes how the problem can be formulated as a shortest path problem and Sec. 2.5 discusses VLC optimization issues. Finally, results are presented and discussed in Sec. 3.

2. PROPOSED ALGORITHM

In this paper we solve the problem of contour approximation optimally in the ORD sense. Contours are approximated by connected 2^{nd}-order B-spline segments, each defined by 3 consecutive control points, (p_{u-1}, p_u, p_{u+1}). Thus an ordered set of control points constitutes a code for a shape approximation. A 2^{nd}-order spline is a parametric curve (parameterized by t) that starts at the midpoint between p_{u-1} and p_u and ends at the midpoint between p_u and p_{u+1} as t sweeps from 0 to 1. These midpoints are also called knots. A precise mathematical definition

of this curve is given in [2]. A sequence of 2^{nd}-order B-splines solves the interpolation problem at the knots, while being differentiable everywhere, including the knots. This smoothness property, coupled with the simplicity of definition, makes B-splines a natural choice for the shape coding applications.

Although an ordered set of control points defining approximating splines may contain elements from anywhere in the image, it is unlikely that locations far from the original boundary would lead to an ORD optimal approximation. This leads naturally to the concept of the admissible control point band [9], thus excluding from consideration all pixels located farther than the band width away from the original boundary.

2.1. Temporal Correlation

It is intuitively clear that object boundaries between frames are correlated. However, efforts to gain coding efficiency based on this apparent redundancy have, so far, been relatively unsuccessful [2]. In the proposed context methods [1, 10], one global motion vector, which minimizes the number of mismatched pixels, is employed to align corresponding objects in two consecutive frames. A context for a pixel in the current frame is then computed from its spatio-temporal neighborhood. The main disadvantage of this approach is its pixel-based nature, which suffers from mis-alignments due to motion and noise. Noise is introduced to contours during the frame acquisition and segmentation processes, which causes consecutive contours to be different even without motion.

In this paper we reduce contour noise effects by utilizing temporal contexts for predictive encoding of the $(angle,\ run)$ symbols, instead of the underlying pixels. The $(angle,\ run)$ coding framework for consecutive control point locations was used in [5, 9] in conjunction with B-splines to arrive at an ORD optimal representation of a boundary in the intra mode. The contexts for the $angle$ and run components are searched for in a local window in the motion compensated reference frame and are computed for every pixel in the admissible control point band. Figure 1 depicts a hypothetical context window in the reference frame after motion compensation. Centered on a pixel in the admissible control point band, it is used to extract both the most likely direction and the most likely length of the vector pointing from that pixel to the next potential control point. That is, if an actual control point is located at the current position, this context provides an estimate of where the next control point is most likely to be. The context for the $angle$ component is obtained by selecting the direction in which most of transitions between consecutive boundary pixels occur. This corresponds to the North-West (NW) direction in Fig. (1). Note that this estimate of the direction is very robust to contour noise. Similarly, the run length context is obtained by selecting the most frequently occurring distance between consecutive control points in that window. In this example, a run length of 4, occurring 3 times, is selected. Due to motion and occlusions, however, certain parts of the boundary will have too few reference pixels for a meaningful computation of the context, in which case the algorithm reverts to the intra mode ([5]) for encoding the $(angle,\ run)$ symbol.

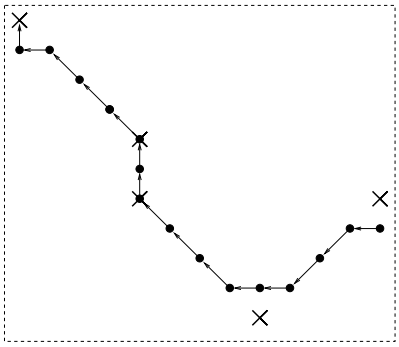

Figure 1: Control points (X marks) and boundary pixels (circles) in a temporal context window. Context: NW direction, run of 4.

With non-homogeneous and non-rigid motion it is often the case that certain contour segments are well approximated in the motion-compensated reference frame, while others are not. For this reason we include, in the source alphabet, symbols representing the tracking of pixels in the reference boundary. Thus, based on the chosen tradeoff between the rate and the distortion, the encoder may select to approximate stretches of the contour under consideration by following the reference contour for n pixels, with each value of n corresponding to one symbol.

The issue of selecting a suitable global motion vector is a non-trivial one. All approaches evaluated by MPEG-4 [2] use the global motion vector minimizing the number of pixels in error between the current and the reference objects. That is, a vector $\bar{M}$ is chosen such that

$$\bar{M} = arg\min ||E_{\bar{M}}||, \qquad (1)$$

where $E_{\bar{M}}$ is the set of all pixels in error under motion compensation of the reference contour by $\bar{M}$. The above criterion tends to spread the error pixels all around the boundary when object motion is non-rigid, which is inconsistent with the objective of tracking the reference contour or utilizing contexts where possible. To better capitalize on the proposed code structure (context and tracking) the following criterion is used to choose a global motion vector:

$$\bar{M} = arg\min \sum_{(i,j)\in E_{\bar{M}}} (i-\bar{i})^2 + (j-\bar{j})^2, \qquad (2)$$

where the summation is over all pixels in error whose center of mass is at $(\bar{i}, \bar{j})$. Roughly speaking, application of the proposed criterion has the effect of pushing the majority of error pixels to one side of the contour, while accurately approximating the rest. This together with the tracking mode makes the encoder efficient with respect to non-rigid object motion. Figure (2) illustrates this concept. Here the

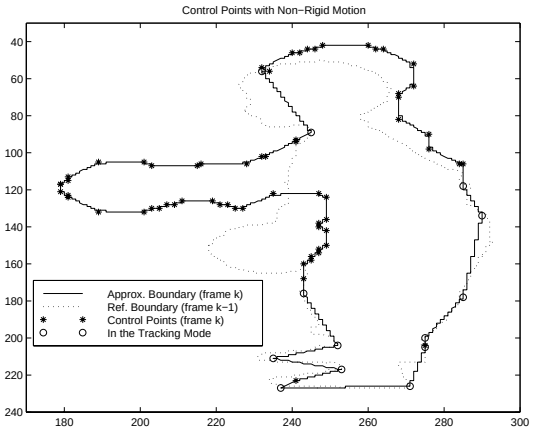

Figure 2: Control point placement under very low bit rate. Circles correspond to the tracking mode.

contour in the current frame is shown by the solid line and the motion compensated (under the proposed criterion) contour is shown by the dotted line. The resulting control points are shown by $*$ and the control points where the current boundary is approximated by tracking the reference boundary are shown by $\circ$.

2.2. Rate

Once the angle and run length contexts are known, the location of the next control point of the object in the current frame is encoded by $(angle,\ run)$ predictively with respect to the contexts. In this framework, shorter codewords are assigned to directions and runs closest to the contexts. Figure (3A) shows a typical conditional direction probability distribution, given the NW context. Similarly, a hypothetical but likely conditional probability distribution for the run length is shown in Fig. (3B) for the case the context is 4. In both cases, vector lengths are proportional to the probability of the corresponding symbol. If a search in a local window results in too few reference boundary or control points, the context is marked as unknown and the intra mode is used, i.e., the angle is encoded predictively with respect to the previous angle in the current frame and the run is encoded in absolute terms. In addition to *angle* and *run* symbols, there are also several symbols in the encoder alphabet corresponding to tracking the reference contour for a number of pixels.

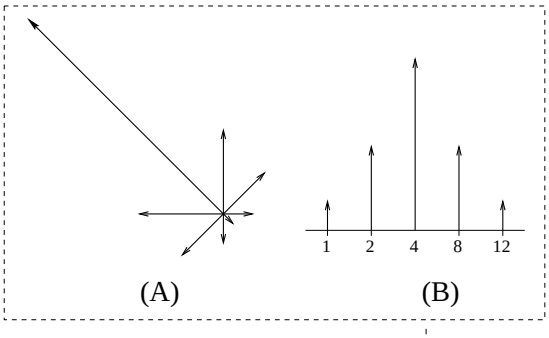

Figure 3: Typical probability assignment for direction (A), and run length (B), with contexts NW and 4, respectively.

Having established the $angle$ and run encoding scheme, we define the total object rate in terms of constituent segment rates. If $r(p_{u-1}, p_u, p_{u+1})$ denotes the segment rate for representing p_{u+1} given control points p_{u-1}, p_u, then the total rate is given by

$$R(p_0, \ldots, p_{N_P-1}) = \sum_{u=0}^{N_p-1} r(p_{u-1}, p_u, p_{u+1}). \qquad (3)$$

Note that $r(p_{u-1}, p_u, p_{u+1})$ implicitly assumes the knowledge of the context at point p_u.

Regardless of the context, the first control point location is encoded absolutely and that cost together with the cost of sending a global motion vector, constitutes an overhead outside the realm of the ORD optimization described in Sec. 2.4. An iterative procedure for selecting efficient variable-length codes for the direction and the run will be discussed in Sec. 2.5.

2.3. Distortion

In MPEG-4 the following additive distortion metric has been used per frame to evaluate performance of competing algorithms:

$$D_{MPEG4} = \frac{\text{number of pixels in error}}{\text{number of interior pixels}}, \qquad (4)$$

where a pixel is said to be in error if it belongs to the interior of the original object and the exterior of the approximating object, or vise-versa.

Spline segment distortions need to be defined in order to evaluate the total boundary distortion. This is done by first associating segments of the approximating curve with segments of the original boundary, as shown in Fig. (4). Here the midpoints of the line segments (p_{u-1}, p_u) and (p_{u+1}, p_u), l and m, respectively, are associated with the points of the boundary closest to them, l' and m'. That is the segment of the original boundary (l', m') is approximated by the spline segment (l, m).

The spline segment distortion $d(p_{u-1}, p_u, p_{u+1})$, shown in Fig. (4), is computed by counting the number of pixels in error (hollow circles on the figure). Note that this requires quantizing the continuous spline to fit the pixel grid of the image. Special care is taken when associating

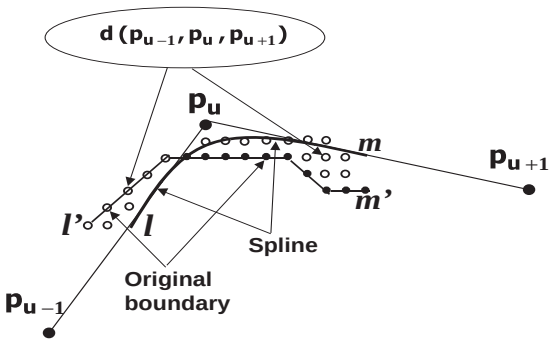

Figure 4: Area between the original boundary segment and its spline approximation (circles).

a spline segment to a segment of the original boundary to ensure that the starting boundary pixel of the next segment coincides with the last boundary pixel of the current segment and that error pixels on the border line between m and m' are not counted twice when computing the next segment distortion. Based on the segment distortions, the total boundary distortion is therefore defined by

$$D(p_0, \ldots, p_{N_P-1}) = \sum_{u=0}^{N_p} d(p_{u-1}, p_u, p_{u+1}), \qquad (5)$$

where N_p is the number of control points and $p_{-1} = p_{N_p+1} = p_{N_p} = p_0$. The last equality ensures that an approximation to a closed contour is also closed and simplifies implementation. It is mentioned here that other additive distortion metrics can be used [2, 8, 9].

2.4. Determining the Optimal Solution

Within the confines of the chosen code structure, we seek an ordered sequence of control points p_i, and their number N_P, which is the solution to:

$$\min_{p_0,\ldots,p_{N_P-1}} D(p_0, \ldots, p_{N_P-1}), \quad subject\ to: \\ R(p_0, \ldots, p_{N_P-1}) \leq R_{max}, \qquad (6)$$

We convert the above constrained minimization problem into an unconstrained one by forming the Lagrangian

$$J_\lambda(p_0, \ldots, p_{N_P-1}) = \\ D(p_0, \ldots, p_{N_P-1}) + \lambda \cdot R(p_0, \ldots, p_{N_P-1}), \quad (7)$$

where for any choice of the multiplier λ, J_λ is the cost function to be minimized. This cost function is expressed as a sum of incremental spline segment costs defined as,

$$w(p_{u-1}, p_u, p_{u+1}) = \\ d(p_{u-1}, p_u, p_{u+1}) + \lambda \cdot r(p_{u-1}, p_u, p_{u+1}). \quad (8)$$

The optimal set of control points $(p_0^*, \ldots, p_{N_P-1}^*)$ is then found by casting the problem as a shortest path in a Directed Acyclic Graph (DAG) with control points playing the role of vertices and incremental costs $w()$ serving as edge weights [2]. Dynamic Programming (DP) is employed to find the shortest path in the DAG for a fixed rate-distortion tradeoff λ. We employ a Bezier curve search [8] in order to arrive at λ^*, the multiplier resulting in the total rate closest to the target rate of R_{max}, in very few iterations.

2.5. Optimizing the VLCs

Clearly our claim of optimality is contingent on the chosen code structure, the motion compensation scheme, the width of the control point band, and, to a great extent, on the VLC tables [5]. Here we follow the iterative procedure proposed in [7, 5] to remove the conditioning of the ORD optimal solution on an ad-hoc VLC. As a result of its application, the solution to the following optimization problem is found

$$\{p_0^*, \ldots, p_{N_P-1}^*\} = \\ arg \min_{p_0,\ldots,p_{N_P-1}; f \in F} [D^*(\cdot) + \lambda \cdot R^*(\cdot)], \qquad (9)$$

where f is a member of the family of context-conditioned parameter probability mass functions F. Hence the shape approximation and the parameter probability model are found jointly and ORD optimally.

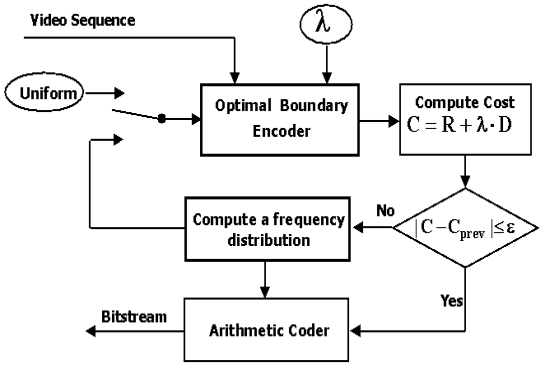

Figure 5: The entropy encoder structure.

In the beginning of the iterative process, depicted in Fig. 5, the encoder compresses a sequence of binary frames in the inter mode with a fixed rate-distortion tradeoff λ and an initial probability mass function for (*direction* | *context*) and (*run* | *context*). Having encoded the input sequence

at iteration k, based on the probability mass function $f^k()$, we use the frequency of the output symbols to compute $f^{k+1}()$, and so on. Since the sequence of selected control points at iteration k is available to the encoder at iteration $k+1$, with the VLCs derived from that sequence, the cost C is a non-increasing function of k. This procedure is guaranteed to converge and the process stops when the cost improvement is less than ϵ, at which point the symbols are arithmetically encoded and sent to the decoded together with the overhead of the two probability mass functions.

3. RESULTS AND CONCLUSIONS

Figure 6 shows the ORD curves of the proposed algorithm for the SIF sequence "kids". The distortion axis d_n represents the average of the D_{MPEG4}'s defined in Eq. (4) for one frame, over 100 frames. As the figure demonstrates, our result compares favorably with both the baseline [3] and the vertex-based [6] algorithms in the inter mode across most of the range of bitrates. In the very low distortion region ($d_n \leq 0.006$) of operation, however, the proposed algorithm requires more bits than both the baseline and the vertex-based methods. This is due to the fact that for near-lossless boundary encoding the chosen code structure (direction plus run) is inefficient.

In this implementation, 8 directions (separated by 45^o) were allowed in the case the context is present. Encoded differentially with respect to the angle context, they correspond to 8 out of 12 conditional symbols for the direction component. The other 4 symbols are used when a context is not present. The run component was represented by 25 symbols, with only 5 symbols (corresponding to runs of 1, 2, 4, 8, and 12) used for any given context. Additionally, 7 symbols were used for the tracking mode, representing 7 different lengths (spaced uniformly from 15 to 45) for which a reference contour could be tracked. Bit-rates for the proposed method, reported in Fig. 6, take into account bits for the global motion vectors, searched in a 32×32 window.

Although the proposed algorithm clearly outperforms existing inter mode techniques, its overall improvement in efficiency of shape representation with respect to the intra mode is not comparable to that of texture representation, where temporal correlation is exploited to a much larger degree. We expect that further gains can be made by the use of a finer quantizer for the angle component, by treating both the angle and the run as one symbol, by adaptively adjusting the size of the context window with λ, and, possibly, by using a more sophisticated motion model. Also, since the use of contexts couples consecutive frames, the global (across objects) optimality is lost. Hence an approach using different values of λ for the same object in different frames may potentially be more efficient.

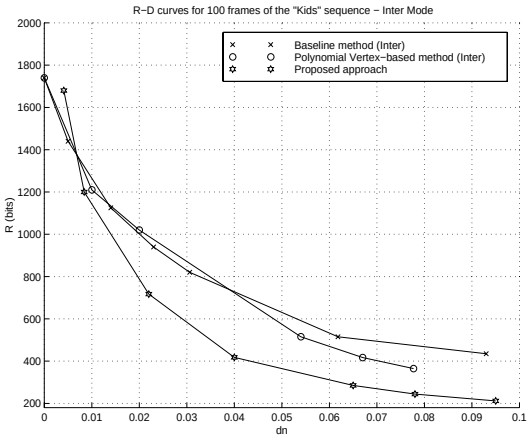

Figure 6: Rate-Distortion curves.

4. REFERENCES

[1] N. Brady, F. Bossen, and N. Murphy, "Context-Based Arithmetic Encoding of 2D Shape Sequences", *Proc. ICIP97*, pp. I-29-32, 1997.

[2] A. K. Katsaggelos, L. Kondi, F. W. Meier, J. Ostermann, G.M. Schuster, "MPEG-4 and Rate Distortion Based Shape Coding Techniques", *Proc. IEEE*, pp. 1126-1154, June 1998.

[3] S. Lee, *et al*. "Binary Shape Coding using 1-D Distance Values from Baseline", *Proc. ICIP97*, pp. I-508-511, 1997.

[4] G. Melnikov, P. V. Karunaratne, G. M. Schuster, A. K. Katsaggelos, "Rate-Distortion Optimal Boundary Encoding using an Area Distortion Measure", *Proc. ISCAS98*, Jun. 1998.

[5] G. Melnikov, G. M. Schuster, A. K. Katsaggelos, "Simultaneous Optimal Boundary Encoding and Variable-Length Code Selection", *Proc. ICIP98*, pp. I-256-260, Oct. 1998.

[6] K. J. O'Connell, "Object-adaptive Vertex-Based Shape Coding Method", *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 7, pp. 251-255, Feb. 1997.

[7] D. Saupe, "Optimal Piecewise Linear Image Coding", *Proc. SPIE Conf. on Visual Comm. and Image Proc.*, vol. 3309, pp. 747-760, 1997.

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

[9] G. M. Schuster, G. Melnikov, and A. K. Katsaggelos, "Optimal Shape Coding Techniques," *IEEE Signal Processing Magazine*, Nov. 1998.

[10] N. Yamaguchi, T. Ida, and T. Watanabe, "A Binary Shape Coding Method using Modified MMR", *Proc. ICIP97*, pp. I-504-508, 1997.