

A SCALABLE VERTEX-BASED SHAPE INTRA-CODING SCHEME FOR VIDEO OBJECTS

M. Hu, S. Worrall, A. H. Sadka, A. M. Kondoz

*Centre for Communication Systems Research, University of Surrey,
Guildford, Surrey, GU2 7XH, U.K.*

ABSTRACT

Scalable shape encoding is one of the important steps to achieving highly scalable object-based video coding. In this paper, a new scalable vertex-based shape intra-coding scheme has been described. To improve the encoding performance, we propose a new vertex selection scheme, which can reduce the number of approximation vertices. We also propose a new vertex encoding method, in which the information on the coarser layers and statistical entropy coding are exploited for high encoding efficiency. Experimental results show that the proposed scheme can provide 25-60% gain over the scalable encoding method in [7]. For some sequences, it can achieve 5-10% gain over conventional non-scalable vertex-based coding method [4] in bit rate, at the price of additional complexity.

1. INTRODUCTION

The emergency of new multimedia applications, such as searching, indexing and manipulation of visual information at the semantic object level, require further research on video representation and coding. In MPEG-4, video frame is composed of objects and each object is represented by three sets of parameters (shape, texture and motion) so that the object can be encoded, accessed and manipulated in arbitrary shape. Among these three sets of parameters, shape information is crucial for object representation and object-based coding. In order to transmit the shape of an object efficiently, a larger number of techniques have been proposed [1-5]. These coding techniques are typically classified as block-based techniques [1] and contour-based techniques [2-5]. These coding algorithms can achieve lossy and/or lossless coding.

In multimedia networks, as devices with different bandwidths and available decoding powers are interconnected, bitstream scalability is also desirable. In recent years, much research has focused on this topic and some methods have been proposed [7]. However, experimental results show that these methods cannot achieve higher compression efficiency for (near-) lossless shape coding.

In this paper, we propose a new scalable vertex-based shape intra-coding scheme for high coding efficiency. To improve coding efficiency, our efforts are concentrated on the progressive approximation and scalable intra-coding. First, an optimal vertex selection scheme is proposed, which can achieve less approximation vertex number. Second, an efficient vertex-encoding scheme is proposed. The information of the transmitted layers is exploited to

improve coding efficiency. This paper is organized as follows. The proposed shape-coding scheme is described in details in section 2. Section 3 presents some experimental results and concludes the whole paper.

2. SCHEME DESCRIPTION

Our proposed scalable vertex-based shape intra-coding scheme consists of two parts: progressive approximation, and scalable vertex coding, which will be discussed in details in the following sections.

2.1. Progressive vertex selection

In recent years, many vertex selection algorithms have been proposed [5-9]. The iterative refinement method (IRM) [5] has been widely used as it can be easily implemented and make control feasible. However, this method cannot give optimal vertices. Recently, optimised vertex selection method [6] has been proposed, which has high computational complexity.

A new vertex selection scheme is proposed to approximate object contour progressively. Before vertex selection, majority filtering [3] and contour refinement are first applied to impose some constraints on the contour. This can reduce the number of vertices during approximation. During approximation, the vertices are classified into several layers according to the selected error bands. In our work, 4 layers are used for QCIF sequences. Layers 0, 1, 2, and 3 have the corresponding error band $d_{\max} = 4$, $d_{\max} = 2$, $d_{\max} = 1$, $d_{\max} = 0$ respectively. These selections are based on the research results in [7]: it does not seem useful to encode more than 4 layers and lossy shape coding should be limited to small distortions for video coding.

For the vertex selection of layer 0, curvature scale space image [9] has been exploited. The vertices of layer 0 should include the salient point of object contour, which can feature contour efficiently, so that they can be decoded first. This is very important for the applications, such as shape retrieving and matching. However, for the contour with small curvature, the method, which is just based on curvature, cannot achieve smaller mismatch error during coding. Therefore, IRM method is used after the curvature-based selection is made.

For other refinement layers, IRM, together with a novel merging scheme, has been exploited. Each approximating polygon edge of the coarser layers is recursively split by introducing a new vertex at the contour point with the largest distance, until the desired accuracy $d < d_{\max}$ is reached. For the layer with $d_{\max} = 0$, the intrinsic image grid quantisation is taken into account.

In order to reduce the approximation vertices and encoding bitrate, a new merging algorithm is proposed and summarized as follows:

- Suppose, at layer $i \in \{1,2,3\}$, the contour has been divided into $\sum_{k=0}^{i-1} N_k$ segments. Along every segment, if the vertex number of layer i is 2 or more, these vertices will be evaluated. If the vertex is removed and the polygon approximation still satisfies the error condition, remove it. Otherwise, keep it.
- If $d_{\max} = 0$, and, for every segment $\overline{p_1 p_2}$, the vertex number of layer i is $n_j \geq 2$, arrange these vertices along the contour in array $pt[]$, including two terminals. Then:


```

      For (k=0; k<nj-1; k++){
      For (kk= pt[k]; kk< pt[k+1]; kk++){
      If (kk== pt[k])
          If we can find  $m_j < n_j - k$  points to approximate segment  $\overline{kk, p_2}$ , recode the points and number;
      Else
          If we can find  $m_j < n_j - k - 1$  points to approximate segment  $\overline{kk, p_2}$ , recode the point positions and number; }
      }
      If (no such point sets are found) { No merge is needed; }
      Else { Replace the vertices of layer  $i$  with the point set with the smallest number. }
      
```

Fig. 1 illustrates an example for the lossless approximation, where (a) is achieved by the published methods in [5] and [7] and (b) is achieved by the proposed method. For the approximation of the contour segment between $X1$ and $X2$, four points are required by using the method in [5] and [7]. While using our proposed method, only one point is chosen. More experimental results in section 3 show that our proposed method can achieve less number of vertices to approximate the object contour, especially for (near-) lossless approximation.

2.2. Scalable vertex encoding

For scalable vertex-based shape coding, there are two kinds of information to be encoded: contour configuration and contour location. Contour configuration is represented by an ordered set of vertices that can be used by decoder to correctly produce the ordered list of vertices. The contour location is represented by the coordinates (x, y) of the vertices. In our proposed method, the scalable vertex encoding consists of the encoding of the vertices in

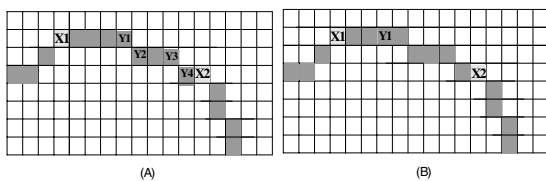


Fig.1 Comparison of different lossless approximation methods

layer 0, vertex connectivity and position encoding of refinement layers.

2.2.1. Encoding of layer 0

As there is no parent approximation for layer 0, the corresponding vertices have to be encoded directly. The encoding of layer 0 consists of two parts: initial vertex encoding and the encoding of other vertices.

In our proposed method, a vertex-reordering step is conducted before initial vertex encoding. As we know, the relative distance between the initial vertex v_0 and the last vertex v_{N_0-1} does not need to be coded. In Object-adaptive vertex-based shape encoding (OAVE) scheme in [4], as the bit assignment during encoding is dynamically determined by the maximum relative distance, the reordering of vertices can reduce the number of encoded bits. After initial vertex is decided, two prediction models have been proposed to encode its position: (a) relative to the origin of the Video object plane (VOP) boundary box; (b) relative to the origins of other contours. During experiments, we found that, if the number of contour of VOP is high, (b) is more efficient than (a).

In the proposed scheme, two encoding models have been used to encode the non-initial vertices: (a) OAVE encoding scheme in [4]. (b) Absolute addressing. Experimental results in [10] show that OAVE combined with adaptive arithmetic coding for encoding a composite relative address shows the best performance. However, at high distortion levels ($d_{\max} \gg 0$), the redundancy of successive octant numbers may be small so that the coding of differential chain code is less efficient. Furthermore, at high distortion levels, the distribution of vertices along the object contour is more irregular. When relative distance of at least one vertex is large, the bit assignments to all other vertices are also large. So, in our scheme, if above circumstance is encountered, absolute addressing is preferable. We choose one of above methods that can produce fewer bits.

2.2.2. Vertex-connectivity encoding of refinement layer

For scalable shape coding, the connectivity, the number of child vertices along the coarser polygon edges, between the vertices should be encoded and transmitted to the decoder. In our research, 2-D symbol (A, B) is defined to indicate the positions, where new vertices will be added. A is to define the number of edges (or vertices) before a

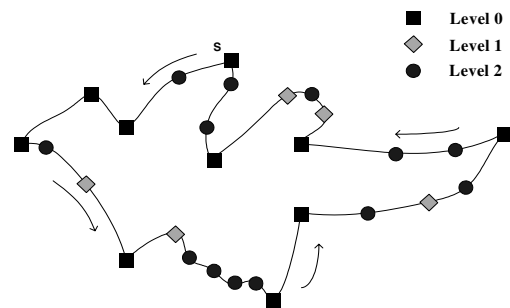


Fig.2 Illustration on vertex-connectivity coding

vertex should be added. B is defined as the number of vertices to be added along this approximation edge. For each approximation layer (except layer 0), 2-D symbols are formed and variable-length coded to produce the bit streams.

The following example illustrates the symbol construction. Assume that we plan to encode the vertices along the object contour in Fig.2. First, 10 vertices in layer 0 are encoded and the vertex S is chosen as the starting point. For layer 1 and layer 2, the 2-D reference symbols are formed as follows:

(4,1), (1,1), (2,1), (2,2) (Layer 1)
 (1,1), (3,1), (3,4), (2,1), (1,1), (1,2), (2,1), (2,2) (Layer 2)

During encoding of these symbols, variable length coder (VLC) has been used. As the number of vertices for every layer has been included in the layer header, the end-of-layer (EOL) information is not needed.

50 video objects from QCIF sequences have been used to study the statistics of symbols, which is shown in Fig. 3. It is found that layer2 1 and 2 have almost the same statistics, which is different from that of layer 3. Therefore, two VLC tables are designed. From Fig. 3, it is also found that, for layer 1 and layer 2, almost 75% of (A, B) symbols with B=1; and for layer 3, almost 75% of (A, B) symbols with A=1. During vertex position encoding, this has been used to improve the encoding efficiency.

2.2.3. Vertex-position encoding of refinement layers

For the vertex position encoding of refinement layers, the improved OAVE encoding scheme in [4] has been used, in which the information from already transmitted coarser layers and approximation error band is exploited for high encoding efficiency. The encoding process includes: Determine and encode two indicators; encode the octant number; and encode the major and minor components.

Determine and encode two indicators: In the proposed method, only the segments containing the newly inserted vertices are used for determining x_{\max} and y_{\max} , which are used to select two indicators: $indicator_x$ and $indicator_y$ as described in details in [4]. The selected indicators are then encoded using a 3-bit FLC.

Encode the octant numbers: In the proposed method, the determinant of octants is different from that in [4]. According to the vertex number n_j along the approximation edge, two octant determination methods

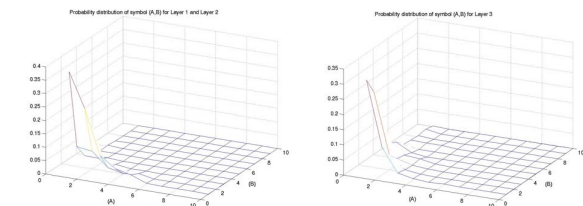


Fig. 3 statistical distribution of (A, B) pairs for Layers 1, 2, and 3

are proposed. If $n_j = 1$, we decide its octant from Fig.4 (a). Otherwise, their octants are decided by Fig.4 (b).

In Fig.4 (a), X_1 and X_2 belong to the coarser layers. Now, if one point Y_1 belongs to current layer, it must satisfy $d_2 \leq d_{\max} < d_1$ and be located in the regions between ℓ_{10} and ℓ_{11} or between ℓ_{20} and ℓ_{21} . If $n_j \geq 2$, they are located in the region between ℓ_{11} and ℓ_{21} in Fig. 4 (b). The regions corresponding to octant from 0 to 7 are separated by ℓ_1 and ℓ_2 as indicated in Fig. 4 (a) and (b) from R_0 to R_7 respectively.

After deciding the octants, they are encoded by using conditional differential chain coding (CDCC). The differential octant of Y_1 is obtained by the difference between neighbouring octant value. In our research, three patterns of previous differential direction are used for classifications and designing the VLC tables.

Encode the major and minor component:

The vertex number along the approximation edges, as shown in Fig.4, also decides the encoding methods.

Situation 1: If $n_j = 1$ and the octant of this vertex is 0, 3, 4, or 7, its x is encoded by using $indicator_x$ of current layer. Its y is decided by value d in Fig.4 (a) and is encoded by VLC. If its octant is 1, 2, 5, or 6, such as Z_1 in Fig. 3 (a), its y is decided by d and encoded by VLC. Its x is encoded by using $\min\{indicator_x, \lceil \log_2(x_1 + 1) \rceil\}$, where x_1 can be deduced from value d and line ℓ_1

Situation 2: If $n_j \geq 2$, as shown in Fig. 4 (b), the first

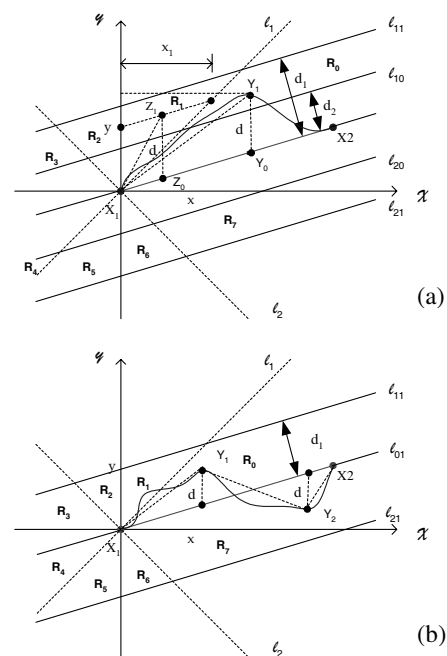


Fig. 4. Encoding vertex position for refinement layers

vertex is encoded by using the same method as above described, except that different VLC Table is used. For other vertices, the method in [4] is used except that the error band is also used to decide the number of bits.

For every refinement layer, two VLC tables are designed for above two situations. For situation 1, the discrete set of d is $\{2,3,4,5,6\}$, $\{1,2,3\}$, $\{0,1\}$ for layers 1, 2, 3, respectively. For situation 2, the discrete set of d is $\{0,1,2,3,4,5,6\}$, $\{0,1,2,3\}$, $\{0,1\}$ for layers 1, 2, 3, respectively. For layer 1 and 2, about 75% of total vertices satisfy situation 1, the value d can be encoded by 1-2 bits. For layer 3, the value d can also be encoded using a single bit as the discrete set of d is $\{0,1\}$.

Therefore, much improvement can be achieved especially for (near-) lossless shape coding as also shown in the experimental results of section 3.

3. EXPERIMENTAL RESULTS AND CONCLUSION

Several video objects are selected to study the performance of vertex selection scheme. Fig 5 lists the required average number vertices under different error criteria for different video objects. Compared with the method in [5] and [7], in which majority filtering is also exploited during performance comparison, our proposed method can achieve up to 30-80% and 20-30% of the total number of vertices for lossless reconstruction of test video objects. Simulation results show that our vertex selection scheme can get less number of vertices for contour progressive approximation.

We test the performance of the proposed encoding scheme by encoding several widely used QCIF shape sequences: Weather, Children-Kids, and News. The performance is compared to that of the existing shape encoding schemes in [4], and [7]. Corresponding results are plotted in Fig. 6. Our proposed scheme can provide 25-60% gain in bit rate over the scalable encoding method in [7]. For some sequences, it can achieve 5-10% gain over conventional non-scalable vertex-based coding [3] in bit rate.

In conclusion, our proposed scalable shape coding method can achieve great improvement of compression performance by exploiting the geometrical knowledge of coarser levels and statistical entropy coding, although more computation is needed.

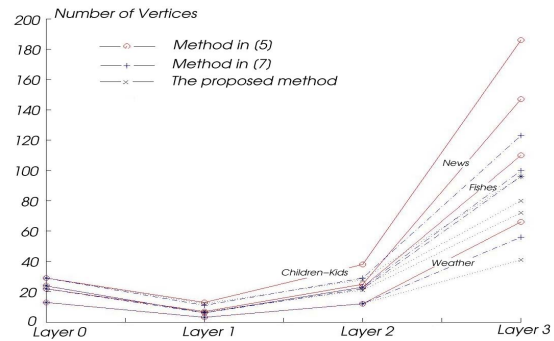


Fig. 5 Comparison of different vertex-selection methods

4. REFERENCES

1. N. Brady, F. Bossen, and N. Murphy, "Context-based arithmetic encoding of 2D shape sequences", in Special session on shape coding, ICIP'97, 1997
2. G. Melnikov and A. K. Katsaggelos, "A rate-distortion optimal scalable vertex based shape coding algorithm," *Proc. ICASSP-2000*, Turkey, June 5-9, 2000
3. C. Gu and M. Kunt, "Contour simplification and motion compensated coding", *signal processing: Image communications*, Vol.7, pp.279-296, Nov. 1995
4. K. J. O'Connell, "Object-adaptive vertex-based shape coding method", *IEEE Trans. on CSVT*, Vol.7, No.1, pp.251-255, Feb. 1997
5. P. Gerkin, "Object-based analysis-synthesis coding of image sequences at very low bit rates", *IEEE Trans. on CSVT*, Vol. 7, pp.251-255, Feb. 1997
6. G. Melnikov and A.K. Katsaggelos, "Shape approximation through recursive scalable layer generation," *Proc. of ICIP*, Canada, Sept 10-13, 2000
7. C.L. Bhanu Jordan, T. Ebrahimi, and M. Kunt "Progressive content based shape compression for retrieval of binary images ", *Computer Vision and Image Understanding*, Vol.71, No.2, pp.1126-1154, Aug. 1998
8. N. Ansari, E.J. Delp, "On detecting dominant points", *Pattern Recognition*, Vol. 24, No.5, pp.441-451, May 1991
9. F. Mokhtarian and Alan K. Mackworth, "A theory of multiscale, curvature-based shape representation for planar curves", *IEEE Trans. on PAMI*, Vol.14, No.8, pp. 789-805, Aug. 1992
10. Jae-won Chung, et.al., "A new vertex-based binary shape coder for high coding efficiency", *Signal Processing: Image Commun.*, Vo..15, pp.665-684, 2000

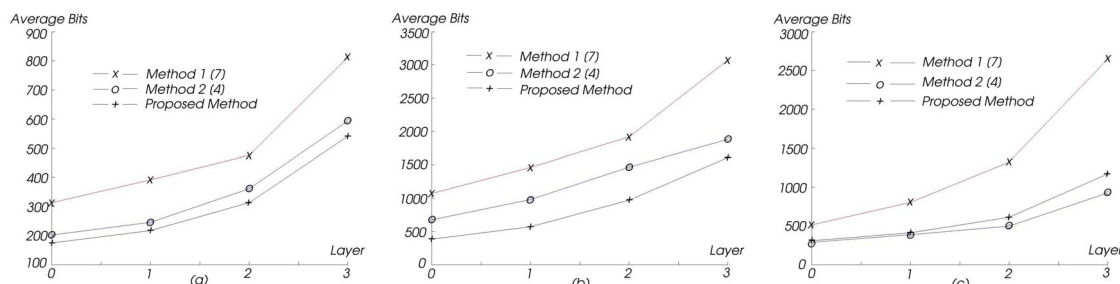


Fig.6. Comparison of various encoding methods for three sequences: (a) Weather, (b) Children-Kids, (c) News