2010 Data Compression Conference

Arbitrary Directional Edge Encoding Schemes for the Operational Rate-Distortion Optimal Shape Coding Framework *

Zhongyuan Lai[1], Junhuan Zhu[1], Zhou Ren[1], Wenyu Liu[1], *Member, IEEE*, Baolan Yan[2]
[1]*Dept. of Electron. and Inf. Eng., Huazhong Univ. of Sci. and Technol.*
[2]*Dept. of Inf. Technol., Huazhong Normal Univ.*
Email: {laizhy, zhujunhuan, renzhou200622}@gmail.com, liuwy@hust.edu.cn.

Abstract

We present two edge encoding schemes, namely 8-sector scheme and 16-sector scheme, for the operational rate-distortion (ORD) optimal shape coding framework. Different from the traditional 8-direction scheme that can only encode edges with angles being an integer multiple of π/4, our proposals can encode edges with arbitrary angles. We partition the digital coordinate plane into 8 and 16 sectors, and design the corresponding differential schemes to encode the short and the long component of each vertex. Experiment results demonstrate that our two proposals can reduce a large number of encoding vertices and therefore reduce 10%~20% bits for the basic ORD optimal algorithms and 10%~30% bits for all the ORD optimal algorithms under the same distortion thresholds, respectively. Moreover, the reconstruction contours are more compact compared with those using the traditional 8-direction edge encoding scheme.

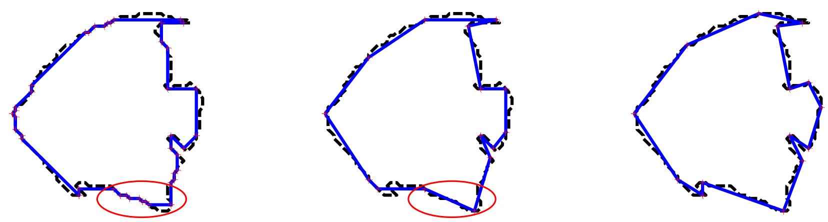
1 Introduction

Modern multimedia communication requires the convenience of video content access on an object basis to facilitate the applications of object-oriented storage, retrieval, editing and interaction. Given these application requirements, the video object needs to be described not only by texture but also by shape. Because of the inherent limitations of wireless bandwidth and rigorous power restriction of mobile terminals, this description can benefit significantly from faster and more efficient shape coding schemes [1-3].

There are two main frameworks for vertex-based shape coding. One is the top-down/bottom-up framework, which makes the vertex selection and encoding independent [4, 5]. And the other is the operational rate-distortion (ORD) optimal framework, which finds the rate-distortion (RD) optimality between the vertex selection and encoding. Many performance enhancement schemes used in the ORD optimal framework have been focused on the admissible vertex band (AVB) and the edge distortion measurements [6-14], but few have paid attention to the edge encoding. Their problem to be addressed is to find the optimal solution to the vertex selection in the rate-distortion sense in the condition that giving the 8-direction edge encoding scheme [6]. Thus the edge encoding scheme plays a vital role in the compression performance. However, the edge available for the 8-direction scheme should be restricted to intersect the horizontal axis in an angle that is an integer multiple of $\pi/4$. It has at least the following two shortcomings, as shown in Fig 1.

* This work received supports from National 863 project of China (grant No. 2007AA01Z223), National Natural Science Foundation of China (grant No. 60873127, 60903096, 60903172) and The National Science & Technology Pillar Program of China (grant No. 2006BAH02A24). The authors would like to thank Zhen Zuo and Tengyu Chen for fruitful discussion. The corresponding author is Wenyu Liu.

1068-0314/10 $26.00 © 2010 IEEE
DOI 10.1109/DCC.2010.10



(a) VN = 43, R = 339bits D = 2pels (b) VN = 15, R = 283bits D = 2pels (c) VN = 15, R = 242bits D = 2pels

Fig. 1. Comparisons of the edge encoding schemes embedded in the basic ORD optimal algorithm using accurate distortion measurement for both the shape coding and the reconstruction distortion measurement [10]. The original contour given by [11] is denoted in dashed while its polygonal reconstructions with the maximum distortion threshold= 2pels are denoted in solid. The encoding vertices are denoted by crosses. (a) Traditional 8-direction scheme, (b) proposed 8-sector scheme, and (c) proposed 16-sector scheme.

- The encoding edge could not describe the original object contour segment appropriately, since the trend of real contour segments may not be exactly toward these eight restricted directions.
- It may lead to a great number of selected vertices, called vertex number (VN), for the original contour reconstruction, which may need a lager number of encoding bits.

Based on the above analysis, we focus on designing the schemes that can encode the edges with arbitrary directions. We propose two edge encoding schemes, called the 8-sector scheme and the 16-sector scheme, to achieve this aim. We first partition the digital plane into 8 or 16 sectors, and then determine the short component and the long component according to the position of encoding vertex. Then we use the run-length scheme to encode the short component and the difference between the short and the long component employing their correlations. These schemes can be seamlessly embedded into any concrete ORD optimal algorithms to improve the RD performance and provide more compact contour representation.

The reminder of this paper is organized as follows. Section 2 introduces the ORD optimal shape coding framework, summarizes the related work and analyzes the traditional 8-direction edge encoding scheme in details. Section 3 and 4 present our proposed 8-sector and 16-sector edge encoding schemes, respectively. Full analyses of experiment results are presented in Section 5. Finally, some conclusions and future work is given in Section 6.

2 Related work

The goal of the ORD optimal shape coding algorithms in this paper is to find the object contour's approximate polygon that can be encoded by minimum rate under the specified distortions. Let $r(v_i, v_j)$ represent the encoding rate of edge $\overline{v_i v_j}$ and $R(v_i)$ represent the minimum rate to reach the admissible vertex point v_i from the source vertex v_0 via an approximation. Let $q(v_i)$ be the back pointer that is used to remember the optimal path. Fig. 2 shows a simple directed acyclic graph (DAG) for a polygon approximation. To find the shortest path through this graph that selects the optimal vertices, we use the algorithm below.

The previous work on performance enhancements of the ORD optimal framework focuses on two aspects. One is the AVB, which is used for vertex selection to improve its RD performance. Katsaggelos *et al.* firstly proposed the concept of admissible vertex set, and employ the fixed bandwidth for its calculation [1, 2].

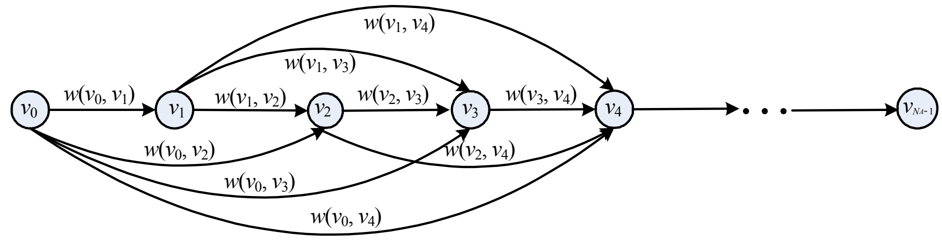


Fig. 2. A simple DAG for a polygon approximation. The weights are defined as the edge rates between two vertices.

Algorithm 1 *The ORD optimal shape coding algorithm*

Inputs: the admissible vertex band A and the distortion threshold $D_{\max}$

Output: the set of ordered vertices for object contour approximation $\{q_1,\ldots,q_{N_A-1}\}$

```
1)  R(v_0) = w(v_-1, v_0);
2)  for i = 1,…,N_A −1
3)      R(v_i) = ∞;
4)  end for
5)  for i = 0,…,N_A −2
6)      for j = i+1,…,N_A −1
7)          if d(v_i, v_j) ≤ D_max
8)              w(v_i, v_j) = r(v_i, v_j);
9)          else
10)             w(v_i, v_j) = ∞;
11)         end if
12)         if R(v_i) + w(v_i, v_j) < R(v_j)
13)             R(v_j) = R(v_i) + w(v_i, v_j);
14)             q(v_j) = v_i;
15)         end if
16)     end for
17) end for
```

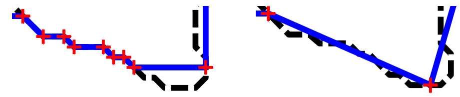
Sohel *et al.* then extended its bandwidth to be a variable with respect to its distortion bandwidth [7, 8], but only 3%~9% bits have been saved compared to the one with fixed bandwidth. The other is the edge distortion, which is used in step 7) in Algorithm 1 to improve its reconstruction quality. Schuster *et al.* summarized the two edge distortion categories, namely the absolute distance and the distortion band (DB), and designed two implementations, namely shortest absolute distance (SAD) and fixed DB [6]. For the absolute distance, Sohel *et al.* introduced the accurate distortion measurement for generic shape coding (ADMSC) from the geometrical viewpoint [9, 10] and then provided its fast approximate calculation strategy using chord length parameterization [11]. And Lai *et al.* presented the perceptual relevance measurement (PRM) from the psychological viewpoint [12]. And for the DB, Kondi *et al.* extended its bandwidth to be a variable with respect to the texture profile of the original video frame [13, 14].

However, little research has focused on the edge rate, which is used in Step 8) in Algorithm 1 to improve the RD performance. Almost all the enhanced algorithms employ the 8-direction edge encoding scheme, which is proposed by Schuster *et al.* in the year of 1998 [6]. In this scheme, the eight-connected chain code and the run-length encoding has been combined by representing the edge between two admissible pixels by an angle α and a run β, which forms the symbol (α,β), as illustrated in Fig. 3. For a given symbol (α,β), the first three bits indicate one of the eight possible values for α followed by $(\beta-1)$ zeros and a final “1” to encode the number of runs. Algorithm 2 shows how to calculate the rate of the encoding edge.

The above 8-direction edge encoding scheme is quite simple and easy to implement. However, its limitation can be clarified from two aspects. From the algorithmic aspect, note that the shorter the run, the less efficient this scheme is. Fig. 1 and 4 has shown that this scheme may lead to a great number of short edges. Therefore, it is a waste of bits for encoding a large number of edge angles. From the graphic aspect, note that the rate of the edge with its angle not a integer multiple of $\pi/4$ will be assigned to infinity. It may lead to a great number of weights in the DAG

(3, 1)	(2, 1)	(1, 1)
(4, 1)	V	(0, 1)
(-3, 1)	(-2, 1)	(-1, 1)

Fig. 3. Edge symbols of the eight closest neighbors of the vertex V



(a) $VN = 9$ (b) $VN = 2$

Fig. 4. Magnified portions in Fig. 1(a) and (b) enclosed by the ellipses

Algorithm 2 *The traditional 8-direction edge rate algorithm*

Inputs: coordinate differences between the vertices v_i, v_j of the encoding edge, denoted by (x, y)

Output: the number of bits needed for encoding the edge $\overline{v_i v_j}$, denoted by $r(v_i, v_j)$

1) $\alpha = \text{angle}(x, y)$;
2) $\beta = \max\{|x|, |y|\}$;
3) if $\text{mod}(\alpha, \pi/4) = 0$
4) $\quad r(v_i, v_j) = 3 + \beta$;
5) else
6) $\quad r(v_i, v_j) = \infty$;
7) end if

to be infinity, which yields much longer shortest path from the source node v_0 to the destination node v_{N_t-1}. Therefore, the traditional 8-direction scheme is inefficient, which motivates us to relax the direction restriction. The next two sections follow this idea.

3 The 8-sector edge encoding scheme

A simple and intuitive approach to encode the arbitrary directional edges is to directly encode the x- and y-coordinate of that edge by run-length codes. However, it does not make good use of the correlation between the x- and y-coordinate and may lead to a waste of encoding bits. Here, we represent each edge by a sector number, a short component and a long component. We use the reference of [15] and define the sector number as: sector 0 is the set of (x, y) such that $0 \le y < x$, sector 1: $0 < x \le y, \ldots,$ and sector 7: $0 < -y \le x$, as illustrated in Fig. 5. The short component is the smaller component of the edge's x and y components, whose value is denoted by α. And the long component is the larger component of the edge's x and y components, whose value is denoted by β. The long component should be encoded differentially, providing a further reduction in the number of bits used. The procedure to encode each edge is as follows.

1) Determine the sector number;
2) Encode the sector number using three bits fix-length code (FLC);
3) Determine the *short component* and the *long component* according to the sector number, and calculate both α and β;
4) If $\alpha = 0$, it is impossible that $\beta = 0$ at the same time. Thus, we use "1" to encode the *short component* and use $(\beta - 1)$ zeros and a final "1" to encode the *long component*;
5) If $\alpha > 0$, we first use α zeros and a final "1" to encode the *short component*, and then use $(\beta - \alpha)$ zeros and a final "1" to encode the *long component*.

Algorithm 3 shows how to calculate the edge rate in 8-sector scheme. To further illustrate the potential bits it saves, we consider a particular edge (6, 3), as shown in Fig. 5. For the traditional 8-direction edge encoding scheme, at least two sub-edges (illustrated by two dashed edges in Fig. 5) with the smallest sum of runs 6 were needed no matter what the contour segment would be. In this case, at least 12 bits were needed for the contour segment approximation. However, 11 bits were needed if we employ the proposed 8-sector scheme and at least 1 bit could be saved.

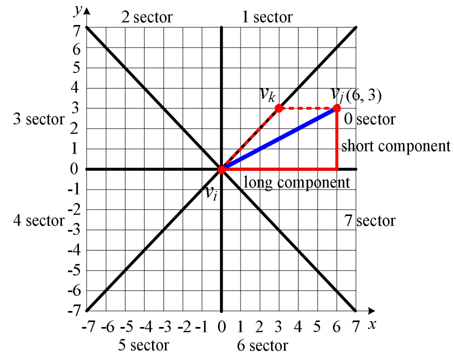


Fig. 5. Definition of the sector number, the short component and the long component in 8-sector edge encoding scheme

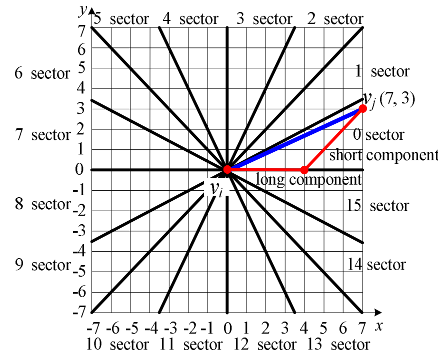


Fig. 6. Definition of the sector number, the short component and the long component in 16-sector edge encoding scheme

Algorithm 3 *The proposed 8-sector edge rate algorithm*

Inputs: coordinate differences between the vertices v_i, v_j of the encoding edge, denoted by (x, y)

Output: the number of bits needed for encoding the edge $\overline{v_i v_j}$, denoted by $r(v_i, v_j)$

1) $\alpha = \min\{|x|, |y|\}$;
2) $\beta = \max\{|x|, |y|\}$;
3) if $\alpha = 0$
4) $\quad r(v_i, v_j) = 4 + \beta$;
5) else
6) $\quad r(v_i, v_j) = 5 + \beta$;
7) end if

4 The 16-sector edge encoding scheme

The above 8-sector scheme encodes the given edge's x- and y-coordinate, respectively. These two coordinates are perpendicular to each other. As a result, the maximum encoding coordinate may be much longer since the direction of at least one coordinate is not similar with that of the encoding edge. We also take a particular edge (6, 3) as an example. In this case, the y-coordinate makes little contribution to the length of this edge. However, we still need 4 bits for it as the short component and 4 bits for the long component, which leads to 8 bits in total. If we make both directions of the short and the long component similar with that of the encoding edge, for example, the horizontal and the $\pi/4$ direction (illustrated by two dashed edges in Fig. 5), the number of bits needed is reduced to 4 bits and 1 bit, respectively. In this case, 5 bits are needed in total and 3 bits have been saved. However, it needs to indicate whether the horizontal or the $\pi/4$ directional component is the short component, which needs 1 bit for further direction indication.

The above analysis motivates us to represent each edge by a sector number, a short component, and a long component as follows. The sector numbers are defined as: sector 0 is the set of (x, y) such that $0 \le 2y < x$, sector 1: $y < x \le 2y, \ldots$, and sector 15: $0 < -2y \le x$, as illustrated in Fig. 6. The short component is the smaller component of the two adjacent octant directional components, whose value is denoted by α. And the long component is the larger component of those components, whose value is denoted by β. The long component should also be encoded differentially, providing a further reduction in the number of bits used. The procedure to encode each edge is as follows.

1) Determine the sector number;
2) Encode the sector number using four bits FLC;
3) Determine the *short component* and the *long component* according to the sector number, and calculate both α and β;
4) If $\alpha = 0$, it is impossible that $\beta = 0$ at the same time. Thus, we use "1" to

encode the *short component* and use $(\beta-1)$ zeros and a final "1" to encode the long component;

5) If $\alpha>0$, we first use α zeros and a final "1" to encode the *short component*, and then use $(\beta-\alpha)$ zeros and a final "1" to encode the *long component*.

Algorithm 4 *The proposed 16-sector edge rate algorithm*

Inputs: coordinate differences between the vertices v_i, v_j of the encoding edge, denoted by (x, y)

Output: the number of bits needed for encoding the edge $\overline{v_i v_j}$, denoted by $r(v_i, v_j)$

1) $\alpha=\min\{\| x|-|y\|,\min\{|x|,|y|\}\}$;
2) $\beta=\max\{\| x|-|y\|,\min\{|x|,|y|\}\}$;
3) if $\alpha=0$
4) $r(v_i, v_j) = 5 + \beta$;
5) else
6) $r(v_i, v_j) = 6 + \beta$;
7) end if

Algorithm 4 shows how to calculate the edge rate in 16-sector scheme. To further illustrate the potential bits it saves, we consider a particular edge (7, 3), as shown in Fig. 6. For the 8-sector edge encoding scheme, 12 bits are needed. However, 10 bits are needed if we employ the proposed 16-sector scheme and 2 bits can be saved.

5 Experiment results

To both qualitatively and quantitatively analyses the performance of our proposed two edge encoding schemes compared with the traditional 8-directiion scheme, the ORD optimal shape coding algorithms including the AVB and the sliding window (SW) strategy were implemented in Matlab 7.1 (The Mathworks Inc.). The nomenclature used in this section is summarized in Table 1 [10].

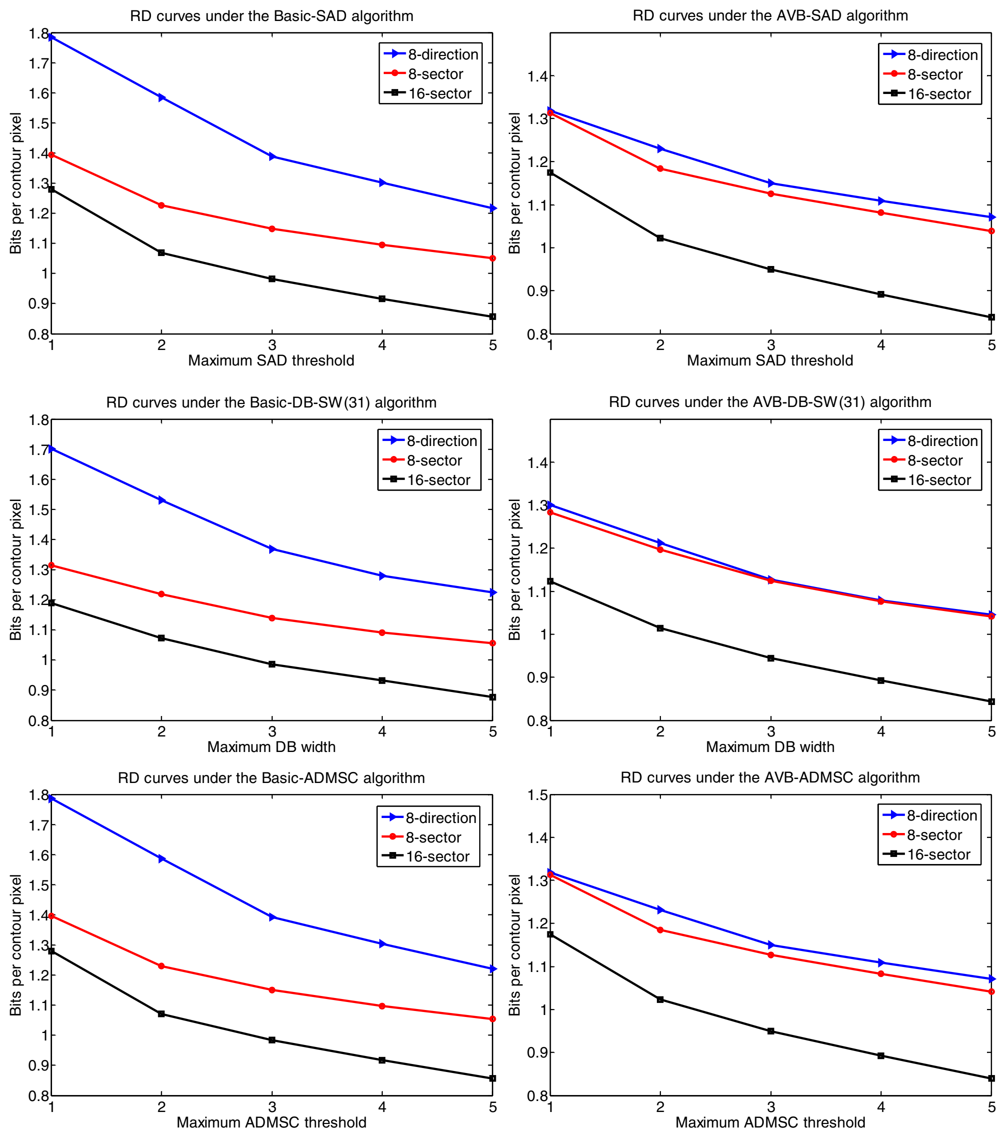
Tab. 1. Nomenclature used for the different ORD optimal shape coding algorithms

Label	Denotation
Basic-SAD	ORD optimal algorithm using SAD with neither AVB nor SW
Basic-DB-SW(*x*)	ORD optimal algorithm using DB considering SW of length *x*-pel but not AVB
Basic-ADMSC	ORD optimal algorithm using ADMSC with neither AVB and SW
AVB-SAD	ORD optimal algorithm using SAD considering AVB
AVB-DB-SW(*x*)	ORD optimal algorithm using DB considering both AVB and SW of width *x*-pel
AVB-ADMSC	ORD optimal algorithm using considering AVB

5.1 The RD performance assessments

For the RD performance assessments, four MPEG-4 test sequences – *Weather*.qicf (spatial resolution 176×144, 100 frames), *News*.qcif (176×144, 100), *Stefan*.sif (352×240, 100) and *Children-Kids*.sif (352×240, 100) have been used. All the sequences are available at [16] and all the results represent averages over these 100 frames.

Fig. 7 shows the cumulative RD curves under the different ORD optimal algorithms. For the 8-sector scheme, there are 14% ~ 23% bits reduction for the basic algorithms while no more than 4% bits reduction for the AVB based algorithms. For 16-sector scheme, there are 22% ~ 29% and 11% ~ 22% bits reduction for the basic algorithms and the AVB based algorithms, respectively, compared with the traditional 8-direction scheme. Not only dose it indicate how much the RD performance enhances, but also it points out how important the role of the edge encoding schemes play in the ORD optimal framework.



(a) (b)

Fig. 7 RD comparisons under (a) the basic and (b) the AVB based ORD optimal algorithms

5.2 The reconstruction assessments

For the reconstruction assessments, the *second kid* in the 17th frame of the *Children-Kids* sequence, which has been employed in both [2] and [3] as a standard object, has been used for test. The distortion metric given in [10], called the ADMSC, denoted by D, has been utilized for reconstruction distortion measurement.

Fig. 8 shows the reconstruction polygons under the basic ORD optimal algorithms. Considering the fifth column as an example, we see that the traditional 8-direction scheme needs 109 vertices for the original object contour approximation, but both of the proposed 8-sector and 16-sector schemes only need 37 vertices. As a result, the traditional one needs 671 bits while our two proposals need 521 and 469 bits for contour encoding, and therefore 150 and 202 bits have been saved, respectively, under the condition that D = 1pel. One may argue the comparison availability in column 1 ~ 4, since the resulting object ADMSCs are larger than the prescribed threshold when we choose SAD or DB for our edge distortion calculations. This

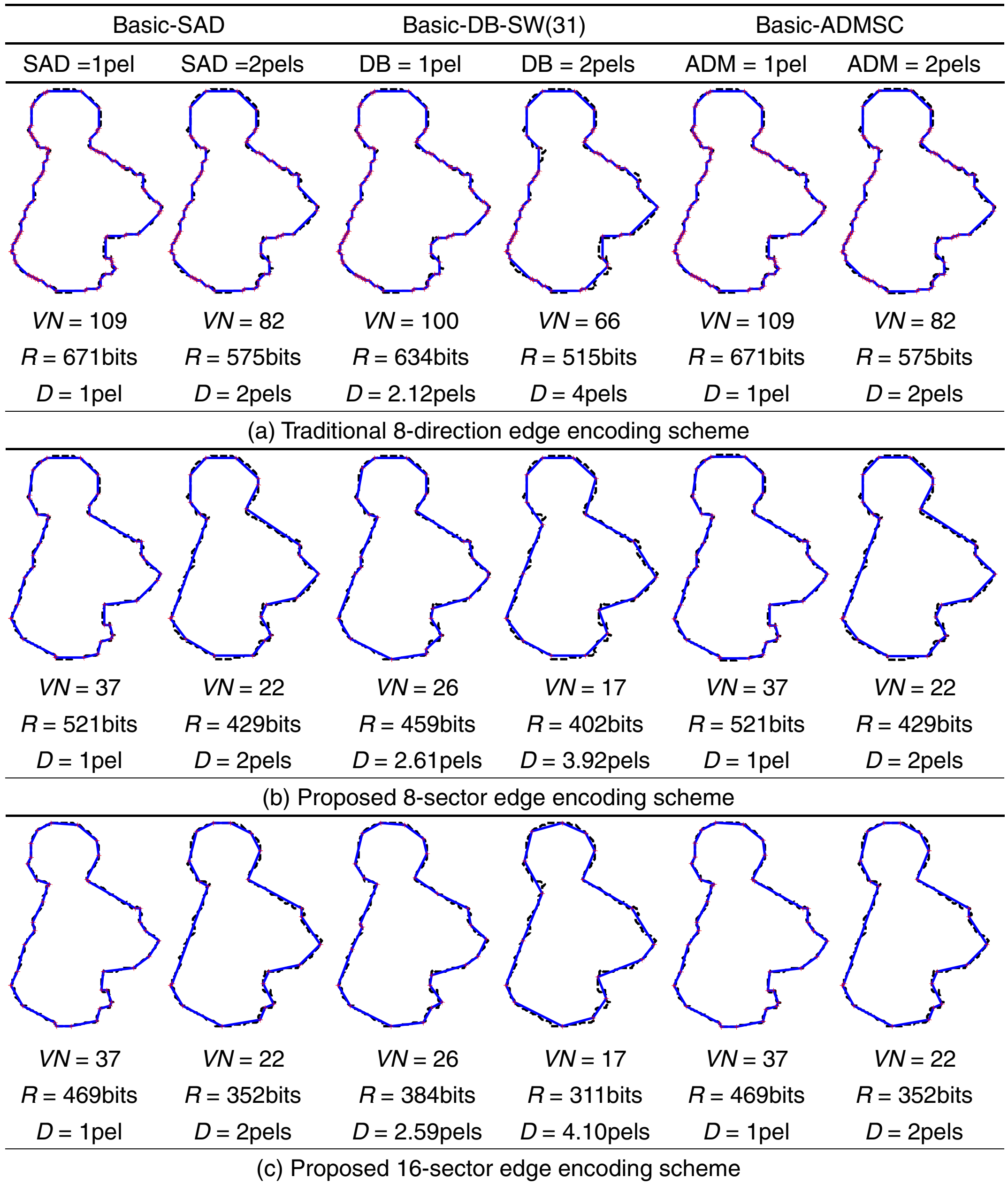


Fig. 8. Reconstruction comparisons under the basic ORD optimal algorithms

phenomenon has been discovered by Sohel *et al.* in the year of 2006 and the ADMSC has been introduced to address this issue [10]. It is interesting that their ADMSCs are similar in the digital sense when the prescribed SAD or DB threshold is the same, since their distortion excess mechanisms are the same. So it guarantees the availability to compare the compression performance of the SAD and DB based algorithms under the same prescribed thresholds. Moreover, we see that the polygons reconstructed by our proposals are quite compact and have stronger ability to reflect the characteristics of the original object contour, since the relatively smooth contour segment are easier to be approximate by the edge so that the turns of the reconstruction polygon are more likely to be the contour corners. Thus they can benefit the efficiency of the successive applications such as shape retrieval and boundary editing.

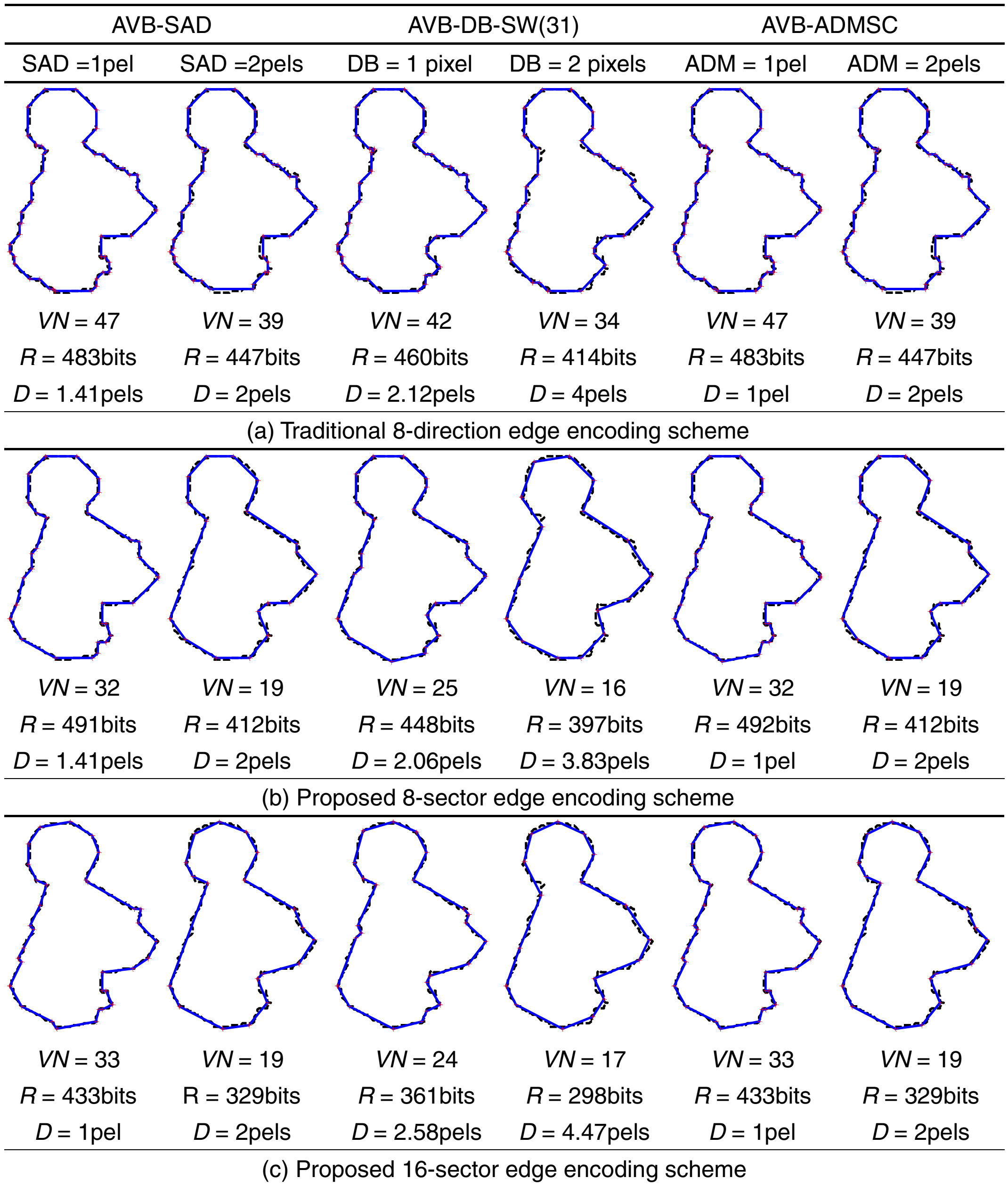


Fig. 9 Reconstruction comparisons under the AVB based ORD optimal algorithms with AVB width = 1pel.

Fig. 9 shows the reconstruction polygons under the AVB based ORD optimal algorithms. Considering the fifth column as an example, we see that the traditional 8-direction scheme needs 47 vertices for the original object contour approximation, but our proposed 8-sector and 16-sector schemes only need 32 and 33 vertices. Thus, the traditional one needs 483 bits and our two proposals need 492 and 433 bits, respectively, under the condition that D = 1pel. The reason why the 8-direction scheme needs a bit less bits (8 bits) than the 8-sector one is that the introduced AVB extends the vertex selection region so that it can make longer edge satisfying both the distortion and direction constraints. At the same time, the 16-sector scheme still reveals high encoding efficiency. In addition, for the subjective assessments, the similar conclusions can be drawn as in Fig. 8.

6 Conclusions and future work

While traditional research of the ORD optimal shape coding framework has focused on the AVB expansions and the edge distortion measurements, little research has paid attention to the edge encoding schemes. This paper has shown the inherent limitations of traditional 8-direction edge encoding scheme that cannot well encode the original object contour segment and introduce the two arbitrary directional edge encoding schemes to address this issue. Our experiments have demonstrated that our proposed 8-sector and 16-sector schemes can reduce a large number of encoding vertices and save 10% ~ 20% bits for the basic ORD optimal algorithms and 10% ~ 30% bits for all the ORD optimal algorithms, respectively. Moreover, the reconstruction contours are more compact compared with those using the traditional 8-direction edge encoding scheme.

We can make at least two extensions in our future work. Firstly, we can employ the idea of arbitrary direction to encode the ordered control points of the B-spline based ORD optimal framework, since the traditional one has also been restricted to the eight directions. Secondly, we can design the fix-length codes for edge encoding if we apply the SW strategy, since sometimes the length of the encoding edge is long.

References

[1] G. M. Schuster and A. K. Katsaggelos, *Rate-distortion Based Video Compression: Optimal Video Frame Compression and Object Boundary Encoding*. Norwell, MA: Kluwer, 1997.

[2] A. K. Katsaggelos, L. P. Kondi, F. W. Meier, J. Ostermann, and G. M. Schuster, "MPEG-4 and rate-distortion-based shape-coding techniques," *Proc. IEEE*, vol. 86, no. 6, pp. 1126–1154, Jun. 1998.

[3] G. M. Schuster, G. Melnikov and A. K. Katsaggelos, "Operationally optimal vertex-based shape coding," *IEEE Signal Process. Mag.* vol. 15, no. 6, pp. 91–108. Nov. 1998.

[4] J. Chung, J. Lee, J. Moon, and J. Kim, "A new vertex-based binary shape coder for coding efficiency," *Signal Process.: Image Commun.*, vol.15, no.7–8, pp. 665–684, May 2000.

[5] C.-M. Kuo, C.-H. Hsieh, and Y.-R. Huang, "A new adaptive vertex-based binary shape coding technique," *Image Vision Comput.*, vol. 25, no. 1, pp. 863-872, Jun. 2007.

[6] G. M. Schuster and A. K. Katsaggelos, "An optimal polygonal boundary encoding scheme in the rate distortion sense," *IEEE Trans. Image Process.*, vol. 7, no. 1, pp. 13-26, Jan, 1998.

[7] F. A. Sohel, L. S. Dooley, and G. C. Karmakar, "Variable width admissible control point band for vertex based operational-rate-distortion optimal shape coding algorithms," *IEEE Int. Conf. Image Process.*, pp: 2461-2464, Oct. 2006.

[8] F. A. Sohel, L. S. Dooley, and G. C. Karmakar, "New dynamic enhancements to the vertex-based rate-distortion optimal shape coding algorithm," *IEEE Trans. Circ. Sys. Video Tech.*, vol. 17, pp. 1408-1413, Oct. 2007.

[9] F. A. Sohel, L. S. Dooley, and G. C. Karmakar, "A modified distortion measurement algorithm for shape coding," in *Proc. 3rd Workshop Internet, Telecommun.Signal Proces.*, Adelaide, Australia, Dec. 2004.

[10] F. A. Sohel, L. S. Dooley, and G. C. Karmakar, "Accurate distortion measurement for generic shape coding," *Pattern Recognition Letters*, vol, 15, pp: 133-142, Jan. 2006.

[11] F. A. Sohel, G. C. Karmakar, and L. S. Dooley, "Fast distortion measurement using chord-length parameterization within the vertex-based rate-distortion optimal shape coding framework," *Signal Process. Letters*, vol. 14, pp: 121-124, Feb. 2007.

[12] Z. Lai, W. Liu, and Y. Zhang, "Perceptual relevance measure for generic shape coding," in *Proc. Data Compression Conference*, Snowbird, Utah, U.S.A., Mar. 2009, pp: 453.

[13] L. P. Kondi, F. W. Meier, G. M. Schuster, and A. K. Katsaggelos, "Joint optimal object shape estimation and encoding," in *Proc. SPIE Conf. Visual Commun. and Image Proc.*, Jan. 1998, pp. 14-25.

[14] L. P. Kondi, G. Melnikov, and A. K. Katsaggelos, "Joint optimal object shape estimation and encoding," *IEEE Trans. Circ. Syst. Video Tech.*, vol. 14, pp: 528-533, Apr. 2004.

[15] K. J. O'Connell, "Object-adaptive vertex-based shape coding method," *IEEE Trans. Circ. Sys. Video Tech.*, vol. 7, pp. 252–255, 1997.

[16] ftp://ftp.tnt.uni-hannover.de/pub/mpeg/mpeg4_masks/.