Sliding-Window Designs for Vertex-Based Shape Coding

Ferdous A. Sohel, *Member, IEEE*, Gour C. Karmakar, Laurence S. Dooley, *Senior Member, IEEE*, and Mohammed Bennamoun

***Abstract*—Traditionally the *sliding window* (SW) has been employed in vertex-based *operational rate distortion* (ORD) optimal shape coding algorithms to ensure consistent distortion (quality) measurement and improve computational efficiency. It also regulates the memory requirements for an encoder design enabling regular, symmetrical hardware implementations. This paper presents a series of new enhancements to existing techniques for determining the best SW-length within a *rate-distortion* (RD) framework, and analyses the nexus between SW-length and storage for ORD hardware realizations. In addition, it presents an efficient bit-allocation strategy for managing multiple shapes together with a *generalized adaptive SW* scheme which integrates localized curvature information (*cornerity*) on contour points with a bi-directional spatial distance, to afford a superior and more pragmatic SW design compared with existing adaptive SW solutions which are based on only cornerity values. Experimental results consistently corroborate the effectiveness of these new strategies.**

***Index Terms*—Image processing, shape coding, sliding window.**

I. Introduction

ADVANCES in shape-based, object-oriented video coding [1]–[7] are increasingly facilitating more efficient retrieval, manipulation and interactive functionality for both natural and synthetic sequences. The pursuit for greater coding efficiency coupled with the inherent bandwidth limitations of existing communication technologies mean a large number of diverse applications will significantly benefit from more effectual shape coding strategies. Examples of such applications include medical imaging and patient monitoring, video-on-demand and Internet streaming of multimedia content, biometric authentication systems, mobile video transmissions for handheld devices and hyperlinked video/television.

Shape coders have evolved into two major classes [6]: 1) *bitmap-based* which encode every pixel within the shape and 2) *contour-based* which focus on only the object shape outline. The review of shape coding techniques in [6] concluded that the vertex-based polynomial shape coding framework is optimal in an *operational-rate-distortion* (ORD) sense. Both polygon and quadratic B-spline, based shape encoding strategies were deployed, providing the foundation for a suite of algorithms [6], [8]–[10], which had as their explicit aim, that for some prescribed distortion, a shape contour would be optimally encoded in terms of the number of bits, by selecting the set of *control points* (CP) that incurred the lowest bit rate and vice versa. Aside from rate and distortion, the *sliding window* (SW) is a key design parameter in the vertex-based shape coding framework, as its length directly impacts upon the bit-rate.

Katsaggelos *et al.* introduced the SW to force the encoder to track the shape boundary so as to avoid the trivial solution problem [6], as well as to improve computational speed by limiting the CP search space. The SW confines the search space for the next CP to only those candidate points lying within the window, thereby compromising global optimality in a bit-rate sense and defining an implicit lower bound on the number of CP and hence, the requisite number of bits to encode a particular shape. A corollary is this implicitly defines the storage requirement for any hardware encoder implementation. One of the main reasons for the vertex-based shape coder not being adopted by MPEG4 was its asymmetric memory access requirements [6]. This paper investigates how by automatically determining the SW-length, the storage can effectively be managed. From both a computational and memory efficiency perspective, a narrow SW is desirable though decreasing the SW-length conversely increases the overall bit-rate, because at least one CP must be selected from within the window. The selection of an appropriate SW-length is therefore of paramount design importance to achieving better admissible bit-rate utilization together with minimal memory and computational complexity. Previous SW-length selection studies [10], [11] have concentrated on analyzing the *rate-distortion* (RD) performance and computational speed, with the resultant storage needs not being considered. This paper reviews these various SW-length strategies together with a corresponding memory requirement analysis, prior to presenting a series of new enhancements which concomitantly improve encoder performance from a RD, computational and memory standpoint.

The remainder of the paper is organized as follows: Section II provides a brief review of vertex-based ORD optimal shape coding algorithms with a particular design focus towards the influence of the SW-length, while Section III details the underlying theoretical principles of the proposed SW-length selection strategies. Section IV presents a comprehensive analysis of the empirical results, with some concluding remarks provided in Section V.

Manuscript received September 06, 2011; accepted December 19, 2011. Date of publication January 26, 2012; date of current version May 11, 2012. This work was supported in part by a University of Western Australia (UWA) Postdoctoral Fellowship and an ARC discovery grant (DP0771294). The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Ketan Mayer-Patel.

F. A. Sohel and M. Bennamoun are with the School of Computer Science and Software Engineering, The University of Western Australia, Crawley, Australia (e-mail: Ferdous.Sohel@csse.uwa.edu.au; m.bennamoun@csse.uwa.edu.au).

G. C. Karmakar is with the Gippsland School of Information Technology, Monash University, Churchill, Australia (e-mail: Gour.Karmakar@infotech.monash.edu.au).

L. S. Dooley is with the Department of Communication and Systems, The Open University, Milton Keynes, U.K. (e-mail: L.S.Dooley@open.ac.uk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2011.2182507

II. Review of the Vertex-Based ORD Framework Relating to Sliding Windows

Let $B = \{b_0, b_1, \ldots, b_{N_B-1}\}$ be an ordered set of points that defines the contour, where N_B is the total number of contour points and $b_0 = b_{N_B-1}$ for a closed contour. If $S = \{s_0, s_1, \ldots, s_{N_S-1}\}$ is an ordered set of CP used to approximate B, where N_S is the total number of polygon-edges in S and in the basic framework $S \subseteq B$. As a representative example, the ORD polygon-based shape coding algorithm [9] to determine the optimal S for contour B for a prescribed admissible distortion pair $T_{\max}$ and $T_{\min}$ is formalized in Algorithm 1 below.

Algorithm 1: The Basic Polygon-Based ORD Optimal Shape Coding Framework

Input: B—the contour; $T_{\max}, T_{\min}$—the admissible distortion pair.

Variables: $MinRate\,(b_i)$—current minimum bit-rate to encode up to vertex b_i from b_0; $pred\,(b_i)$—preceding CP of b_i.

Output: S—the ordered set of CP approximating B.

1) Initialize $MinRate\,(b_0)$ with the total bits required to encode the first contour point b_0;

2) Set $MinRate\,(b_k)\,,\ 0 < k < N_B$ to infinity;

3) FOR each vertex $b_i,\ 0 \le i < N_B - 1$

4) FOR each vertex $b_j,\ i < j \le N_B - 1$

5) Calculate the edge-distortion; $dist\,(b_i, b_j)$

6) Determine bit-rate $r\,(b_i, b_j)$;

7) Assign edge-weight $w\,(b_i, b_j)$;

8) IF $((MinRate\,(b_i) + w\,(b_i, b_j)) < MinRate\,(b_j))$

9) $MinRate\,(b_j) = MinRate\,(b_i) + w\,(b_i, b_j)$;

10) v$pred\,(b_j) = b_i$;

11) Obtain S with properly indexed values from $pred$;

The kernel element of this framework is the construction of a weighted *directed acyclic graph* (DAG), with the ensuing bit-rate minimization process being formulated as a shortest-path search problem (Steps 3–4). Since the vertices of the contour are ranked, the DAG is formed such that a directed edge is defined as (b_i, b_j), only if $i < j$. The weight/cost of the edge is determined (Step 7) based on the edge distortion (Step 5) and the number of bits required (bit-rate r) to differentially encode the edge (b_i, b_j), given that the path from the source vertex to vertex b_i has already been coded (Step 6). If the edge distortion is within the admissible distortion limit, the weight equals the bit-rate; otherwise, it is infinitely large. It should be noted that the admissible distortion limits are calculated according to the algorithm in [8] using the admissible distortion pair $T_{\max}$ and $T_{\min}$ and shape curvatures. The distortion measurement technique adopted therefore exerts a significant influence upon the performance of the ORD framework. A number of distortion measurement techniques have been proposed including the *distortion band* (DB) [6], *tolerance band* (TB) [8], *perceptual relevance measure* [12], and *accurate distortion measurement for shape coding* (ADMSC) [13], with each algorithm exhibiting different features, degrees of accuracy, and computational efficiency. A comprehensive review of distortion metrics and measurement techniques is furnished in [14]. In the original (Algorithm 1) ORD framework, the DAG was formed using just contour points, though this was subsequently augmented by adopting the popular concept of a *fixed-width admissible control point band* (FCB) to enhance RD performance [6]. The choice of the differential vertex encoding technique employed in Step 6 has a significant influence on the overall RD performance of the encoder, so this particular coding scheme will now be briefly examined.

A combination of *Chain Code* (CC) [4] and *Run Length* (RL) codes [9], [15] has traditionally been used (details in [9]), with the direction of vertex b_j being determined with respect to the already encoded vertex b_i, on an eight-connected grid. This can be encoded using a 3-bit *fixed length code* (FLC) or alternatively a 2-bit *improved orientation-based direction* technique [9]. The distance between these is then calculated for the RL and coded using FLC, *variable length codes* (VLC), or the prefix-free *logarithmic codes* (LC), which have the general form $[Y\ X]$, where Y is the length of X. While the example in [9] used 2 bits for Y giving a maximum of 15 *codewords*, as will be shown in Section III-A, by allocating 3 bits to Y this can be extended to 255 *codewords*. It is emphasized that the RL and the SW-length are measured differently, with the former being the absolute distance between two contour points (potential CP) in the Cartesian coordinate system. In contrast, the SW-length is the difference (in CC format) between the indices of two points, so the SW not only must be at least as long as the RL, but of sufficiently length to accommodate any RL supported by the encoding scheme.

Due to its impact on both DAG formation and distortion measurement, the SW significantly influences shape encoder performance as will now be analyzed.

A. Sliding Window Concept

The main impetus for adopting the SW in vertex-based shape coding [6] was to overcome the aforementioned trivial solution problem. When encoding a shape contour using either the DB or TB as the distortion metric, occasionally only a little part of the contour is actually coded. Large portions of the contour may therefore not be coded, despite the encoder believing the entire contour has been correctly coded within admissible distortion bounds. Fig. 1(a) illustrates this predicament, where only a small part of the *Neck* contour of the *MissAmerica*.qcif sequence has been accurately coded (the 1st, 2nd, and the last contour points are encoded). The reason for this anomaly is the way the DB/TB measures distortion when the encoder seeks the optimal path between the first and last contour points [6]. The introduction of an SW resolves this problem by restricting the search space for the next CP to within the window rather than up to the end contour point, thereby avoiding trivial solutions as illustrated in Fig. 1(b), where an SW-length of 8 pixels is employed. For the 1st contour point in the DAG, the SW only allows the encoder to consider the following L (eight) points on the contour. This holds true for the entire shape, so the encoder cannot bypass

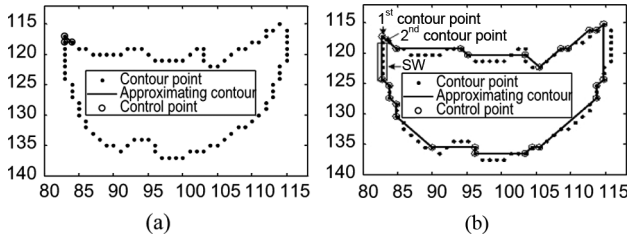

Fig. 1. Example of the trivial solution problem in the vertex-based shape coding algorithm using DB/ TB for the *Neck* contour of frame #31 of the *MissAmerica*.qcif sequence (a) without an SW and (b) using an SW ($L = 8$ pixels)—the next CP must be selected from the contour points within the SW (the horizontal and vertical axes, respectively, represent the Cartesian X and Y axes).

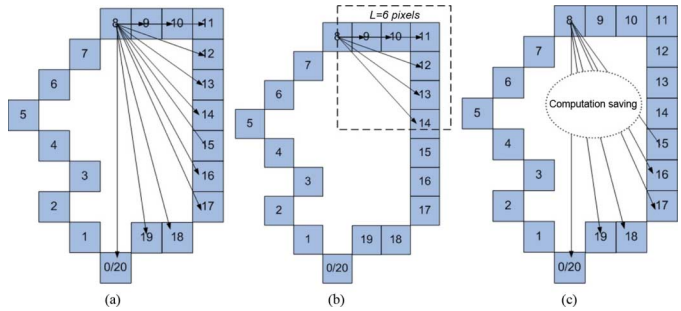

Fig. 2. Computational complexity reduction using an SW for contour point (only the indices are shown in the figures). (a) All links in DAG are searched without an SW. (b) Links only within the SW are searched. (c) Computational saving in (b) compared with (a).

any portion of the shape without upholding the admissible distortion limits. In the unlikely circumstances of an SW being so long it accommodates the entire shape and the distortion measurement is erroneous, a trivial solution can occur. To eliminate this risk, the ADMSC algorithm [13] should be used to guarantee accurate distortion measurement. Furthermore, the length restriction upon the encoder's search space concomitantly reduces the computational time overhead for coding as will be analyzed shortly. Interestingly, because a CP must now be selected from within the window, applying an SW compromises the global optimality of the encoder.

B. Influence of SW on the Computational Complexity

To appreciate the influential role of the SW, consider the computational complexity of Algorithm 1. The overall time requirement depends on the two loops in Steps 3 and 4, and the distortion measuring technique (Step 5). Using ADMSC takes $O(N_B)$ time for polygonal encoding, so Algorithm 1 incurs $O\left(N_B^3\right)$ computational time. Introducing a window of length L in Step 4 limits the range of j to $i < j \leq \min\{(i+L),(N_B-1)\}$, so the overall complexity becomes $O\left(N_B L^2\right)$, since in the worst case the Step 4 loop is executed L instead of N_B times. The distortion measurement process in Step 5 similarly has $O\left(L^2\right)$ complexity. The corresponding complexities for quadratic B-Spline based algorithms are $O\left(N_B L^4\right)$ and $O\left(N_B^5\right)$, depending upon whether the SW is either employed or not, so the overall computational expenditure is always lower provided $L < N_B$. This cost saving can be intuitively explained by means of the example in Fig. 2, where an arbitrary contour point b_8 is considered. When the SW is not employed [Fig. 2(a)], Algorithm 1 checks all the links connected to the contour points with a higher index than itself for the shortest possible path using the RD constraints, giving a total of 12. In contrast, when an SW of length 6 pixels is used [Fig. 2(b)], only the links to the next six vertices are checked for the shortest path, so affording a 50% saving, i.e., six links [Fig. 2(c)].

C. Influence of SW on Bit-Rate

The SW forces the encoder to select the next CP from only those points associated with the next L contour points, so in Fig. 2(b), a CP must be selected from points b_9 and b_{14} inclusive, while in Fig. 2(a) the encoder has the option of selecting a CP from any of points b_9 to b_{20}. For a small SW-length, the number of CP required to encode the same shape is large which is reflected in a higher bit-rate. The overall effect of the SW-length

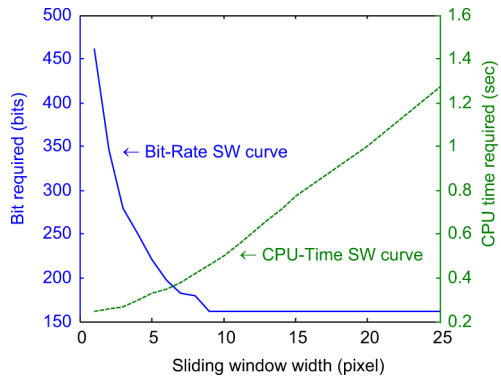

Fig. 3. Effect of SW-length on bit-rate and computational (CPU) time for $T_{\max} = T_{\min} = 1$ pixel on the *Neck* contour of frame #1 of the *MissAmerica*.qcif sequence.

upon the requisite bit-rate and computational time requirement for the *basic* ORD shape coding algorithm is illustrated in Fig. 3, for $T_{\max} = T_{\min} = 1$ pixel on the *Neck* contour of frame #1 of the *MissAmerica*.qcif sequence. As with all the various experiments discussed in Section IV, this algorithm was implemented in Matlab and run on a 2.67-GHz, Pentium-4 processor, with 4 GB of random access memory under a Windows7 operating system. The graph reveals smaller SW lengths incur lower computational times but a higher bit-rate. Furthermore, there is a decreasing rate of gain in bit-rate as the window length grows. For instance, as the SW-length increases from one to two pixels, the bit-rate reduction is much higher than when it is extended beyond 10 pixels where the gain becomes zero. Predictably the computational times increase commensurately with SW-length, so the respective times for $L = 1$ pixel and $L = 15$ pixels are 100 ms and 625 ms.

D. Influence of SW on Memory

Since the encoder chooses the next CP from just L possible contour points, it implicitly regulates the amount of memory needed for a hardware realization. The original vertex-based shape encoder was innately inflexible and asymmetric in terms of its memory access requirement, which was the primary reason it was not chosen for MPEG4 [6]. This asymmetry is caused by the size of the DAG which has $N_B \times L$ links when an SW is applied compared with $N_B \times (N_B - 1)$ links when it is not. The on-cache memory for the vertex-based shape encoder requires two blocks to store: 1) the program and 2) assorted

subroutines, particularly for distortion measurement (Step 5). The storage overhead for the second block depends on the distortion metric applied and the SW-length, so using ADMSC as the distortion measurement algorithm incurs $O(N_B)$ storage, where N_B is an arbitrary number, compared with $O(L)$ when an SW is used. The $O(N_B)$ reflects that the first memory block size is governed by array *MinRate()*, while the on-cache data structure size of *MinRate()* when an SW is applied is constrained to $O(L)$. So instead of two arbitrarily-sized blocks, the coder can be designed with two $O(L)$ memory blocks, with in general $L \leq 16$ pixels, so providing an elegant hardware encoder implementation in for example, a field programmable gate array (FPGA).

In summarizing, SW-length selection provides a trade-off mechanism between bit-rate and memory requirement on the one side and computational efficiency on the other. Smaller SW-lengths are desirable for encoding and memory efficiency, though this is offset by a higher overall bit-rate. Judiciously choosing the most appropriate SW-length is thus important for concomitantly achieving better admissible bit-rate utilization and reduced complexity. A review of existing solutions [10], [11] will be provided as the precursor to introducing a series of new enhancements allied with a corresponding analysis of the SW performance from a memory viewpoint.

III. Automatic SW-Length Selection Strategies

In an RD context, either the admissible bit-rate or distortion constraints are typically prescribed, so the impact of the SW-length upon each will now be, respectively, investigated.

A. SW-Length Selection for a Bit-Rate Constrained Framework

In an encoder constrained by admissible bit-rate, the most appropriate SW-length is determined from the bit-rate and shape contour length. Since the SW provides a lower bound on the number of CP, if the average number of bits required per CP is known, the maximum number of CP can be obtained from the admissible bit-rate and the SW-length L' calculated. If r_{avg} is the average number of bits required to encode a CP, then the maximum number of CP is $nCP \approx R_{\max}/r_{avg}$, where nCP denotes the maximum number of CP, $R_{\max}$ is the admissible bit-rate, and $L' \geq N_B/nCP$, so

$$L' \geq \frac{N_B \cdot r_{avg}}{R_{\max}}. \tag{1}$$

This value of L' only holds true if the furthest FCB point from the current CP within the SW is always selected as the next CP. In practice however, the next CP will be a random distance within the SW, so the effective length of the window will usually be less than the actual window length L. It is thus required to determine the actual SW-length that will produce an *effective length* L'. If $prob(i)$ is the probability of the next selected CP lying a distance i-pixels within the window, then the effective SW-length $L' = \sum_{1 \leq i \leq L} i \cdot prob(i)$ and since $L \geq L'$, the normalized *effective-to-actual window* length L'/L, so from (1)

$$L \geq \frac{N_B \cdot r_{avg}}{ex \cdot R_{\max}}. \tag{2}$$

As discussed in Section II, the encoding method in [9] uses a combination of CC and RL. The direction of the next CP is 3-bit chain coded, with the distance considered as the RL which can be encoded by either an FLC or VLC technique. It needs to be stressed that a 2-bit *improved orientation encoding* scheme for direction coding is unsuitable for SW implementations since for the zero distortion case it will be unable to encode either a straight line of length greater than L or the maximum RL supported by the coder. An efficient VLC, namely the 15-codeword based LC has therefore been used within the ORD optimal shape coding framework [9], and as both FLC and LC-based coding schemes have been proposed, SW-length selection strategies for these will be analyzed in the following two subsections.

Appropriate SW-length for FLC: an SW-length of L mandates $\lceil \lg L \rceil$ bits, where $\lg L = \log_2 L$ and $\lceil\ \rceil$ is the ceiling operator, to encode the RL between two consecutive CP since the next CP must lie within L points of the current CP. Thus $r_{avg} = 3 + \lceil \lg L \rceil$ bits, where the 3-bit component refers to the direction CC length. Furthermore, rigorous experimental data analysis, including statistical distribution tests (full details are provided in Appendix A), revealed that $prob(i),\ 1 \leq i \leq L$ broadly follows a uniform distribution for all values of i. Hence, $prob(i) = 1/L$ is considered, so $L' = (L+1)/2$ and $ex = L + 1/2L$, and the value of L for FLC becomes

$$L \geq \frac{2L \cdot (3 + \lceil \lg L \rceil)}{(L+1) \cdot R_{\max}} \cdot N_B. \tag{3}$$

SW-length for LC: In LC, the next CP is separated from the current CP by i pixels, and so can be encoded by $r_{avg} = 3 + 2 + \lfloor \lg i \rfloor$ bits, where $\lfloor\ \rfloor$ is the floor operator. The initial 3 bits in the direction coding use CC while $2 + \lfloor \lg i \rfloor$ bits are required to encode the RL. Although r_{avg} using FLC is just the codeword length, theoretical calculations are required to obtain its corresponding value using LC. The values of r_{avg} can be calculated as

$$r_{avg} = 5 + \frac{1}{L} \sum_{i=1}^{L} \lfloor \lg i \rfloor. \tag{4}$$

If $mm = \lfloor \lg L \rfloor$, then from [16], r_{avg} in (4) becomes (the formal proof is provided in Appendix B)

$$r_{avg} = 5 + \frac{1}{L}\left(mm \cdot L + mm - 2^{mm+1} + 2\right) \tag{5}$$

$$L \geq \frac{2L \cdot \left(5 + \frac{1}{L}\left(L \cdot mm + mm - 2^{mm+1} + 2\right)\right)}{(L+1) \cdot R_{\max}} \cdot N_B. \tag{6}$$

The final equation for determining the optimal value of L in each of above two cases, i.e., either (3) or (6) forms a *system of nonlinear equations*, which can be solved using numerical methods, such as the *bisection* method [17].

Sliding Window With Multiple Shapes: The above theory implicitly assumes the image and/or video frame comprises a single object shape for a known admissible bit-rate. If multiple objects are to be encoded, the prescribed admissible bit-rate must be appropriately distributed among them. In [18], it was observed the bit-rate requirement for encoding each shape varied linearly with the number of contour points, so a given admissible bit-rate could be commensurately allocated amongst the shapes. For a predetermined number of contour points, one shape can clearly be more complicated than another, so if only the number of contour points is considered in the distribution, more complex shapes will be under allocated and prone to larger

distortions. To ensure the same level of distortion, more bits need to be allocated to shapes exhibiting greater complexity, which implies that shape complexity (C_x) must be considered alongside the number of contour points (N_B). The admissible bit-rate for the ith contour $R^i_{\max}$ can thus be expressed as

$$R^i_{\max} \propto N^i_B.Cx^i. \tag{7}$$

One of the most popular shape complexity measures is the ratio of the number of interior pixels to contour pixels of a shape [18]. Adopting a similar rationale, this paper measures the shape complexity as the ratio of the number of contour pixels to the shape area, so for a given number of contour points, the smaller the area, the greater shape complexity. As a consequence, (7) now becomes

$$R^i_{\max} \propto \frac{N^i_B.N^i_B}{Area^i} = \frac{N^{i^2}_B}{Area^i} = w^i \tag{8}$$

where w^i represents the weight for the ith shape which is proportional to the allocated bit-rate $R^i_{\max}$.

Cognizance needs to be taken as to whether there are unallocated bits, so to maximize the coding efficiency, unused bits from one shape are reallocated to other objects. If O shapes are to be encoded, the bit distribution for each shape within a bit-rate constraint framework can be formalized as

$$R^i_{\max} = \Delta R^i + \frac{R_{\max} \cdot w^i}{\sum\limits_{1 \le i \le O} w^i}. \tag{9}$$

If R'^{i-1} is the number of bits used to encode the $(i-1)$th contour, then $\Delta R^i = R^{i-1}_{\max} - R'^{i-1}$ provides the mechanism to exploit unallocated bits from the previous contour when coding the current contour, with $\Delta R^1 = 0$.

This strategy is germane for any image or video frame comprising either a single or multiple objects when the admissible bit-rate is constrained. In the example in Fig. 4(a) and (b), three different shapes in the frame #7 of the Kids.sif test sequence are displayed—the ball, left kid (LKid), and right kid (RKid) which are shown in Figs. 4(c), 5(d), and Fig. 5(e), respectively. These shapes have 87, 393, and 521 contour points giving corresponding interior shape areas of 743, 5815, and 7103.5.

B. SW-Length Selection in a Distortion Constrained Framework

While bit-rate is a non-increasing function of SW-length, no such functional relationship exists between window length and distortion. It is therefore not feasible either to determine the SW-length for a given admissible distortion or to introduce a parameter to investigate the relationship between SW-length and distortion. This can be visualized in Fig. 6 which shows a typical RD-SW-length dynamic, from which it is clear that for a given distortion, the bit-rate decreases as the SW-length increases. The graph also reveals that for a particular L, there is little change in bit-rate with distortion variations, though smaller SW-lengths mandate higher bit-rates, which will provide faster encoding and vice versa. As was highlighted in Section II, for small SW-lengths the rate-of-change in bit-rate $(\Delta r/\Delta L)$ with respect to SW length is high while for longer window lengths it tends towards zero, so for an admissibly-constrained encoder, the SW-length can always be designed to provide a $\Delta r/\Delta L$ value either less than some predefined threshold or close to zero.

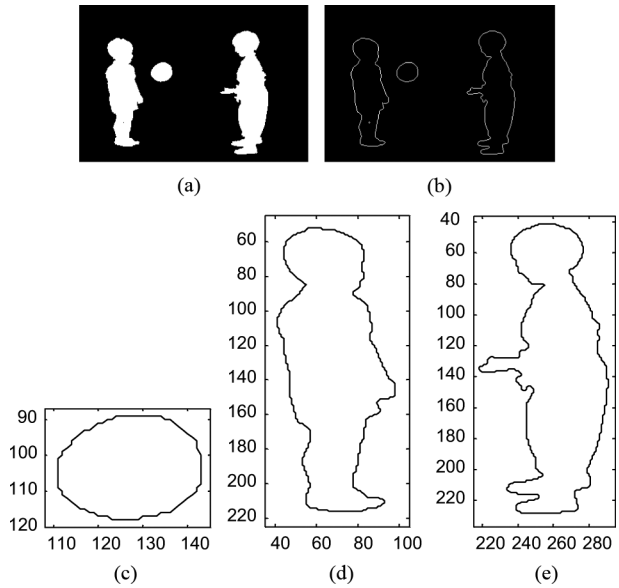

Fig. 4. (a) Three different shapes in frame #7 of the *Kids*.sif sequence, (b) the contours —(c) the *ball*, (d) *left kid* (LKid), and (e) *right kid* (RKid) (the horizontal and vertical axes, respectively, represent the Cartesian X and Y axes).

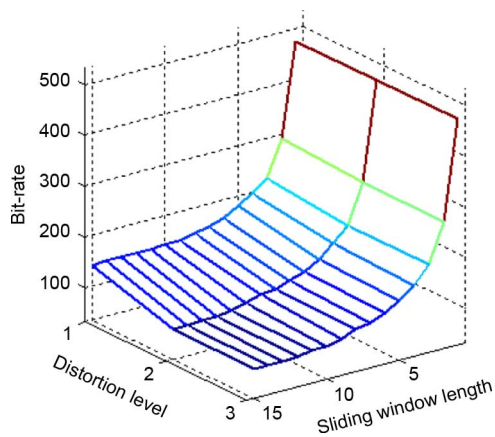

Fig. 5. Typical RD and SW-length dynamic (bit-rate in bits; SW-length in pixels) while the distortion level refers to $T_{\max}$ in pixels with $T_{\min} = 0$ (figure best viewed in color).

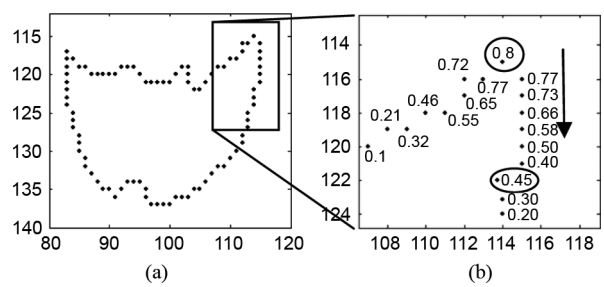

Fig. 6. Cornerity at contour points—(a) *Neck* region contour of frame #31 of the *MissAmerica*.qcif sequence, (b) respective cornerity values at the points indicated by the rectangle in (a) SW (the horizontal and vertical axes, respectively, represent the Cartesian X and Y axes).

For bit-rate constrained encoders, the relevant probability values in (2)–(6) may not be known *a priori* because the CP are usually chosen based on the RD constraints during coding. Furthermore, the SW-length for an allocated bit-rate and contour

length obtained from any of (3)–(6) is fixed irrespective of the shape's complexity, i.e., curvature, for the entire shape which does not assist the encoder in retaining distinctive features, such as sharp corners.

Adaptive design techniques for determining the SW-length in shape coders constrained by either admissible bit-rate or distortion [10], [11], generally use variable window lengths for different contour points based on the localized curvature or *cornerity*. While choosing a narrow SW is desirable when approaching a corner point to preserve a shape's cornerity, it has the limitation that maintaining a narrow SW-length for contour points immediately following a sharp corner point can mean including surplus CPs which comprises the RD performance. This paper presents a generalized SW design strategy which affords improved RD performance, while both retaining shape cornerity and providing an efficient memory solution.

C. Adaptive Sliding Window (ASW) Techniques

Generally adaptive SW techniques automatically adjust the SW-length to a shape's cornerity irrespective of the RD constraint criteria. The cornerity K of a contour point is defined as the maximum deflection angle between the two dominant edge directions in the neighborhood of that point. Any point having a local cornerity maxima is considered a *corner* [19] point. In this paper, the Bues-Tiu corner detection technique [20] is adopted because it provides the most analogous results to human perception [21]. As the cornerity of points along a contour do not change abruptly, but rather gradually follow a trend to and from the local maximum, its value is useful in determining the most appropriate SW-length. By monitoring K for each shape point, the SW-length is adaptively changed so contour points with a higher cornerity induce a smaller length and vice versa. This ensures additional CP for a contour segment that has more frequent sharp changes and corners, and fewer CP for segments where the rate of change in shape is more gradual. In existing techniques, the SW-length is always small in the vicinity of a corner point. While this is crucial as the encoder approaches a corner point to retain cornerity, after passing the corner point, keeping a narrow SW unnecessarily restricts the encoder to a small number of contour points. This increases the overall bit-rate requirement with little (or no) improvement in the distortion quality of a shape, as illustrated by the example in Fig. 6(a) for the *Neck* region contour of frame#31 of the *MissAmerica*.qcif sequence, where the corresponding K values for the highlighted contour section are displayed in Fig. 6(b).

At the two highlighted corner points in Fig. 6(b), the K values are locally maximal before decreasing for contour points further away from this corner point. The maximum and minimum SW-lengths are, respectively, assigned to the contour points having globally the lowest and highest cornerity values along the whole shape, with a mapping then performed between K and the SW-length range. While [10] and [11] provide an effective L determining strategy, when the encoder approaches a corner point it can still be influenced by the legacy of a previous corner point forcing it to include an extra CP which is neither necessary nor desirable. In Fig. 6(b) for example, the corner point with $K = 0.8$ can influence the encoder to include an extra CP between this point and the next corner point with $K = 0.45$. To avoid this situation, once a corner point has been coded its influence is negated so the SW can be effectively applied to reduce the bit-rate. To achieve this objective, a *forward arm length* (FAL) is introduced alongside K to determine the best SW-length in a *generalized SW* (GSW) model.

The GSW strategy: The FAL (ζ) is the distance in terms of the number of contour points between a contour point and the next corner point. A large ζ thus implies the next corner point is some distance away so a larger L can be employed. In the above example, the two contour points with $K = 0.73$ and $K = 0.66$, respectively, will have a corresponding FAL of $\zeta = 6$ and $\zeta = 5$. The SW-length is proportional to ζ and inversely related to K, so at each contour point $L \propto K$. The maximum and minimum SW-lengths are then assigned to the contour points having the lowest and highest global ζ/K values along an entire shape contour. A linear mapping between the cornerity ζ/K of each contour point and this SW-length range is established so the SW-length for the jth contour point is

$$L[j] = \begin{cases} L_{\max}, & \text{if} K_{\max} = K_{\min} \\ L_{\min} + \frac{L_{\max} - L_{\min}}{\frac{\zeta}{K}_{\max} - \frac{\zeta}{K}_{\min}} \cdot \left(\frac{\zeta}{K}_{\max} - \frac{\zeta}{K}[j] \right) & \text{otherwise} \end{cases} \tag{10}$$

where $\zeta/K_{\max}$ and $\zeta/K_{\min}$ are the maximum and minimum ζ/K values of points on a shape contour. The GSW strategy is effective for both cursive and non-cursive shapes, while for the special case of a straight line $K_{\max} = K_{\min}$ and the SW-length $= L_{\max}$.

As the encoder may be constrained by either bit-rate or admissible distortion, in both cases for LC, $L_{\max}$ is $15\sqrt{2} \approx 21$ pixels, while for FLC it is 2^{η}, where η is the code length in bits. If the bit-rate is constrained, $L_{\min}$ is estimated from (1) using the corresponding values of r_{avg} for LC and FLC; while for an encoder constrained by admissible distortion, $L_{\min} = \lceil T_{\max} \rceil$ for both LC and FLC, as otherwise the encoder does not fully exploit the admissible distortion. Note, for multiple object shapes, both the bit-rate allocation and reallocation strategies in (9) are applicable to the new GSW algorithm.

In summary, the ASW technique [10], [11] uses only the cornerity value of contour points to determine the best SW-length, while the GSW paradigm incorporates cornerity together with relative spatial information in the form of the FAL ζ between a contour point and the next corner point. This not only produces better RD performance but a more pragmatic SW design, as a narrower SW is applied leading to, and at a corner point, with a longer SW used once the corner point has been passed.

D. Complexity Analysis

From a complexity perspective, SW-length calculations for the bit-rate constraint are computationally efficient because they are performed just once outside the main loops and are directly related to the numerical method applied to determine L. The new GSW technique is also efficient as it too is performed outside the main loops, and has $O(N_B)$ computational complexity in calculating K, ζ, and L for all the contour points. While FAL calculations do incur a few extra arithmetic operations, this does not impact on the complexity order. In considering the memory requirements, as discussed in Section II-D, the encoder can always be designed using just two $O(L)$ memory blocks, so for

TABLE I
SW LENGTH AND ITS EFFECTS ON BOTH RD AND COMPUTATIONAL SPEED PERFORMANCE FOR GIVEN ADMISSIBLE DISTORTIONS (UNITS: SW-LENGTH L—PIXEL; $T_{\max}$—PIXEL; $T_{\min}$—PIXEL; MPEG4 (D_n) [6] DISTORTION METRIC IN %; TIME—S)

Video sequence↓	$R_{\max} = 400$ *bits*											$R_{\max} = 500$ *bits*										
	SW using (6)						GSW					SW using (6)						GSW				
	L	$T_{\max}$	$T_{\min}$	D_n	*bits*	time	$T_{\max}$	$T_{\min}$	D_n	*bits*	time	L	$T_{\max}$	$T_{\min}$	D_n	*bits*	time	$T_{\max}$	$T_{\min}$	D_n	*bits*	time
MissAmerica.qcif	12	3	3	2.12	398	2.4	2	1	1.88	400	2.6	8	2	1	1.68	497	2.2	1	1	1.68	429	2.6
Bream.qcif	12	5	4	3.4	399	3.5	3	2	3.1	398	3.7	9	3	3	2.79	500	3.07	1	1	2.06	497	3.6
Weather.qcif	9	2	1	1.73	373	1.8	1	1	1.37	359	2.0	7	1	0	1.23	499	1.5	1	0	1.23	431	2.0

the GSW approach, the maximum SW-length L is always considered as this provides a worst-case design.

The models presented for determining the most appropriate SW-length can seamlessly be embedded into the dynamic vertex-based ORD optimal polynomial shape coding framework [6], [10]. The next section presents experimental results and analysis to assess the efficacy and performance of these new techniques.

IV. RESULTS AND ANALYSIS

The first set of experiments was performed to evaluate the effectiveness of the SW-length selection models for admissible bit-rates using *Polygon-FCB–ADMSC*[1] technique. Beside the two admissible peak distortion bounds ($T_{\max}$ and $T_{\min}$), the experimental results are compared with the MPEG4 standard distortion metric D_n is defined as the ratio of number of erroneous pixels to the total number of vertices pixels in the shape [6].

Table I summarizes the estimated SW-lengths L, RD and computational speed results when applying: 1) the fixed-length SW strategy using LC in (6), and 2) the new GSW algorithm (Section III-C). For the admissible bit-rates of $R_{\max} = 400$ and $R_{\max} = 500$ bits in frame #1 of the *Weather*.qcif sequence, (6) gave SW-lengths of 9 pixels and 7 pixels, respectively. These values were subsequently used in the encoding process, so for $R_{\max} = 400$ bits, the admissible distortion pair $T_{\max}$ and $T_{\min}$ were, respectively, 2 pixels and 1 pixel (with the corresponding $D_n = 1.73\%$), with 373 bits utilized and a total coding CPU time of 1.8 s. In contrast, the two distortion bounds for GSW were both 1 pixel ($D_n = 1.37\%$), with a corresponding bit-rate utilisation of 359 bits and a computing time of 2.0 s. These results reflect the superior RD performance and the rationale for GSW which is greater flexibility in using smaller SW-length at corner points and a larger L at more gradual parts of the shape. The corollary, however, is that L must be calculated at every contour point, so GSW is computationally slower than the fixed-length scheme. The results also corroborate that increasing the admissible bit-rate results in a smaller SW-length and vice versa. This is especially noteworthy when consideration is made that in calculating the total computational time overhead, the time for determining L is included. Finally, from an implementation viewpoint, further efficiency gains are feasible for both these methods by employing a *look-up table* (LUT) of appropriate SW-lengths for a range of admissible bit-rates and shape length combinations in terms of the number of contour points, so the most appropriate L can be obtained from a straightforward LUT access.

[1]This nomenclature denotes the combination of polygonal encoding with FCB and ADMSC being the source of potential CP and the distortion measurement technique, respectively.

TABLE II
AVERAGE BIT-RATE REQUIREMENTS FOR VARIOUS FIXED SW-LENGTHS L; THE MINIMUM SW-LENGTHS WHERE THE GAIN IN BIT-RATE/SW-LENGTH RATIO ≤ 15; GSW AND ASW APPROACHES FOR A PEAK DISTORTION WITH $T_{\min} = 1$ pixel

SW-length (L)/type→		**5**	**10**	**15**	**20**	$\frac{\Delta r}{\Delta L} \leq 15$	**ASW [10, 11]**	**GSW**
Video sequence↓		**Bit rate requirement**				***L*, Rate**	**Rate**	**Rate**
MissAmerica.qcif (176×144, 100 frames)	$T_{\max} = 1$	613	484	432	429	13, 441	432	429
	$T_{\max} = 2$	609	468	405	405	13, 420	405	405
	$T_{\max} = 3$	609	463	385	369	14, 387	372	369
Bream.qcif (176×144, 100 frames)	$T_{\max} = 1$	700	558	498	497	13, 514	498	497
	$T_{\max} = 2$	695	538	454	448	13, 474	450	448
	$T_{\max} = 3$	681	503	408	398	13, 432	402	398
Weather.qcif (176×144, 100 frames)	$T_{\max} = 1$	504	389	356	356	12, 369	362	359
	$T_{\max} = 2$	504	368	331	331	13, 340	331	331
	$T_{\max} = 3$	504	348	283	282	13, 299	282	282
Akiyo.sdtv (720×480, 100 frames)	$T_{\max} = 1$	2425	2177	2091	2091	14, 2093	2095	2091
	$T_{\max} = 2$	2423	2163	2067	2056	14, 2078	2061	2056
	$T_{\max} = 3$	2418	2115	1971	1950	14, 1983	1952	1950
Stefan.sdtv (720×480, 100 frames)	$T_{\max} = 1$	2068	1695	1523	1517	15, 1523	1521	1518
	$T_{\max} = 2$	2058	1659	1442	1434	15, 1442	1438	1435
	$T_{\max} = 3$	2058	1631	1380	1352	15, 1366	1362	1357

The next series of experiments were performed from an admissible distortion perspective using a fixed-length SW, the diminishing *rate-of-return* principle, the original ASW algorithm [10], [11] and new GSW technique. Table II summarizes the quantitative results, where the first 4 columns display the bit-rate requirements for arbitrarily selected fixed-length windows, the next column displays the SW-length at which the bit-rate gain falls below a certain threshold (see Section III-B), while the final two columns show the results for ASW and GSW, respectively.

For the *Weather*.qcif shape for example, it was found that for SW-lengths of 5, 10, 15, and 20 pixels, the corresponding bit-rate requirements for $T_{\max} = T_{\min} = 1$ pixel were 504, 389, 356, and 356 bits, so for $L = 15$ pixels and $L = 20$ pixels both the bit-rate and distortion remain the same. In these circumstances, using a 20 pixel SW incurs unnecessary computational time and memory without improving RD performance. The results in the next column are thus very insightful, as they reveal the automatically determined SW-length from the gain in bit-rate performance when it falls below a threshold of 15 pixels, with the most appropriate SW-length being 12 pixels,

TABLE III
MPEG4 DISTORTION MEASURE $D_n(\%)$ AND COMPUTATIONAL TIME REQUIREMENTS (s) FOR VARIOUS FIXED AND ADAPTIVE SW-LENGTHS L AT DIFFERENT PEAK DISTORTIONS AND $T_{\min}$ PIXEL

SW-length (L)/type→		**5**	**10**	**15**	**20**	**ASW [10, 11]**	**GSW**
Video sequence↓		***D_n*(%), Time requirement (*secs*)**					
MissAmerica.qcif (176×144, 100 frames)	$T_{\max}=1$	1.66, 1.95	1.68, 2.34	1.66, 2.70	1.69; 3.09	2.28, 2.10	2.3, 2.12
	$T_{\max}=2$	1.53, 1.96	1.77, 2.38	1.88, 2.69	1.88, 3.07	2.3, 2.02	2.3, 2.11
	$T_{\max}=3$	1.53, 1.96	1.84, 2.39	2.4, 2.76	2.42, 3.13	2.3, 1.98	2.3, 2.01
Bream.qcif (176×144, 100 frames)	$T_{\max}=1$	2.75, 1.79	3.25, 1.78	2.13, 3.73	2.13, 4.26	2.16, 3.58	2.16, 3.60
	$T_{\max}=2$	1.95, 2.78	2.33, 3.26	2.65, 3.70	2.8, 4.18	2.71, 3.51	2.70, 3.68
	$T_{\max}=3$	2.11, 2.76	2.65, 3.26	3.48, 3.75	3.7, 4.20	3.7, 3.55	3.7, 3.70
Weather.qcif (176×144, 100 frames)	$T_{\max}=1$	1.47, 1.43	1.34, 1.75	1.37, 2.04	1.37, 2.35	1.23, 2.06	1.22, 2.06
	$T_{\max}=2$	1.47, 1.45	2.01, 1.78	2.01, 2.04	2.01, 2.35	2.01, 2.08	2.00, 2.07
	$T_{\max}=3$	1.47, 1.46	2.31, 1.78	2.86, 2.20	2.86, 2.34	2.86, 2.09	2.86, 2.09
Akiyo.sdtv (720×480, 100 frames)	$T_{\max}=1$	0.21, 21.44	0.23, 23.75	0.24, 24.65	0.24, 26.24	0.25, 22.75	0.24, 22.95
	$T_{\max}=2$	0.22, 21.72	0.24, 23.43	0.27, 24.74	0.27, 25.90	0.25, 22.65	0.25, 23.01
	$T_{\max}=3$	0.22, 21.86	0.28, 23.21	0.35, 24.68	0.37, 25.75	0.25, 22.72	0.25, 23.05
Stefan.sdtv (720×480, 100 frames)	$T_{\max}=1$	0.9, 15.50	1.11, 16.93	1.32, 17.81	1.37, 19.36	1.37, 17.57	1.37, 17.69
	$T_{\max}=2$	0.94, 15.57	1.43, 17.08	1.81, 17.88	1.87, 19.47	1.88, 17.65	1.86, 17.755
	$T_{\max}=3$	0.94, 15.57	1.52, 17.24	2.29, 17.97	2.70, 19.49	2.69, 17.74	3.67, 17.858

which gives a corresponding bit-rate of 369 bits. The ASW and GSW techniques conversely required 362 and 359 bits, respectively, for the encoding, with the improvement in the latter being directly due to the introduction of the FAL distance ζ alongside the *cornerity* K. Interestingly, Table II confirms the proposed enhancement are equally applicable to higher spatial resolution shapes, with for example, similar consistent bit-rate improvements evidenced in both the *Stefan*.sdtv and *Akiyo*.sdtv sequences. In addition since GSW improves the bit-rate requirements over ASW for most shapes and does not deteriorate it for any for any shape (see Table II), this exhibits that the proposed GSW algorithm will provide considerable benefit to all applications specially the applications that require low bit-rate video coding.

The corresponding CPU times and distortion values in D_n are presented in Table III and show for the same distortion setup, increasing SW-lengths lead to longer CPU times. For the *MissAmerica*.qcif sequence for instance, with $T_{\max} = 1$ and $T_{\min} = 1$ pixel, the time requirements were 1.96, 2.38, 2.69, and 3.07 s and the produced D_n were 1.53%, 1.84%, 2.4%, and 2.42% for SW-lengths of 5, 10, 15, and 20 pixels, respectively. When the adaptive SW-length strategies are considered, ASW and GSW efficiently balanced the overall time requirement as the lengths were variable, and while GSW was marginally higher, the times were still comparable, so for *MissAmerica*.qcif, GSW and ASW required 2.12 and 2.10 s, respectively.

Tables II and III present a detailed performance analysis into applying different SW-lengths and both confirm the effectiveness of the new strategies for determining the best L. The results in Tables II and III are for a polygon vertex-based ORD optimal shape coding algorithm and while higher-order curves such as B-splines will produce superior RD performance, this is achieved at the pyrrhic cost of higher computational expenditure. For completeness, supplementary results using quadratic B-splines have been included in Appendix C.

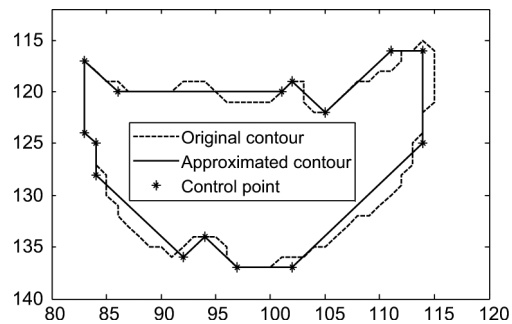

Fig. 7. Shape coding of the *Neck* region using *Polygon-FCB–ADMSC-GSW* for $T_{\max} = 2$, $T_{\min} = 1$ pixel.

Fig. 7 demonstrates the performance of the GSW-length strategy displaying the decoded shape using *Polygon-FCB–ADMSC* for $T_{\max} = 2$, $T_{\min} = 1$ pixel. It is readily apparent that the approximating shape has retained all its sharp corners by adaptively assigning more CP within these regions, while concomitantly using fewer CP in the flatter regions to vindicate the combined usage of both shape *cornerity* and FAL at all contour points.

To investigate the effectiveness of the new admissible bit-rate distribution strategy in (9) for multiple shapes, a further set of experiments were undertaken. Comparative results for both the new and established scheme [22] using (3) are presented in Table IV. For an admissible bit-rate of $R_{\max} = 1600$ bits for example, the LKid was allocated 567 and 666 bits, respectively, by the new and existing bit-allocation scheme [22], with (3) giving the corresponding SW-lengths of 10 pixels and 8 pixels. Comparing the overall distortion and computational time re-

TABLE IV
ESTIMATED SW-LENGTH WITH CONSTRAINED BIT-RATES, OBTAINED PEAK DISTORTIONS $T_{\max}$ AND $T_{\min}$ (PIXEL), D_n (%), BIT-RATE UTILIZATION, AND TIME REQUIREMENT (*S*) FOR *POLYGON*-FCB-ADMSC UPON THE KIDS.SIF (RESOLUTION: 352×240 pixels) SEQUENCE

			New bits allocation scheme (9) using (3)						Old allocation scheme [22] using (3)						
Total no of admissible *bits*	Shape	Admissible *bits*	L	Bits used	$T_{\max}$	$T_{\min}$	D_n	Time (*secs*)	Admissible *bits*	L	Bits used	$T_{\max}$	$T_{\min}$	D_n	Time (*secs*)
1300	*LKid*	461	14	461	3	2	4.3	3.8	541	12	513	3	2	4.1	3.7
	ball	176	5	162	1	1	2.5	0.22	160	8	112	1	1	4.9	0.38
	RKid	676	10	648	2	2	3.7	4.94	675	10	648	2	2	3.7	4.94
1600	*LKid*	567	10	561	2	2	2.3	3.43	666	8	634	2	1	2.27	3.43
	ball	223	4	180	1	1	2.4	0.21	192	6	148	1	1	3.4	0.31
	RKid	859	7	858	1	0	0.5	4.45	818	7	779	1	1	2.2	4.6

TABLE V
MEMORY REQUIREMENT RESULTS (UNITS: *DOUBLE* BYTES) FOR FRAME #30 OF THE *MISS*AMERICA .QCIF $N_B = 310$ SEQUENCE WITH $T_{\max} = 1$, $T_{\min} = 1$ pixel AND AN FCB WIDTH OF ZERO PIXEL $L_{\max} = 16$

	On-cache Memory				Off-cache Memory	
SW type↓	*Block 1*		*Block 2*			
Fixed SW	16 + 17;	$L_{\max}+17$	16×2 + 10;	$2 \cdot L_{\max}+10$	310×2 + 1;	$2 \cdot N_B+1$
GSW	16×2 + 16;	$2 \cdot L_{\max}+17$	16×2 + 10;	$2 \cdot L_{\max}+10$	310×3;	$3 \cdot N_B$
Without SW	310 + 16;	N_B+16	310×2+10;	$2 \cdot N_B+10$	310×2;	$2 \cdot N_B$

sults, the superior performance of the new scheme based on (3) is clear, with the overall time requirement for the proposed and old bit allocation schemes being 8.09 and 8.34 s, respectively. In addition, the average D_n distortion values were respectively 1.73% and 2.62%, so for a given bit-rate, the new bit allocation scheme provided both improved distortion performance allied with lower computational overheads.

Finally, to analyze the storage implications of applying an SW, the maximum data structure sizes required during the execution of non-optimised code (Matlab program) was recorded. As discussed in Section II-D, there are two main blocks of on-cache memory for the main program (*Block 1*) and various functions and subroutines (*Block 2*). In addition, off-cache memory is used for operations outside the main loops of Algorithm 1. Table V summarizes the memory requirements to execute the program for frame #30 of the *MissAmerica*.qcif sequence with $T_{\max} = 1$, $T_{\min} = 1$ pixel and an FCB width of zero pixels ($L_{\max} = 16$).

The *Block 1* on-cache memory requirement without an SW is $(310 + 16)$ units, where the 310 is the data structure array *MinRate()* size, i.e., number of contour points, while the 16 units refers to the overhead for all other variables. For a fixed-length SW, *Block 1* only requires $(16 + 17)$*double bytes*, since 16 is the *MinRate()* for the SW-length and 17 the storage for all other variables plus one extra SW-length variable. The corresponding GSW *Block 1* storage size is $(16 \times 2 + 16)$ units, where the two 16 units, respectively, reflect the array size of *MinRate()* and SW-length variables. *Block 2* represents the maximum memory size required for all subroutines, with the *Edge-distortion()* calculation subroutine (Step 5) of Algorithm 1 incurring the highest memory cost. The *Block 2* memory sizes for the three scenarios in Table V, namely a fixed-length SW, GSW, and no SW, respectively, are: $16 \times 2 + 10$, $16 \times 2 + 10$, and $310 \times 2 + 10$ *units of memory*, revealing the size of *Block 2* has been significantly reduced by applying an SW. The off-cache memory requirement is proportional to the contour length, so GSW uses more memory than the other two cases because of the variable window-length at each contour point. The off-cache memory however, is accessed only once per contour and so has negligible influence on the computational time/memory performance of the encoder. It must be stressed the on-cache memory cost when using an SW is solely constrained by the maximum SW-length $L_{\max}$ and not by the contour length. This means the SW ensures consistent, symmetrical on-cache *Block 1* and *Block 2* memory arrangements independent of the shape length, thus enabling any hardware implementation such as in an FPGA, to be both elegant and straightforward.

V. CONCLUSION

The SW is an established tool in vertex-based ORD shape coding algorithms for reducing the computational impost, and avoiding trivial solutions while affording straightforward hardware implementations. This paper has introduced a series of new design techniques for SW-length including an efficient bit-allocation strategy for managing multiple shapes, and both a statistical and shape adaptive approach to automatically determining the most appropriate SW-length. The statistical approach maximizes utilization of the prescribed bit-rate so improving the overall shape distortion, together with the computational complexity and memory requirements for ORD implementations. For the shape adaptive scheme, relative spatial information combined with cornerity is exploited to reduce the requisite bit-rate for various peak distortions. The results substantiate the superior SW designs achieved with these ORD enhancements from a *rate distortion* perspective, compared with existing techniques.

REFERENCES

[1] S. M. Aghito and S. Forchhammer, "Context based coding of binary shapes by object boundary straightness analysis," in *Proc. Data Compression Conf. (DCC)*, Snowbird, UT, 2004, pp. 399–408.

[2] S. M. Aghito and S. Forchhammer, "Context-based coding of bilevel images enhanced by digital straight line analysis," *IEEE Trans. Image Process.*, vol. 15, no. 8, pp. 2120–2130, Aug. 2006.
[3] N. Brady, F. Bossen, and N. Murphy, "Context-based arithmetic encoding of 2D shape sequences," in *Proc. Int. Conf. Image Processing (ICIP)*, Washington, DC, 1997, pp. 29–32.
[4] H. Freeman, "On the encoding of arbitrary geometric configurations," *IRE Trans. Electron. Comput.*, vol. EC-10, pp. 260–268, 1961.
[5] I. E. Richardson, *H.264 and MPEG-4 Video Compression: Video Coding for Next Generation Multimedia*. New York: Wiley, 2003.
[6] 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.
[7] D. Venkatraman and A. Makur, "A compressive sensing approach to object-based surveillance video coding," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, 2009.
[8] L. P. Kondi, G. Melnikov, and A. K. Katsaggelos, "Joint optimal object shape estimation and encoding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 4, pp. 528–533, Apr. 2004.
[9] G. M. Schuster and A. K. Katsaggelos, *Rate-Distortion Based Video Compression: Optimal Video Frame Compression and Object Boundary Encoding*. Norwell, MA: Kluwer, 1997.
[10] F. A. Sohel, L. S. Dooley, and G. C. Karmakar, "New dynamic enhancements to the vertex-based rate-distortion optimal shape coding framework," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 10, pp. 1408–1413, Oct. 2007.
[11] F. A. Sohel, G. C. Karmakar, and L. S. Dooley, "Dynamic sliding window width selection strategies for rate-distortion optimal vertex-based shape coding algorithms," in *Proc. Int. Conf. Signal Processing (ICSP)*, Guilin, China, 2006.
[12] Z. Lai, W. Liu, and Y. Zhang, "Perceptual relevance measure for generic shape coding," in *Proc. Data Compression Conf. (DCC)*, 2009.
[13] F. A. Sohel, L. S. Dooley, and G. C. Karmakar, "Accurate distortion measurement for generic shape coding," *Pattern Recognit. Lett.*, vol. 27, pp. 133–142, 2006.
[14] F. A. Sohel, G. C. Karmakar, L. S. Dooley, and M. Bennamoun, "Geometric distortion measurement for shape coding: A contemporary review," *ACM Comput. Surv.* , vol. 44, 2012.
[15] F. W. Meier, G. M. Schuster, and A. K. Katsaggelos, "A mathematical model for shape coding with B-splines," *Signal Process.: Image Commun.*, vol. 15, pp. 685–701, 2000.
[16] F. A. Sohel, G. C. Karmakar, L. S. Dooley, and M. Bennamoun, "Geometric distortion measurement for shape coding: A contemporary review," *ACM Comput. Surv.*, 2010.
[17] R. L. Graham, D. E. Knuth, and O. Patashnik, *Concrete Mathematics: A Foundation for Computer Science*, 2nd ed. Reading, MA: Addison-Wesley Longman, 1994.
[18] J. B. Scarborough, *Numerical Mathematical Analysis*. Baltimore, MD: The Johns Hopkins Univ. Press, 1966.
[19] Z. Chen and K. N. Ngan, "Linear rate-distortion models for MPEG-4 shape coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 6, pp. 869–873, Jun. 2004.
[20] T.-Y. Phillips and A. Rosenfeld, "A method of curve partitioning using arc-chord distance," *Pattern Recognit. Lett.*, vol. 5, pp. 285–288, 1987.
[21] A. Melkman, "On-line construction of the convex hull of a simple polygon," *Inf. Process. Lett.*, vol. 25, pp. 11–12, 1987.
[22] F. A. Sohel, "Vertex-based shape coding framework," Ph.D. dissertation, Gippsland Sch. Inf. Technol., Monash Univ., Churchill, Australia, 2007.

Ferdous A. Sohel (M'07) received the B.Sc. degree in computer science and engineering from Bangladesh University of Engineering and Technology, Dhaka, Bangladesh, and the Ph.D. degree from Monash University, Churchill, Australia.

He is currently a Research Assistant Professor at the University of Western Australia, Crawley, Australia. His research interests include computer vision, pattern recognition, shape coding, and error concealment. He has published over 30 scientific publications. He secured more than $700 000 of research grants.

Dr. Sohel is a recipient of the Mollie Holman Doctoral Medal and Faculty of Information Technology Doctoral Medal from Monash University.

Gour C. Karmakar received the B.Sc. Eng. degree in computer science and engineering from Bangladesh University of Engineering and Technology, Dhaka, Bangladesh, in 1993 and the M.Sc. and Ph.D. degrees in information technology from the Faculty of Information Technology, Monash University, Churchill, Australia, in 1999 and 2003, respectively.

He is currently a Senior Lecturer at the Gippsland School of Information Technology, Monash University. He has published over 90 peer-reviewed research publications, including 13 international peer-reviewed reputed journal papers. He received more than $500 000 external and internal research grants, including the prestigious ARC linkage grant. His research interest includes image and video processing, mobile ad hoc, and wireless sensor networks.

Dr. Karmakar received 3 best paper awards in reputed international conferences.

Laurence S. Dooley (M'82–SM'92) received the B.Sc. (Hons), M.Sc., and Ph.D. degrees in electrical and electronic engineering from the University of Cymru/Wales (Swansea) in 1981, 1983, and 1987, respectively.

Since 2008, he has been a Professor of information and communication technologies in the Department of Communication and Systems at The Open University, Milton Keynes, U.K., where his principal research interests include: next-generation multimedia technologies, cognitive radio networks, distributed source coding, 3-D multimodal medical imaging, MANET/4G security, and SME technology transfer and commercialization strategies. He has co-edited one book and published 225 scientific journals, book chapters, monographs, and conference papers. He has held professorial positions in both Australia and Germany. He has supervised 18 Ph.D. students to completion together with being a recipient of significant public and private sector research funding.

Dr. Dooley received has 3 papers that were awarded international research prizes/nominations. In 2010, he received the IEEE Certificate of Award for promoting international exchange. He is a Chartered Engineer and a Fellow of the British Computer Society.

Mohammed Bennamoun received the M.Sc. degree in control theory from Queen's University, Kingston, Canada, and the Ph.D. degree in computer vision from Queen's/QUT, Brisbane, Australia.

He is currently a Winthrop Professor at the University of Western Australia, Crawley, Australia. His research interests include control theory, robotics, obstacle avoidance, object recognition, artificial neural networks, signal/image processing, and computer vision. He published more than 150 journal and conference publications.

Dr. Bennamoun served as a guest editor for a couple of special issues in international journals such as the *International Journal of Pattern Recognition and Artificial Intelligence* (IJPRAI). He was selected to give conference tutorials at the European Conference on Computer Vision (ECCV '02) and the International Conference on Acoustics Speech and Signal Processing (ICASSP) in 2003. He organized several special sessions for conferences; the latest was for the IEEE International Conference in Image Processing (ICIP) held in Singapore in 2004. He also contributed in the organization of many local and international conferences.