An Efficient Rate-Distortion Optimal Shape Coding Approach utilizing a Skeleton-based Decomposition

Haohong Wang, Guido M. Schuster, Aggelos K. Katsaggelos, and Thrasyvoulos N. Pappas*

Department of Electrical and Computer Engineering
Northwestern University
Evanston, IL 60208, USA
Email: {haohong, aggk, pappas}@ece.northwestern.edu

*Abteilung Elektrotechnik
Hochschule für Technik, Rapperswil
CH-8040 Rapperswil, Switzerland
Email: guido.schuster@hsr.ch

Abstract: In this paper, we present a new shape-coding approach, which decouples the shape information into two independent signal data sets; the skeleton and the boundary distance from the skeleton. The major benefit of this approach is that it allows for a more flexible trade-off between approximation error and bit budget. Curves of arbitrary order can be utilized for approximating both the skeleton and distance signals. For a given bit budget for a video frame, we solve the problem of choosing the number and location of the control points for all skeleton and distance signals of all boundaries within a frame, so that the overall distortion is minimized. An operational rate-distortion (ORD) optimal approach using Lagrangian relaxation and a four-dimensional Direct Acyclic Graph (DAG) shortest path algorithm is developed for solving the problem. To reduce the computational complexity from $O(N^5)$ to $O(N^3)$, where N is the number of admissible control points for a skeleton, a suboptimal greedy-trellis search algorithm is proposed and compared with the optimal algorithm. In addition, an even more efficient algorithm with computational complexity $O(N^2)$ that finds an ORD optimal solution using a relaxed distortion criterion is also proposed and compared with the optimal solution. Experimental results demonstrate that our proposed approaches outperform existing ORD optimal approaches, which do not follow the same decomposition of the source data.

EDICS: IP 1.1 Coding

1. Introduction

In recent years, object oriented video coding has received a lot of attention because it facilitates retrieval, interactive editing, and manipulation of videos. Within the object-oriented framework, a video sequence is represented through the evolution of video object planes (VOP), with each frame composed of one or more VOPs. Evolution of these VOPs in time is described in terms of shape, texture and motion information. In MPEG-4 [1] and most of the work in the literature, shape or boundary encoding is completely decoupled from texture encoding. That is, as shown in Fig. 1, for given a frame, the shape is extracted using a certain technique. The problem then becomes the encoding of the shape, which is represented either in binary form (as shown in Fig. 1(b)) or as a two-dimensional boundary or contour (as shown in Fig. 1(c)). Freeman [2] originally proposed the use of chain coding for boundary quantization and lossless boundary encoding, which has attracted considerable attention over the last forty years. The most common chain code is the 8-connect chain code, which is based on a rectangular grid superimposed on a planar curve. The curve is quantized using the grid intersection scheme and the quantized curve is represented using a string of increments. Since the planar curve is assumed to be continuous, the increments between grid points are limited to the eight grid neighbors, and hence an increment can be represented by 3 bits.

Within the MPEG-4 standardization effort [3], several contour-based shape coding methods have been developed and proved to be very efficient. In [4-6], the shape is represented using a vertex-based polygonal approximation for lossy shape coding. The placement of vertices allows for a direct control of the local variations of the shape approximation error. For lossless shape coding, the polygonal approximation "degenerates" to a chain code. In [7], a baseline shape coder places the shape into a 2-D coordinate system such that the projection of the shape onto the x-axis is the longest. The x-axis is called the baseline, from which the distance (y-coordinate) between the baseline and a point on the shape outline is measured. The shape contour is sampled clockwise. The contour points at which the direction changes

are called turning points. The shape boundary is represented by traced one-dimensional distance data with turning points, followed by entropy coding.

None of the encoders mentioned above is operationally rate-distortion (ORD) optimal because they do not provide a rigorous tradeoff between the encoding cost and the resulting distortion. In [8-10], a framework for the rate-distortion operationally optimal encoding of shape information in the intra and inter modes is proposed. First order (polygons) and higher order (i.e., splines) approximation techniques are adopted to represent the boundary, and the control points of these curves are encoded to achieve the ORD optimal result.

In addition to the approximation mentioned above which operate directly on the boundary vertices, a number of investigations have also considered alternative approaches that first transform or decompose the shape information before processing (the baseline coder already mentioned performs such transform). In [11] the shape description algorithms are classified into external and internal. The former ones are based on the description of the shape boundary, like Fourier descriptor [12,13], time series [14], and shape matrices [15]. The latter ones are mainly area descriptor algorithms, like moment based approaches [16], skeletons or medial axis transform [17-19], and shape decomposition [20].

In [17], the skeleton decomposition was proposed as an alternative shape description method for biology research. The main idea behind it is to find a set of points that are equidistant from the object boundary by means of maximally inscribed disks. The description consists of the locus of the center of each inscribed disk and its associated radius [18]. This is the morphological definition of a skeleton, and has been employed for coding of binary images [19] and motion estimation [21]. In the case of many extra branches (bones) in the skeleton, when the boundary has many outward ripples (see example in [18, p. 377]), the skeleton decomposition results in coding inefficiencies, since the skeleton points are sparsely distributed. Furthermore, if progressive contour transmission of images is considered, coding becomes even more inefficient, because the skeleton points at coarser levels are farther apart from each other.

In this paper, a new definition of the skeleton is proposed which is suitable for the problem under consideration and an ORD optimal shape coding approach is presented, which allows for more flexible

tradeoff between accuracy and bit-allocation cost. The object shape is decomposed into the skeleton (defined as the midpoints between the two boundary points) and the distance of the boundary points from the skeleton in the horizontal direction. The skeleton points are connected in the vertical (y-axis) direction (most of them are 8-connected), which facilitates processing. The decoupling of skeleton and distance signal sets allows in principle for more flexibility in encoding, that is, it allows for the application of different transform and compression methods for each data set, according to their characteristics. Furthermore, the skeleton of an object can be used for the estimation of the object motion in the inter-mode. As an example of a way to encode the two signal data sets, we apply polygonal approximation on both the skeleton and distance signals. The resulting symbols are then encoded using arithmetic coding. The scope of this paper is limited to intra frame boundary coding.

This paper is organized as follows. Section 2 provides a description of the skeleton-based shape representation. Section 3 mathematically defines the problem of the ORD optimal boundary encoding of a single object. Section 4 shows how the Lagrangian multiplier method can be applied to solve the proposed constrained problem as a series of unconstrained problems. We describe the lossy polygonal approximations of the skeleton and distance signals, and illustrate the ORD optimal process for allocating bits between the skeleton and distance signal data sets. In addition, a four-dimensional DAG shortest path algorithm is proposed, which is compared with more efficient but only near optimal methods. In Section 5, we discuss more general cases and extend the results for the jointly optimal encoding of multiple boundaries. Section 6 presents experimental results, and section 7 draws conclusions.

2. Skeleton-based shape representation

The basic idea of skeletonization is to represent an object by two or more 1D signals (skeletons with associated distance data). Each pixel of the skeleton is associated with the distance to the closest boundary pixel in a given direction. We use the horizontal distance for both simplicity and efficiency. The object boundary can be exactly recovered from its skeleton and distance data. Figure 2 shows examples of skeletonization for a frame from the “kids” sequence.

In general, there are two basic and interchangeable ways to define an object shape, by the boundary of the shape or the binary mask of the shape. By labeling the pixels on the boundary of the object with "1" while filling the inside pixels of the object with "0", the binary mask of the object is generated. On the other hand, the boundary shape can be obtained by tracing the contour of the binary mask. In both cases, there is an ambiguity which needs to be resolved, whether the boundary pixels belong to the object or the background. In this paper, we use the boundary form of the object shape, and we therefore assume that the boundary pixels belong to the object.

The definition of the boundary is as follows

$$B=\{b_1(x,y), b_2(x,y), \ldots\ldots, b_Z(x,y)\}, \quad (1)$$

where Z is the number of pixels on the boundary and $b_i(x,y)$ is the ith pixel of the boundary with x and y its horizontal and vertical coordinates, respectively. For any integer i $(1 \le i \le Z-1)$, $b_{i+1}(x,y)$ is an 8-connect neighbor pixel of $b_i(x,y)$.

We can represent the extracted skeletons R as the set of points (x,y) at the "center" of the object in the horizontal direction and the associated distance d from the boundary, i.e.,

$$R=\{(x,y,d)|\ (x+d,y) \in B,\ (x-d,y) \in B,\ (x+d+1,y) \text{ and } (x-d-1,y) \text{ are outside of the object}\}, \quad (2)$$

where d has half-pixel accuracy. A generic shape may contain more than one skeletons.

The decoupling of the boundary data into skeleton and distance signals allows for their independent encoding. Since the skeleton and the distance signals typically have different characteristics, such an approach allows us to capitalize on their difference. As an example, let us consider two special cases shown in Fig. 3. In Fig. 3(a), the skeleton signal conveys all the information of the boundary (the distance signal is constant), while in Fig. 3(b), the opposite occurs. In both cases, the 2D-shape information is represented by an 1D signal, which can result in higher compression efficiency.

Another decomposition example with actual data is shown in Fig. 4. In Fig. 4(a) a frame with two objects and three major skeletons is shown. Figure 4(b) shows the skeleton and distance signal of the leftmost major skeleton in Fig. 4(a), while Fig. 4(c) shows the skeleton and distance signal of the rightmost major

skeleton in Fig. 4(a). Notice that in Fig. 4(b), the skeleton is "smoother" than the distance signal because the boundary of the left kid in the frame is highly correlated. In Fig. 4(c), neither function is very smooth, because the boundary of the right kid in the frame is less correlated. Our experiments have shown that in most video frames, highly correlated boundaries are quite common, so our approach can result in a substantial advantage.

The skeleton and distance data in Figs. 4(b) and 4(c) contain constant or nearly constant subintervals, which can be encoded very efficiently by straight lines. Such an example is shown in Fig. 4(d), where a lossy approximation of the skeleton and distance signal of the left kid of the first frame (Fig. 4(b)) is shown. Each constant subinterval can be treated as the special case of the example shown in Fig. 3.

3. Problem formulation

In the following, we first introduce the notation to be used in this paper and then formulate the optimization problem to be solved. To simplify the problem description, we first assume that there is only one object that contains only one skeleton, and defer the solution of the general case of encoding multiple objects with multiple skeletons to section 5. Let (x_0, y_0) denote the starting point of the skeleton. The points of the skeleton are specified by $S=\{S_1, S_2,...,S_N\}$, where $(S_i + x_0, y_0+i-1)$ are the coordinates of the ith pixel point on the skeleton, with $i=1,2,...,N$. Let $T=\{T_1, T_2,..., T_N\}$ denote the corresponding distances. Thus, the coordinates of the boundary points will be $(S_i+ x_0+T_i, y_0+i-1)$ and $(S_i+ x_0-T_i, y_0+i-1)$.

Before the problem is formulated, the means for measuring the quality of a boundary approximation or a distortion metric needs to be determined. The problem of finding the most appropriate distortion metric is open and is application dependent. We do not address it in this work, but instead we are using a distortion metric adopted by MPEG-4, as explained in the next section. This allows us to also compare the results obtained with the proposed method with published results. Let us denote the distortion of the skeleton by $D(S)=\{D_{s1},D_{s2}, ...,D_{sN}\}$, where D_{si} is the distortion incurred by the ith skeleton pixel. Correspondingly, the distortion of the distance signal is denoted by $D(T)=\{D_{T1},D_{T2}, ...,D_{TN}\}$. Clearly, all distortion elements are non-negative.

3.1. Distortion Metric

The distortion metric adopted by MPEG-4, which is also utilized in this work, is given by

$$D_{MPEG-4} = \frac{Number\ of\ \ pixels\ in\ error}{Number\ of\ interior\ \ pixels}, \tag{3}$$

where a pixel is said to be in error if it belongs to the interior of the original object and the exterior of the approximating object, or vice-versa. D_{MPEG-4} in Eq. (3) refers to one object; if there are more than one objects in the scene, the sum of the object distortions will form the distortion for the complete frame.

In lossy shape coding, distortion in both the skeleton and distance will contribute to the distortion of the shape. Figure 5 shows examples of shape distortion. The object in Fig. 5(a) is the original object with its skeleton. The objects in Fig. 5(b) are the original object and its approximation resulting from the shifting of its skeleton to the left. The areas marked with dots are pixels in error to be used in measuring the distortion. Similarly, the object in Fig. 5(c) results from the expansion of the distance data. The dark areas represent pixels in error. When both skeleton and distance data are encoded in a lossy fashion, the overall distortion has a nonlinear relationship to the individual distortions of the skeleton and distance data, as explained next.

Since the number of interior pixels is fixed, we are clearly concerned only with the "Number of pixels in error" in the numerator of Eq. (3), which is denoted as $D_{tot}(S,T)$. The following lemmas address properties of $D_{tot}(S,T)$.

Lemma 1 If lossy coding is only applied on the skeleton signal and the distance signal is coded losslessly, then $D_{tot}(S,T) = 2 \cdot \sum_{i=1}^{N} D_{Si}$.

Lemma 2 If lossy coding is only applied on the distance signal, and the skeleton signal is coded losslessly, then $D_{tot}(S,T) = 2 \cdot \sum_{i=1}^{N} D_{Ti}$.

Lemma 3 If lossy coding is applied on both skeleton and distance data, then $D_{tot}(S,T) = 2 \cdot \sum_{i=1}^{N} \max(D_{Si}, D_{Ti})$.

Proof: In proving this lemma, we assume that after encoding, the resulting skeletons are inside the original shape boundary. This is an assumption which holds true is almost all cases and rates of interest. Let the x-coordinates of the boundary points at the skeleton at location $y=i$ be given by x_1 and x_2. With the given skeleton distortion D_{Si} and distance distortion D_{Ti}, there are the following four possibilities for the location of the x-coordinates pair of the reconstructed boundary points: $(x_1-D_{Si}-D_{Ti}, x_2- D_{Si} + D_{Ti})$, $(x_1- D_{Si} + D_{Ti}, x_2- D_{Si} - D_{Ti})$, $(x_1+ D_{Si} - D_{Ti}, x_2+ D_{Si} + D_{Ti})$, $(x_1+D_{Si} + D_{Ti}, x_2+ D_{Si} - D_{Ti})$. Based on the MPEG-4 distortion metric, all four possibilities have the same resulting distortion given by:

$$D=\left|D_{Si} - D_{Ti}\right| + \left|D_{Si} + D_{Ti}\right| = 2\max(D_{Si}, D_{Ti}).$$

Finally, by summing up the distortion in each offset of the y direction for all skeletons, we get

$$D_{tot}(S,T) = 2 \cdot \sum_{i=1}^{N} \max(D_{Si}, D_{Ti})$$

Lemma 4 $D_{tot}(S,T) \le 2 \cdot \sum_{i=1}^{N} \max(D_{Si}, D_{Ti}) \le 2 \cdot \sum_{i=1}^{N} (D_{Si} + D_{Ti})$

Proof: Given x, y $(x \ge 0, y \ge 0)$, it is clear that $\max(x,y) \le x+y$. Therefore,

$$\sum_{i=1}^{N} \max(D_{Si}, D_{Ti}) \le \sum_{i=1}^{N} (D_{Si} + D_{Ti}).$$

From Lemma 3, we have that when the skeletons are inside the shape boundary, $D_{tot}(S,T) = 2 \cdot \sum_{i=1}^{N} \max(D_{Si}, D_{Ti})$. We now prove that the last expression with inequality for those skeleton points outside the shape boundary. Without loss of generality, we assume that the skeleton is on the right side of the shape boundary; then, there are only two possible cases as shown in Fig. 6. One is the overlapping object case shown in Fig. 6(a), and the other is the non-overlapping object case shown in Fig. 6(b). For the first case, if we denote that the x-coordinates of the boundary points on the left object by x_1

and x_2, the x-coordinates of the boundary points on the right object are then either $(x_1+D_{Si}-D_{Ti}, x_2+ D_{Si} + D_{Ti})$ or $(x_1+ D_{Si} + D_{Ti}, x_2+ D_{Si} - D_{Ti})$. The distortion is given by

$$D=\left|D_{Si}-D_{Ti}\right|+\left|D_{Si}+D_{Ti}\right|=2\max\,(D_{Si},D_{Ti})$$

For the case of Fig. 6(b), it is obvious that

$$D<2D_{Si}\leqslant 2\max\,(D_{Si},D_{Ti})$$

Therefore, by summing up all the distortions, we obtain

$$D_{tot}(S,T)\leq 2\cdot\sum_{i=1}^{N}\max(D_{Si},D_{Ti})$$

3.2 Bit Rate

Let us denote by R_{tot} the total available bit rate for the encoding of an object shape. Then $R_{tot}(S,T)=R_0+R(S)+R(T)$, where R_0 represents the bits required for the encoding of the starting point of the skeleton, $R(S)$ the bits allocated to the encoding of the skeleton signal, and $R(T)$ the bits allocated to the encoding of the distance signal.

We denote by $V_S=\{V_{S1}, V_{S2}, \ldots\ldots V_{SM}\}$ the set of vertices used for the approximation of the skeleton; with M the number of vertices. Let us denote by $r\,(V_{Si}, \ldots, V_{S(i-o)})$, the rate required for the encoding of the vertex V_{Si}, based on a pre-assigned encoding scheme, when a curve of a certain order o is used for the approximation. For example, if straight lines are used for the approximation, two control points are needed to define a line segment and the order o equals 1; if on the other hand, second order curves are used, such as 2nd order B-Splines, three control points are needed to define a curve segment and the order o equals 2. Then, $R(S)=\sum_{i=1}^{M} r(V_{Si},\ldots,V_{S(i-o)})$. Similarly, the set of vertices defining the approximation of the distance signals is represented by V_T, and $R(T)=\sum_{i=1}^{Q} r(V_{Ti},\ldots,V_{T(i-o)})$, where $r(V_{Ti},\ldots,V_{T(i-o)})$ represents the rate for encoding the ith control point $(i=1,\ldots Q)$ of the approximating curve. Therefore,

$$R_{tot}=R_0+\sum_{i=1}^{M} r(V_{Si},\ldots,V_{S(i-o)})+\sum_{i=1}^{Q} r(V_{Ti},\ldots,V_{T(i-o)}). \quad (4)$$

3.3 Problem Formulation

With the notation and the quantities introduced in the previous section, we now proceed with the formulation of the problem we are set to solve. Given a skeleton signal and the corresponding distance signal, we want to determine, the vertices $\{V_{Si}\}\,(1 \le i \le M)$ of the curve providing an approximation to the skeleton signal, and the vertices $\{V_{Tj}\}\,(1 \le j \le Q)$ of the curve providing an approximation to the distance signal, as well as, the number of skeleton signal vertices M and distance signal vertices Q, which minimize the overall distortion of the reconstructed shape, $D_{tot}(S,T)$, utilizing $R_{tot}(S,T)$ bits which satisfy a bit budget of R_{max} bits. That is, the problem to be solved is given by

$$\min_{V_S, V_T} D_{tot}(S,T), \text{ subject to } R_{tot}(S,T) \le R_{max}. \tag{5}$$

Let $R_{budget}=R_{max}-R_0$, that is the bit budget available after the encoding of the starting point. By utilizing Lemma 3 and Eq. (4), problem (5) can be rewritten as:

$$\min_{V_S, V_T} \sum_{i=1}^{N} \max(D_{Si}, D_{Ti}) \text{ , subject to } \sum_{i=1}^{M} r(V_{Si},...,V_{S(i-o)}) + \sum_{i=1}^{Q} r(V_{Ti},...,V_{T(i-o)}) \le R_{budget}. \tag{6}$$

4. Solution method

In this section, we introduce both optimal and near optimal solutions to the problem for various requirements from different applications. The optimal solution to be derived in section 4.1 is based on the Lagrange multiplier method and dynamic programming adopted to solve a DAG shortest path problem. To speed up the algorithm, two suboptimal algorithms are proposed in section 4.3.

4.1. Optimal solution

We derive a solution to problem (6) using the Lagrange multiplier method to relax the constraint, so that the relaxed problem can be solved using a shortest path algorithm. We first define the Lagrangian cost function

$$J_\lambda(V_S,V_T) = \sum_{i=1}^{N} \max(D_{Si}, D_{Ti}) + \lambda \{\sum_{i=1}^{M} r(V_{Si},...,V_{S(i-o)}) + \sum_{i=1}^{Q} r(V_{Ti},...,V_{T(i-o)})\}, \tag{7}$$

where λ is called the Lagrange multiplier. It has been shown in [22] and [23] that if there is a λ^* such that $\{V_S^*, V_T^*\}=arg \min_{V_S, V_T} J_{\lambda^*}(V_S, V_T)$, and which leads to $\sum_{i=1}^{M} r(V_{Si},...,V_{S(i-o)}) + \sum_{i=1}^{Q} r(V_{Ti},...,V_{T(i-o)}) = R_{budget}$, then $\{V_S^*, V_T^*\}$ is also an optimal solution to (6). It is well known that when λ sweeps from zero to infinity, the solution to problem (6) traces out the convex hull of the operational rate distortion function, which is a nonincreasing function. Hence, bisection or the fast convex search we present in [24] can be used to find λ^*. Therefore, if we can find the optimal solution to the unconstrained problem

$$\min \sum_{i=1}^{N} \max(D_{Si}, D_{Ti}) + \lambda \{\sum_{i=1}^{M} r(V_{Si},...,V_{S(i-o)}) + \sum_{i=1}^{Q} r(V_{Ti},...,V_{T(i-o)})\}, \quad (8)$$

we can find the optimal λ^* and the convex hull approximation to the constrained problem of (6).

Clearly, if the skeleton is encoded without loss, while the distance signal with loss, then based on Lemmas 1 and 2, problem (8) becomes the minimization of $\left\{\sum_{i=1}^{N} D_{Ti} + \lambda \sum_{i=1}^{Q} r(V_{Ti},...,V_{T(i-o)})\right\}$. Similarly, for the lossy encoding of the skeleton signals with lossless distance signals, optimization (8) becomes the minimization of $\left\{\sum_{i=1}^{N} D_{Si} + \lambda \sum_{i=1}^{M} r(V_{Si},...,V_{S(i-o)})\right\}$. The last two optimizations can be solved using a DAG (Directed Acyclic Graph) shortest path algorithm proposed in [8-10,25]. We proceed next with the development of an algorithm for solving optimization (8).

4.2 The 4D DAG shortest path algorithm

The optimization problem of Eq. (8) is more complicated since the skeleton and distance signals are coupled through the distortion. We are expanding the DAG shortest path algorithm developed in [8-10,25] to a 4D space to solve this problem. In other words, the states in the DAG state space will be represented by 4-tuple elements instead of 1-tuple elements. In the following, we are showing how to map the original problem (8) into a graph theory problem. We will start from the simplest case when the control points are restricted to belong to the original signal set (skeleton or distance data). In general we utilize a set of

points outside the original signal set as the admissible control point set, as was done in [8,9]. This set typically forms a band of a certain width around the original signal set. The simplest case we study next results when the width of this control point band is set equal to zero.

A. Node space

Given a polygonal approximation of both skeleton and distance signals, we define a node space with elements the 4-tuple *(i,j,p,q)*, representing all combinations of the last two control points in the skeleton approximation *(i)* and *(p)* ($i \leqslant p$), and the last two control points in the distance signal approximation *(j)* and *(q)* ($j \leqslant q$). Clearly, there is one node space for each possible approximation. For the skeleton approximation with M vertices and the distance approximation with Q vertices, the size (number of nodes) of the space is equal to $\frac{M(M+1)Q(Q+1)}{4}$ (see Fig. 7 for an example).

Let s denote the next vertex after p in the skeleton approximation and t the next vertex after q in the distance approximation. There are only three links starting at node *(i,j,p,q)* to describe the transitions *(i,j,p,q)* $\rightarrow$ *(p,j,s,q)*, *(i,j,p,q)* $\rightarrow$ *(i,q,p,t)*, and *(i,j,p,q)* $\rightarrow$ *(p,q,s,t)*, indicated by bold arrows in Fig. 8. The above defined node space is providing an alternative (and more useful) way in representing an approximation of the distance and skeleton data. Consider for example, the approximation shown in Fig. 8(a). A 9-point skeleton and distance signals are shown (open circles), along with the vertices of the polygonal approximations (filled circles). The approximation of the skeleton signal consists of the use of the samples numbered 0,1,3 and 8, as vertices of the polygon, and for the distance data of the samples numbered 0,3,5 and 8. The approximation, in other words, consists of these two lists of vertex points. An alternative way to represent their approximation is with the use of the node space, that is, by the linked node list (0,0,1,3) $\rightarrow$ (1,3,3,5) $\rightarrow$ (3,5,8,8). We therefore define a mapping between the vertex lists and the linked node list. As can be easily understood, this mapping is not 1-to-1 but 1-to-η. For example, for the given vertex lists, another linked node list is (0,0,1,0) $\rightarrow$ (1,0,3,3) $\rightarrow$ (3,3,8,5) $\rightarrow$ (8,5,8,8). Actually, η can become a very large number as M and Q increases, since it is an exponential function of M and Q.

B. State space

In order to be able to use dynamic programming for obtaining the optimal solution of problem (8), we define a state space, which is a subset of the union of all node spaces, with elements (so called states) *(i,j,p,q)* satisfying $i \leqslant q$ and $j \leqslant p$, and edges among elements. This definition excludes from consideration those nodes *(i,j,p,q)* with segment *[i,p]* not overlapping with segment *[j,q]*. The motivation for this is twofold: 1) By removing the non-overlapping segments, we can express the distortion as the sum of link distortions between states, as will be shown later. This will make a dynamic programming solution possible. 2) The fewer the states the faster the algorithm, given we do not remove from consideration any feasible polygonal approximations with this pruning.

There are only two kinds of edges allowed starting at state *(i,j,p,q)*, which correspond to the first two kinds of links in node space, as shown in Fig. 8(a) and Fig. 8(b). In other words, the two edges describe the transitions *(i,j,p,q)->(p,j,s,q),* and *(i,j,p,q)->(i,q,p,t)* respectively. Therefore, the total number of edges is $O(N^5)$, where N the total number of points in the skeleton and distance data. It is important to note that excluding the third possibility in Fig. 8(c) does not exclude any optimal path, since, as shown later, any possible approximation can be achieved using only the first 2 possibilities (Fig. 8(c) can be obtained by the concatenation of Figs. 8(a) and 8(b)). This restriction is of considerable help when later we define edge distortion, so that the total distortion is the sum of edge distortions.

We now prove that the exclusion process (of those nodes and links represented in Fig. 8(c)) will not remove any optimal solutions for problem (8), by demonstrating that there is an 1-to-K mapping between the vertex lists and the linked state list (K is a variable nonzero integer considerably smaller than η). First, we show that for any linked state list starting at state *(0,0,0,0)* and terminating at state *(N,N,N,N),* there are corresponding vertex lists. Based on the definition of edges, and the edge terminating at state *(N,N,N,N)*, we can find the last past state of state *(N,N,N,N)*, and therefore the past vertex of the last point (N) in the skeleton approximation, or the past vertex of the last point (N) in the distance approximation. Then by backtracking, vertex points in the skeleton approximation or the distance approximation are

recorded, until state *(0,0,0,0)* is reached and the vertex lists of the approximations for both skeleton and distance signals are completed.

Then, we provide a generic procedure to obtain a corresponding linked state list from any given vertex lists. The idea is straightforward. We start by including state *(0,0,0,0)* into an empty list, (as we know, $y(V_{S1}) = y(V_{T1}) = 0$, where $y(V)$ is the vertical index of vertex V in the original signal set). Then state $(0,0, y(V_{S2}), 0)$ is appended into the list, followed by state $(0,0, y(V_{S2}), y(V_{T2}))$. Then, the value of $y(V_{S2})$ is compared with $y(V_{T2})$. If $y(V_{S2}) \leqslant y(V_{T2})$, then the next state is $(y(V_{S2}),0, y(V_{S3}), y(V_{T2}))$, otherwise, the next state is $(0, y(V_{T2}), y(V_{S2}), y(V_{T3}))$. After that, $y(V_{S3})$ is compared with $y(V_{T2})$, (or $y(V_{S2})$ is compared with $y(V_{T3})$), and the state transition process proceed iteratively. Eventually, the state transition will reach state $(y(V_{S(M-1)}), y(V_{T(Q-1)}), N, N)$ (as we know, $y(V_{SM}) = y(V_{TQ}) = N$). Figure 9 shows an example of the procedure. In Fig. 9 (a), the sequence of steps is labeled and Fig. 9(b) shows the list of the state transitions. From the iterative procedure for obtaining the linked state list (mapping Fig. 9(a) to 9(b)) and the backtracking procedure (mapping Fig. 9(b) to 9(a)), described above, it is clear that there is an 1-to-*K* mapping between the vertex lists and the linked state list. This means that the specific definition of the state space introduced in this section will not cause the exclusion of any optimal paths, since every possible skeleton and distance approximations are maintained.

C. Dynamic Programming

To implement the algorithm to solve the optimization problem (8), we create a cost function $C(p_k)$ (assuming p_k is representing state (i,j,p,q)), which represents the minimum total rate and distortion up to and including state (i,j,p,q) in the state space. That is,

$$C(p_k) = \min\left\{ \sum_{i=1}^{\min(p,q)} \max(D_{Si}, D_{Ti}) + \lambda\left(\sum_{i=1}^{y_s^{-1}(p)} r(V_{Si},\ldots,V_{S(i-o)}) + \sum_{i=1}^{y_d^{-1}(q)} r(V_{Ti},\ldots,V_{T(i-o)}) \right) \right\}, \tag{9}$$

where the function $y_s^{-1}(x)=t$ iff $y(V_{St})=x$, and $y_d^{-1}(x)=t$ iff $y(V_{Tt})=x$. In the following, we assume that the order of the approximating curve is equal to 1, i.e., $o=1$. For $o>1$, we need to construct a state space containing $(2o+2)$-tuple states, instead of 4-tuple states.

The key observation for deriving an efficient algorithm is the fact that given a certain state (p_{k-1}) in a path and the cost function up to and including this state ($C(p_{k-1})$), the selection of the next state p_k is independent of the selection of the previous states $p_0,p_1,\ldots p_{k-2}$. This is true since the cost function can be expressed recursively as a function of the segment rates $\zeta(p_{k-1},p_k)$ and the segment distortion $d(p_{k-1},p_k)$. More specifically,

$$C(p_k)= min\,(C(p_{k-1})+w(p_{k-1},p_k)) \tag{10}$$

where

$$w(p_{k-1},p_k)=d(p_{k-1},p_k)+\lambda\zeta(p_{k-1},p_k) \tag{11}$$

and
$$d(p_{k-1},p_k)=\begin{cases}\sum_{t=i+1}^{q}\max(D_{St},D_{Tt}) & \text{Transition occurs in skeleton data}\\ \sum_{t=j+1}^{p}\max(D_{St},D_{Tt}) & \text{Transition occurs in distance data}\end{cases} \tag{12}$$

and
$$\zeta(p_{k-1},p_k)=\begin{cases}r(y_s^{-1}(p),y_s^{-1}(i)) & \text{Transition occurs in skeleton data}\\ r(y_d^{-1}(q),y_d^{-1}(j)) & \text{Transition occurs in distance data}\end{cases} \tag{13}$$

Figure 10 shows an example of the segment distortion and segment rate. The figure on the right (p_{k-1} represents state *(i,n,p,j)*) shows the next step relative to the figure on the left (p_{k-1} represents state *(m,j,i,q)*). It is easy to see how the edge distortions add up to the total distortion. In other words, we are showing that summing the above segment distortions up will result in the total distortion and, that these segment distortions are only dependent on state p_{k-1} and state p_k.

Recursion (10) needs to be initialized by setting $C(0,0,0,0)$ equal to zero. For all possible approximations, $C(y(V_{SM}),y(V_{TQ}),N,N)$ is equal to the minimum value of $\sum_{i=1}^{N}\max(D_{Si},D_{Ti})+\lambda(\sum_{i=1}^{M}r(V_{Si},V_{S(i-1)})+\sum_{i=1}^{Q}r(V_{Ti},V_{T(i-1)}))$, the Lagrangian cost function for the entire skeleton and distance data.

Using (10), the problem stated in (8) can be formulated as a shortest path problem in a weighted directed graph $G=(V,E)$, where V is the set of graph vertices and E the set of edges, for G the state space, where

any directed edge in G from a vertex to another vertex represents a state transition. A directed edge is denoted by the ordered pair $(u,v) \in E$, which implies that the edge starts at vertex u and ends at vertex v. The weight of edge (u,v) is defined as $w(u,v)$ in (11). Clearly, the problem state in (8) can be solved by finding the shortest path from the vertex corresponding to the state $(0,0,0,0)$ to the vertex corresponding to the state (N,N,N,N), and the ordered list of vertices on the shortest path correspond to an ordered state sequence. By backtracking we get the vertex lists for skeleton and distance signals.

In summary, the state definition and the recursive representation of the cost function in (10) makes the future step of the optimization process independent from its past step, which is the foundation of the dynamic programming technique. The computational complexity of a DAG shortest path DP algorithm is $O(|V|+|E|)$. For the graph G corresponding to the state space, $|V|$ is $O(N^4)$ and $|E|$ is $O(N^5)$, therefore, the computational complexity of the proposed 4D DAG shortest path algorithm is $O(N^5)$. In most cases, it only takes several iterations to find the optimal lambda. So, comparing with N^5, the number of iterations is not a significant factor in consideration of computational complexity.

D. Admissible control point band

To handle the case of a nonzero admissible control point band (i.e., the control points can be non-boundary points) and make the optimal solution still feasible, we need to take the following steps. First, we have to label the admissible control points in a line one by one from left to right and repeat this process line by line from top to bottom in the vertical direction (shown in Fig. 11), so that point i is on the left of point j $(i \leqslant j)$ if they are on the same horizontal line, otherwise, point i is on the line above the line containing point j. The state space is constructed with those states (i,j,p,q) that combine the ith and pth admissible points for skeleton signal, and the jth and qth admissible points for distance signal, with the restrictions that $i \leqslant p$, $i \leqslant q$, $j \leqslant p$ and $j \leqslant q$, and the ith point is not on the same line with the pth point, and also the jth point is not on the same line with the qth point. The weighted directed edges from one state to another state are defined the same way as above. Clearly, the mapping between state space and graph G

exists, and dynamic programming is applicable. Therefore, the optimal solution can be found by the DAG shortest path algorithm.

4.3. Suboptimal algorithms

A. Greedy-trellis search

A greedy suboptimal approach with computational complexity of $O(N^3)$ can be applied to the problem at hand, with no major loss in the quality of the results. Such an approach is obtained by using a 2-tuple state *(i,j)* instead of a 4-tuple state, which consists of the last point in the skeleton approximation *(i)*, and the last point in the distance approximation *(j)*. As shown in Fig. 12, the state space consists of states and weighted directed edges from one state to another. However, the terminating states *(p,q)* of those edges starting at state *(i,j)* are a restricted subset of the whole states set that satisfying $p \geqslant i$ and $q \geqslant j$, resulting in an acyclic graph. The greedy approach keeps the lowest cost branch at all stages in the trellis up to that point, i.e., the selection of the optimal state at the current stage is forced based on the optimal state at the previous stage. So, every state keeps the shortest path from the source state *(0,0)* to itself. The edge weight from state *(i,j)* to state *(p,q)* is defined as

$$w(i,j,p,q) = \sum_{k=\min(i,j)}^{\min(p,q)} \max(D_{Sk}, D_{Tk}) + \lambda(r(y^{-1}(p)) + r(y^{-1}(q))). \tag{14}$$

Obviously, *w(i,j,p,q)* could depend on the states in the past of the state *(i,j)* in the so far selected optimal path, which destroys the optimality of the algorithm, and makes this approach a greedy search algorithm. To speed up the performance, pruning can be implemented by only allowing those edges from state *(i,j)* to state *(i,q) (*$q \geqslant j$*)* or state *(p,j) (*$p \geqslant i$*)*. The total number of edges in the state space then becomes equal to $O(N^3)$.

B. Relaxed distortion optimal solution

Various ways can be envisioned for obtaining suboptimal results. For example, as mentioned in section 4.1, lossy encoding can be performed only on one of the two data sets, while the other is encoded losslessly. An alternative way [26] is to apply polygonal approximation on both the skeleton and distance

data, while assuming that the distortion from the skeleton approximation and the distortion from the distance approximation are additive, although, as shown earlier, they are not.

In this approach, we assume that $D_{tot} \approx 2 \cdot \sum_{i=1}^{N} (D_{Si} + D_{Ti})$. Based on Lemma 4, it is clear that the resulting actual distortion in this case can be smaller that the one calculated by the algorithm. Problem (8) in this case can be written as

$$min \left\{ \sum_{i=1}^{N} (D_{Si} + D_{Di}) + \lambda (\sum_{i=1}^{M} r(V_{Si},...,V_{S(i-o)}) + \sum_{i=1}^{Q} r(V_{Ti},...,V_{T(i-o)})) \right\} \quad (15)$$

Since skeleton and distance data are independent, (15) can be split into 2 optimizations as mentioned in section 4.1, with computational complexity of $O(N^2)$.

Another way to improve the efficiency of the algorithm with near optimality is to use a sliding window [27], which restricts the maximum vertical distance of consecutive starting/ending vertices. By denoting by W_{win} the window size, the computational complexity of the 4D DAG shortest path algorithm will be reduced to $O(N^4 W_{win})$, and the near optimal algorithm mentioned above will have computational complexity equal to $O(NW_{win})$. Experimental results indicate that when W_{win} is increased, the RD curve improves slightly, at the expense of considerably increased execution time.

The width of the admissible control point band is another factor influencing the efficiency of the algorithm. The wider the band, the larger the number of admissible control points N.

5. General cases

We now consider the more general cases where an object contains more than one skeletons, and there are multiple objects in a frame.

A. Object with multiple skeletons

Assume that the L skeletons $\{S^1, S^2, ..., S^L\}$ are encoded with the associated distance signals $\{T^1, T^2, ..., T^L\}$. Then the optimization problem can be stated as

$$\min\ D_{tot}(S^1,S^2,...,S^L,T^1,T^2,...,T^L)\text{, subject to }\sum_{m=1}^{L}\{R(S^m)+R(T^m)\}\le R_{budget} \tag{16}$$

where the total distortion of the object depends on all L skeletons and distances. This problem can be solved by solving the unconstrained problem first

$$\min\ \{D_{tot}(S^1,S^2,...,S^L,T^1,T^2,...,T^L)+\lambda\cdot\sum_{m=1}^{L}[R(S^m)+R(T^m)]\} \tag{17}$$

It can be easily shown that

$$D_{tot}(S^1,S^2,...,S^L,T^1,T^2,...,T^L)\le\sum_{i=1}^{L}D_{tot}(S^i,T^i) \tag{18}$$

In most cases, equality holds, and then the optimization problem decouples into L problems of $\min\ \{D_{tot}(S^m,T^m)+\lambda\,[R(S^m)+R(T^m)]\}\ (1\leqslant m\leqslant L)$, which are identical to problem (8) solved in the previous section. In some rare cases, when the distance distortion pixel sets corresponding to two skeletons overlap, the total distortion is less than the sum of distortions in Eq. (18). In such cases, solving the decoupling problem yields a suboptimal overall solution. Since these cases are rare and the solution of the overall problem becomes extremely complicated, we will not consider it here.

B. Multiple object boundary encoding

Since the distortion calculation is defined on an object-by-object basis, the problem remains decoupled even if the distortion pixel sets of two objects overlap. That is, the results of section 4 apply. It is underlined here that in the cases of decoupled optimizations, the same Lagrange multiplier is used for the optimization of all the relaxed problems.

6. Experimental Result

A number of experiments have been conducted, some of which are reported here. Figure 13 shows a comparison of the results obtained using the greedy-trellis search suboptimal approach and the 4D DAG optimal approach. 100 frames of the SIF sequence Kids were used in this experiment. The distortion axis

represents the average of MPEG-4 distortion (D_{MPEG-4} in Eq. (3) for one frame over 100 frames). The results are comparable, with a great reduction in the computational complexity of the greedy algorithm. In this second experiment, the relaxed distortion optimal solution is compared to the 4D DAG optimal solution. Figure 14 shows the corresponding ORD curves for the first frame of the Kids sequence. In addition, the result obtained by the relaxed distortion optimal solution is evaluated using the MPEG-4 distortion metric, showing some improved performance. As a general conclusion, the suboptimal algorithms studied demonstrate a performance quite comparable to that of the optimal algorithm.

Some of the experiments address the suitability of various new compression techniques for compressing each of the skeleton and distance signals. We compared the methods of encoding skeleton signals without loss while encoding distance signal with loss (Fig. 15), and the method of encoding both skeleton and distance signals with loss (Fig. 16). As expected, by introducing distortion on the skeleton, the rate is reduced by about 40% for distortion around 4% (compare left figure of Fig. 15 to right figure of Fig. 16).

Another experiment is reported in Fig. 17. In this experiment, both the skeleton and distance signals are encoded using polygonal approximation, VLC optimization is applied as in [25], and the 4D DAG shortest path algorithm is applied to get the ORD optimal curves. The results of the optimal approximation are indicated by "*". In addition, in Fig. 17, the results obtained in [15] for a polygonal approximation are shown, indicated by "o", and the results obtained by the CAE method are shown, indicated by "□". As it can be inferred from Fig. 17, the decomposition of the boundary data into two signal data sets (skeleton and distance), with different characteristics, allows for their efficient exploitation resulting in better compression results.

The effectiveness of the proposed approach was also tested utilizing 100 frames of the CIF "Cyclamen" and the QCIF "Weather" sequences. In Fig. 18 the RD curves for the proposed algorithm and CAE are shown, while in Fig. 19 a representative original and compressed frames from the "Cyclamen" sequence are shown. The proposed algorithm outperforms CAE with this sequence as well, which is a challenging sequence for the proposed method. Finally three RD curves are shown for the "Weather" sequences. The proposed method greatly outperforms CAE and an RD optimal vertex method in this case, since the

"Weather" sequence with typically one skeleton per frame is extremely well suited for the skeleton decomposition.

7. Conclusion

In this paper, we presented a skeleton-based shape-coding algorithm. By decoupling the two-dimensional shape object data into one-dimensional skeleton and distance signals, we create a novel scheme for encoding the object boundary. Since the separated skeleton and distance signals are uncorrelated with each other, this encoding method is expected to be efficient. By allowing arbitrary order of curves to represent the approximation form of the skeleton and distance signals, we are facing the problem of how to choose the number and location of the control points of the approximation curves for both skeleton and distance signals, to minimize the overall distortion with a given bit budget for a video frame. In the paper, several solutions are proposed to solve the problem. An optimal solution using the Lagrangian multiplier method and 4D DAG shortest path algorithm is presented with computational complexity of $O(N^5)$. To reduce complexity without a major sacrifice of the quality, a suboptimal greedy-trellis search algorithm is demonstrated with computational complexity of $O(N^3)$. Furthermore, near optimal solutions are proposed reducing the computational complexity to $O(N^2)$. Experimental results demonstrate that the proposed algorithms result in a significant improvement in rate-distortion efficiency with respect to other ORD optimal shape encoders.

Reference

[1] R. Koenen, "MPEG-4 multimedia for our time", *IEEE Spectrum*, vol. 36, pp. 26-33, Feb. 1999.

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

[3] A. K. Katsaggelos, L. Kondi, F. W. Meier, J. Ostermann, and G. M. Schuster, "MPEG-4 and rate distortion based shape coding techniques", *Proc. IEEE*, pp.1126-1154, June 1998.

[4] M. Hotter, "Object-oriented analysis-synthesis coding based on moving two-dimensional objects", *Signal Processing: Image Commun.*, vol. 2, pp. 409-428, Dec. 1990.

[5] P. Gerken, "Object-based analysis-synthesis coding of image sequences at very low bit rates", *IEEE Trans. Circuits Syst. Video Technol.*, vol.4, pp. 228-235, June 1994.

[6] K. J. O'Connell, "Object-adaptive vertex-based shape coding method", *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, pp. 251-255, Feb. 1997.

[7] S. Lee, D. Cho, Y. Cho, S. Son, E. Jang, J. Shin and Y. Seo, "Binary shape coding using baseline-based method", *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, pp.44-58, Feb. 1999.

[8] G. M. Schuster and A. K. Katsaggelos, "An optimal boundary encoding scheme in the rate distortion sense", *IEEE Trans. Image Processing*, vol. 7, pp. 13-26, January 1998.

[9] G. M. Schuster, G. Melnikov, and A. K. Katsaggelos, "Operationally optimal vertext-based shape coding", *IEEE Signal Processing Magazine*, pp. 91-108, Nov. 1998.

[10] G. M. Schuster and A. K. Katsaggelos, *Rate-Distortion Based Video Compression: Optimal Video Frame Compression and Object Boundary Encoding*, Kluwer Academic Publishers, 1997.

[11] T. Pavlidis, "A review of algorithms for shape analysis", *Comput. Graphics Image Process.*, vol. 7, pp. 243-258, 1978.

[12] C. Zahn and R. Roskies, "Fourier descriptors for plane closed curves", *Comput. Graphics Image Process.*, vol. 21, pp. 269-281, 1972.

[13] E. Persoon and K. Fu, "Shape discrimination using fourier descriptors", *IEEE Trans. System Man & Cybernetics*, vol. 7, pp. 170-179, 1977.

[14] B. Kartikeyan and A. Sarkar, "Shape description by time series", *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, pp. 977-984, 1989.

[15] A. Goshtasby, "Description and discrimination of planar shapes using shape matrices", *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 7, pp. 738-743, 1985.

[16] R. J. Prokop and A. P. Reeves, "A survey of moment-based techniques for unoccluded object representation and recognition", *CVGIP: Graphical Models Image Process.*, vol. 54, pp. 438-460, 1992.

[17] H. Blum, "A transformation for extracting new descriptors of shape", In *Models for the Perception of Speech and Visual Form*, (W. Wathen-Dunn, editor), pp. 362-380, Cambridge, Mass., MIT Press, 1967.

[18] J. Serra, *Image Analysis and Mathematical Morphology*, Academic Press, 1982.

[19] P. A. Maragos and R. W. Schafer, "Morphological skeleton representation and coding of binary images", *IEEE Trans. Acoustics, Speech and Signal Processing*, 34(5): 1228-1244, October, 1986.

[20] I. Pitas and A. N. Venetsanopoulos, "Morphological shape decomposition", *IEEE Trans. PAMI*, vol. 12, pp. 38-45, Jan. 1990.

[21] B. C. S. Tom, S. N. Efstratiadis, and A. K. Katsaggelos, "Motion estimation of skeletonized angiographic images using elastic registration", *IEEE Trans. Medical Imaging*, Vol. 13, No.3, pp. 450-460, Sept. 1994.

[22] H. Everett, "Generalized Lagrange multiplier method for solving problems of optimum allocation of resources", *Oper. Res.*, vol. 11, pp. 399-417, 1963.

[23] Y. Shoham and A. Gersho, "Efficient bit allocation for an arbitrary set of quantizers", *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 36, pp. 1445-1453, Sept. 1988.

[24] G. M. Schuster and A. K. Katsaggelos, "Fast and efficient mode and quantizer selection in the rate distortion sense for H.263," in *SPIE Proc. Conf. Visual Communcations and Image Processing*, March 1996, pp. 784-795.

[25] R. Nygaard, G. Melnikov, and A. K. Katsaggelos, "A rate distortion optimal ECG coding algorithm", *IEEE Trans. Biomedical Engineering*, vol. 48, pp. 28-40, Jan. 2001.

[26] H. Wang, A. K. Katsaggelos, T. N. Pappas, "Rate-distortion optimal skeleton-based shape coding", in *Proc. Int. Conf. Image Processing*, Thessaloniki, Greece, Oct. 2001, pp. 1001-1004.

[27] F. W. Meier, G. M. Schuster, and A. K. Katsaggelos, "An efficient shape coding scheme using B-splines which is optimal in the rate-distortion sense," *Signal Processing: Image Communications*, vol. 15(7-8), pp. 685-701, May 2000.

(a) Original video frame (b) Binary Shape information (c) Boundary information

Figure 1. Decoupling of shape and texture information

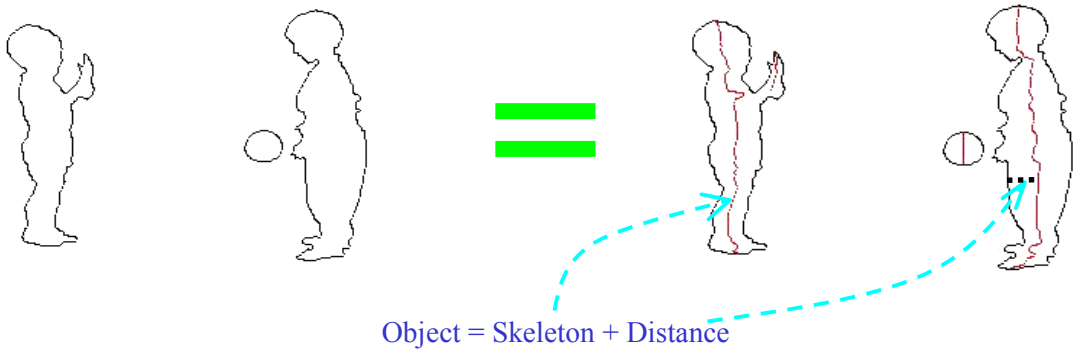
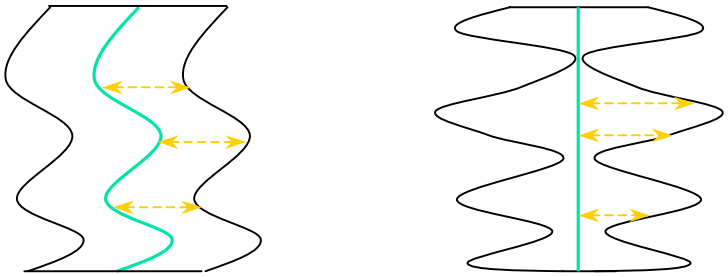
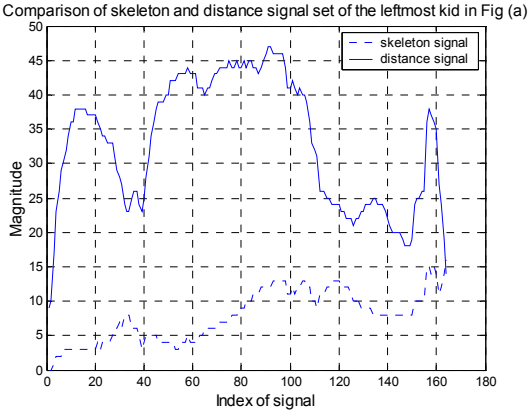


Figure 2. An example of skeletonization

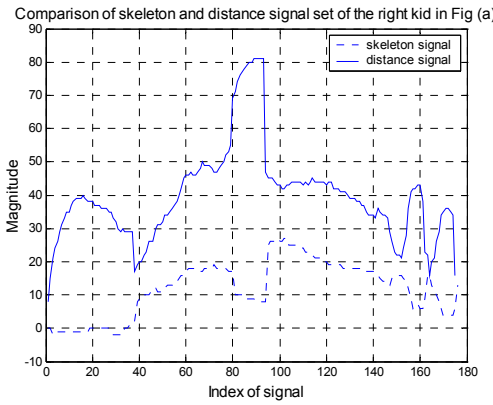


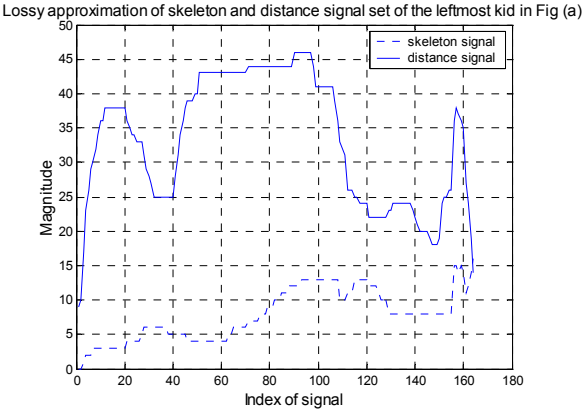
(a) (b)

Figure 3. Highly correlated boundary data, resulting in (a) constant distance signal; (b) constant skeleton signal.



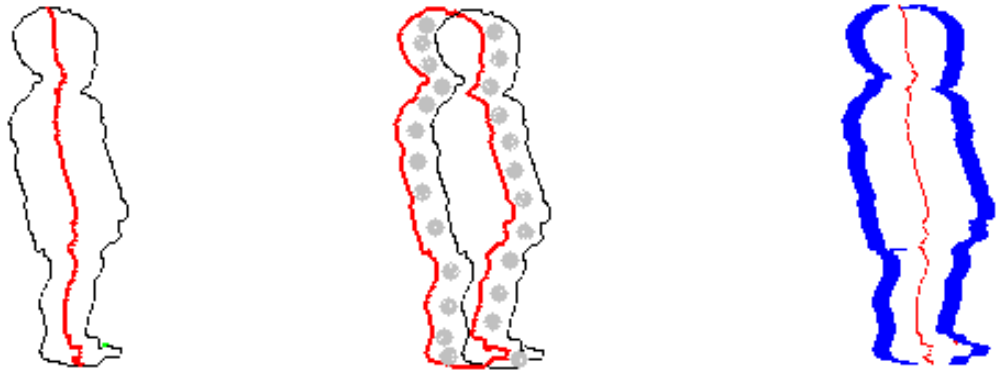
(a) (b)





(c) (d)

Figure 4. Decomposition into skeleton and distance data



(a) (b) (c)

Figure 5 Distortion caused by lossy coding of skeleton or distance

(a. original shape; b. shape after shifting of skeleton; c. shape after shifting of the distance data)

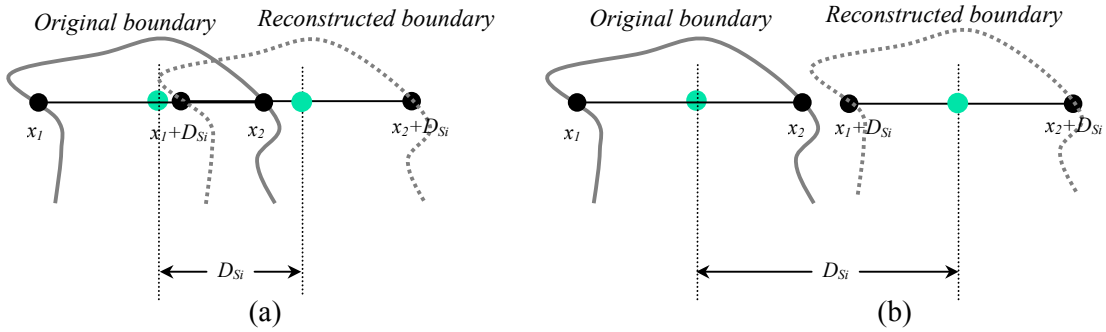


Figure 6 Two cases of the spatial relationship between original and reconstructed objects (assuming $D_{Ti}=0$)

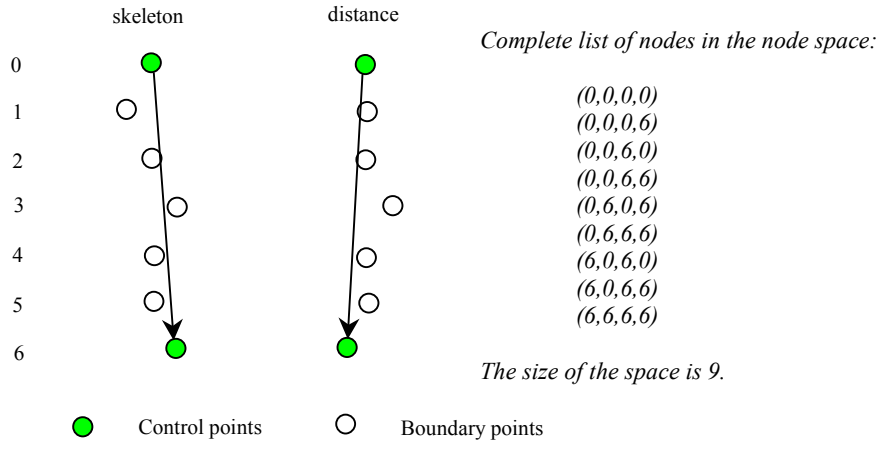
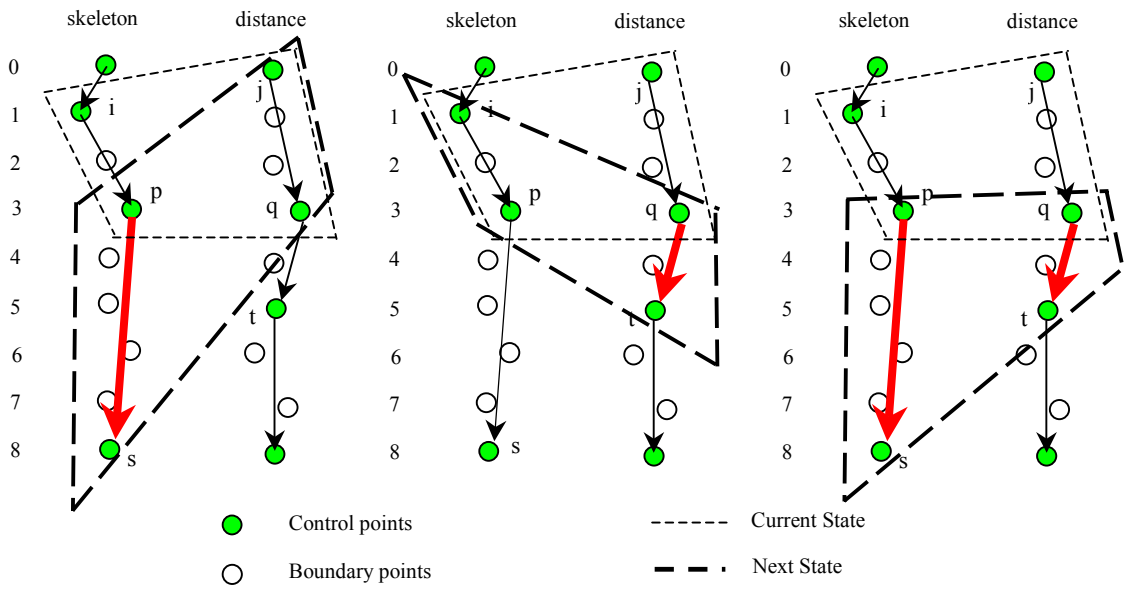
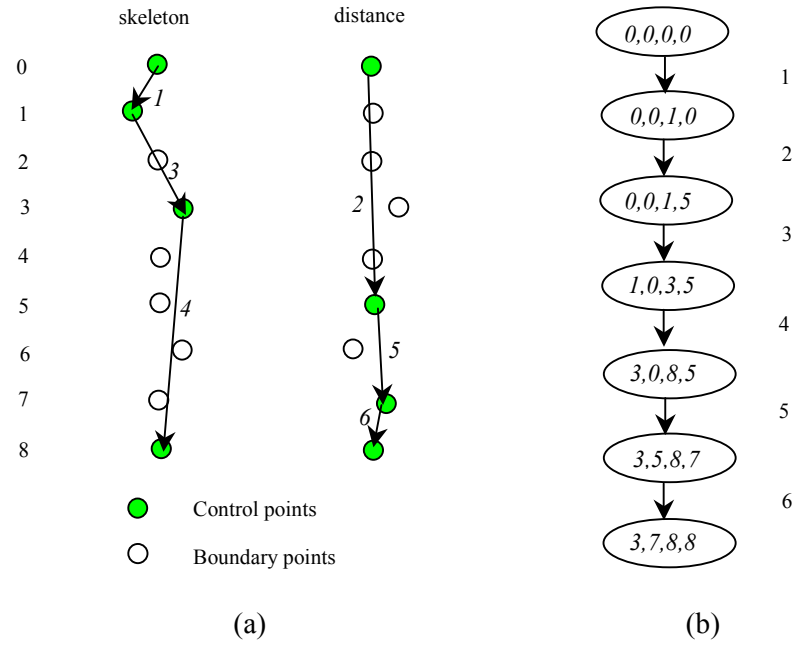


Figure 7. Example of node space



(a) (i,j,p,q)->(p,j,s,q) (b) (i,j,p,q)->(i,q,p,t) (c) (i,j,p,q)->(p,q,s,t)

Figure 8. Examples of connected nodes



(a) (b)

Figure 9. An example of the state transition list (b) corresponding to vertex lists (a)

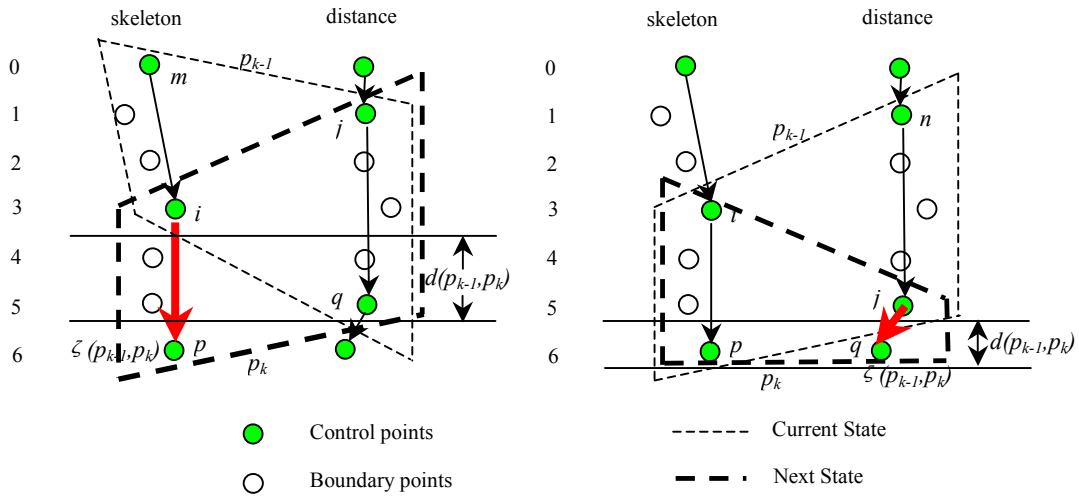


Figure 10. Examples of segment distortion and segment rate

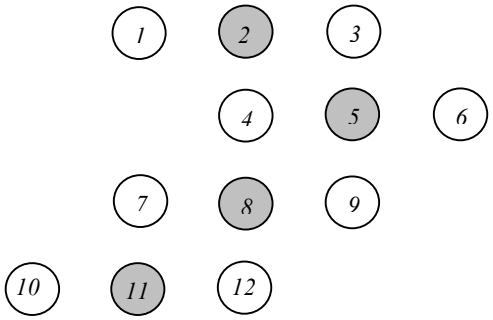


Figure 11. Example of labeling of the admissible control points

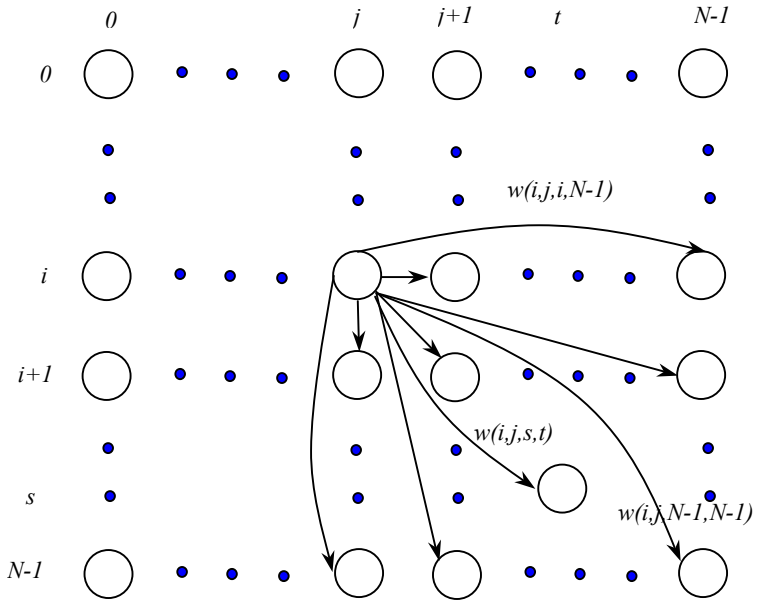


Figure 12. Example of a 2D state space

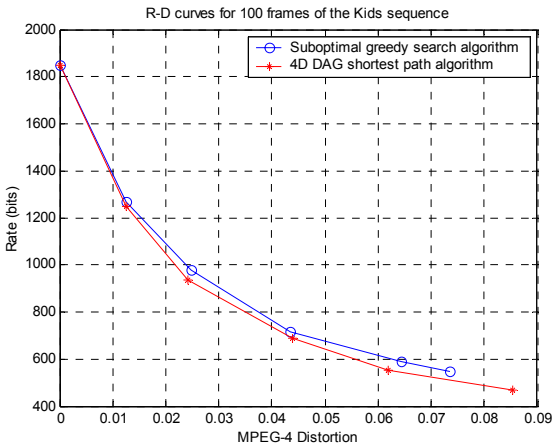


Figure 13. Comparison of optimal and suboptimal (greedy-trellis search) solutions

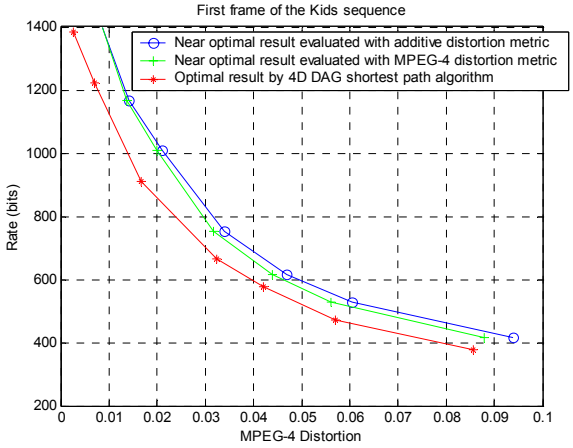
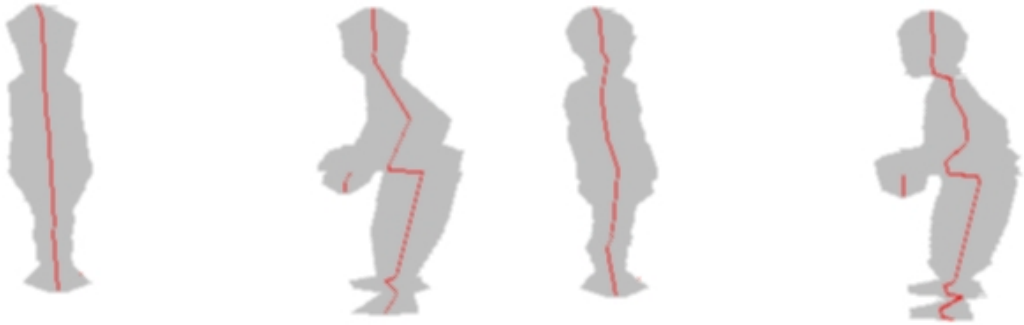


Figure 14. Comparison of RD curve for additive distortion metric and actual MPEG-4 distortion metric obtained by the near-optimal and optimal algorithms

(Rate=976, Distortion=3.9%) (Rate=1080, Distortion=2.35%)

Figure 15. Examples of approximating only distance data



(rate=376, Distortion=8.86%) (rate=592, Distortion=4.1%)

Figure 16. Shape approximation results according to the proposed 4D DAG optimal algorithm

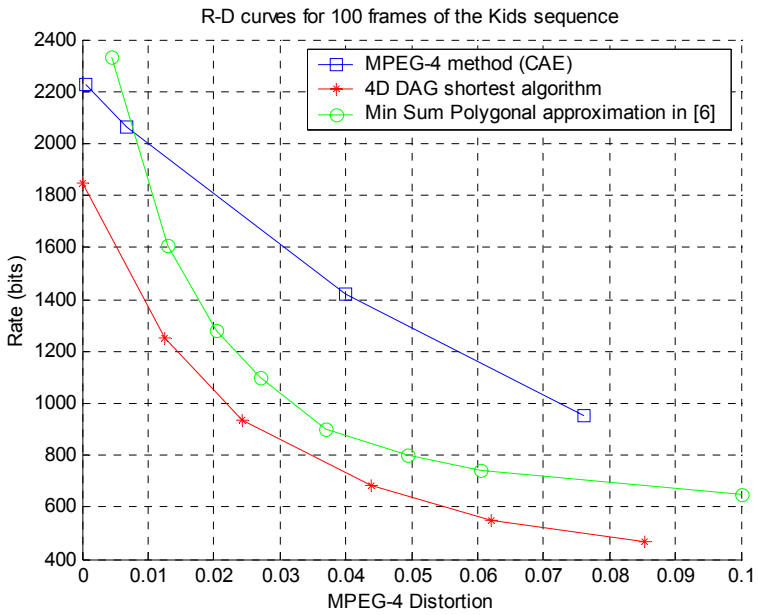


Figure 17. Comparison of RD curves by various algorithms for “Kids” sequence

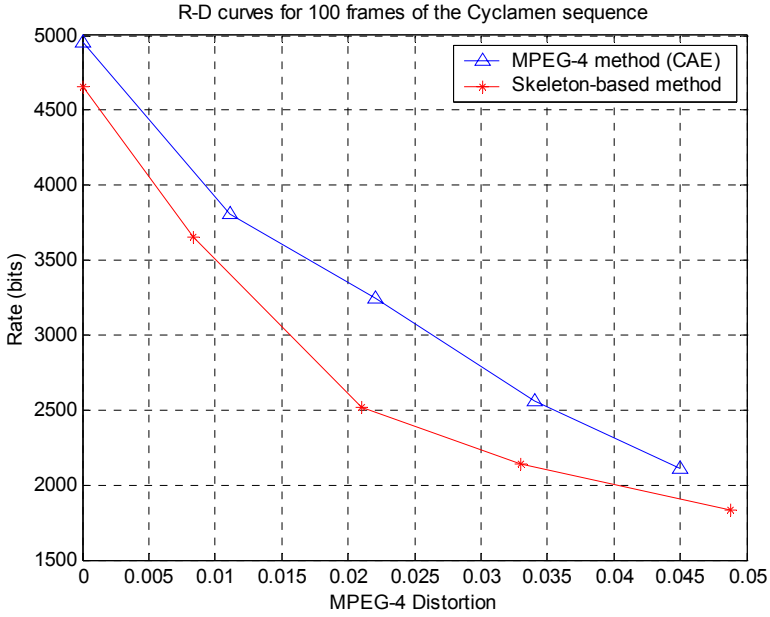


Figure 18. RD curves for the proposed method and CAE for the “Cyclamen” sequence

(a) Original shape with skeleton (b) reconstructed shape (Distortion=6.45%)

Figure 19. Original and reconstructed shapes from Cyclamen sequence

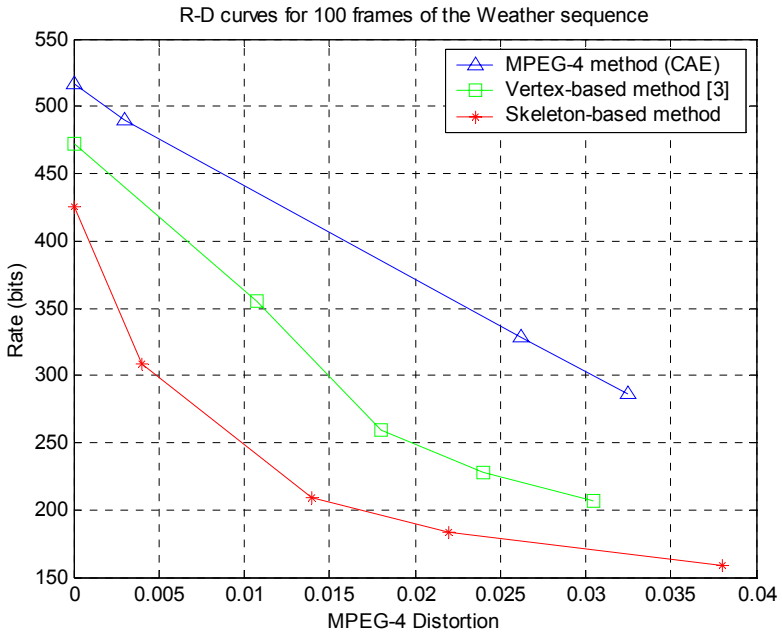


Figure 20. Comparison of RD curves by various algorithms for “Weather” sequence