An Optimal Polygonal Boundary Encoding Scheme in the Rate Distortion Sense

Guido M. Schuster and Aggelos K. Katsaggelos, *Fellow, IEEE*

Abstract—In this paper, we present fast and efficient methods for the lossy encoding of object boundaries that are given as eight-connect chain codes. We approximate the boundary by a polygon, and consider the problem of finding the polygon which leads to the smallest distortion for a given number of bits. We also address the dual problem of finding the polygon which leads to the smallest bit rate for a given distortion. We consider two different classes of distortion measures. The first class is based on the maximum operator and the second class is based on the summation operator. For the first class, we derive a fast and optimal scheme that is based on a shortest path algorithm for a weighted directed acyclic graph. For the second class we propose a solution approach that is based on the Lagrange multiplier method, which uses the above-mentioned shortest path algorithm. Since the Lagrange multiplier method can only find solutions on the convex hull of the operational rate distortion function, we also propose a tree-pruning-based algorithm that can find all the optimal solutions. Finally, we present results of the proposed schemes using objects from the Miss America sequence.

Index Terms— Boundary encoding, dynamic programming, min-max optimization, object-oriented video compression, operational rate distortion theory, shape encoding, tree pruning.

I. INTRODUCTION

THE ENCODING of planar curves is an important problem in many different fields, such as CAD, object recognition, object oriented video coding, etc. This research is motivated by object-oriented video coding, but the developed algorithms can also be used for other applications. A major problem in object-oriented video coding [1], [2] is the efficient encoding of object boundaries. There are two main approaches for encoding the segmentation information: a lossy approach, which is based on a spline approximation of the boundary [2], [3], and a lossless approach, which is based on chain codes [4]. The proposed boundary-encoding scheme is a lossy scheme that can be considered a combination of the spline and the chain code approaches, since the boundary is approximated by a polygon and its vertices are encoded relative to each other. The approximation of the boundary by a polygon is similar to the spline approximation approach, whereas the encoding

of the successive vertices is achieved with a chain code-like scheme.

In [3], B-spline curves are used to approximate a boundary. An optimization procedure is formulated for finding the optimal locations of the control points by minimizing the mean squared error (MSE) between the boundary and the approximation. This is an appropriate objective when the smoothing of the boundary is the main problem. When, however, the resulting control points need to be encoded, the tradeoff between the encoding cost and the resulting distortion needs to be considered. By selecting the MSE as the distortion measure and allowing for the location of the control points to be anywhere on the plane, the resulting optimization problem is continuous and convex and can be solved easily. In order to encode the positions of the resulting control points efficiently, however, one needs to quantize them, and therefore the optimality of the solution is lost. It is well known that the optimal solution to a discrete optimization problem (quantized locations) does not have to be close to the solution of the corresponding continuous problem.

In [2], a boundary is approximated by a combination of splines and polygons, where the distortion measure employed is the maximum distance between the approximation and the original boundary. First the vertices of the polygon are heuristically found, and then these vertices are used to find a spline representation of the boundary. For every segment where the spline representation does not exceed the maximum distance from the original boundary, the spline representation is used instead of the polygon. This leads to a smoother approximation. Again, the approach is well suited for the smoothing of the original boundary and there is also an inherent control over the maximum distance between the approximation and the original boundary, but the resulting rate is not taken explicitly into account. If each control point is encoded using a fixed-length codeword, then minimizing the number of control points is equivalent to minimizing the resulting bit rate. On the other hand, if the control points are along a natural boundary, there exists a high correlation between the location of two consecutive points, and a predictive encoding scheme should be employed instead of a fixed-length codeword scheme. In any case, this approach does not facilitate the rate constrained encoding of a boundary, i.e., the rate is given and one wants to find the best approximation.

Freeman [4] originally proposed the use of chain coding for boundary quantization and encoding, which has attracted considerable attention over the last thirty years [5]–[9]. The most common chain code is the eight-connect chain code

G. M. Schuster is with the Advanced Technologies Research Center, Carrier Systems, 3COM, Mount Prospect, IL 60056-2293 USA (e-mail: gschuste@usr.com).

A. K. Katsaggelos is with Department of Electrical and Computer Engineering, McCormick School of Engineering and Applied Science, Northwestern University, Evanston, IL 60208-3118 USA (e-mail: aggk@ece.nwu.edu).

that is based on a rectangular grid superimposed on a planar curve. The curve is quantized using the grid intersection scheme [4] 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 b. There have been many extensions to this basic scheme, such as the generalized chain codes [5], where the coding efficiency has been improved by using links of different length and different angular resolution. In [8], a scheme is presented that utilizes patterns in a chain code string to increase the coding efficiency and in [9], differential chain codes are presented, which employ the statistical dependency between successive links. There has also been interest in the theoretical performance of chain codes. In [6], the performance of different quantization schemes is compared, whereas in [7], the rate distortion characteristics of certain chain codes are studied. In this paper, we are not concerned with the quantization of the continuous curve, since we assume that the object boundaries are given with pixel accuracy.

Most boundaries contain many straight lines or lines with a very small curvature, which result in runs of the same increment. Therefore, a run-length encoding scheme can be used to encode the eight-connect chain code even more efficiently. Clearly, the larger the number of straight lines a boundary contains, the more efficient a chain code/run-length scheme is. This is the idea behind some preprocessing algorithms [10] that are used to "straighten" the boundary, i.e., these algorithms lead to a new boundary that can be encoded more efficiently. Clearly, such a scheme is a lossy chain code/run-length encoding scheme since this preprocessing introduces an error in the boundary representation. The introduction of an error is usually permissible, as long as the visual distortion is considered insignificant. The main problem with these approaches is that there is at best an indirect control over the resulting rate distortion tradeoff.

In this paper, we present algorithms where the preprocessing step and the eight-connect chain code/run-length encoding are combined into an optimal lossy segmentation encoding scheme. The proposed approach offers complete control over the tradeoff between distortion and bit rate. Note that this is achieved in an optimal fashion, resulting in an efficient encoding scheme.

In Section II, we define the problem and introduce the required notation. In Section III, we focus on a first class of distortion measures that are based on the maximum operator, such as the maximum absolute distance. First, we consider the problem of finding the polygon that requires the smallest bit rate for a given distortion. We solve this problem by introducing a scheme that is based on a shortest path algorithm for a weighted directed acyclic graph. We then consider the dual problem, that of finding the polygon with the smallest distortion for a given bit rate. We derive an iterative scheme which employs the shortest path algorithm and prove that this scheme converges to the global optimum. In Section IV, we focus on a second class of distortion measures that are based on the summation operator, such as the mean squared distance. Again, we consider the problem of finding

the polygon that requires the smallest bit rate for a given distortion. We show how the Lagrange multiplier method can be employed to transform this constrained problem into a series of unconstrained problems that can be solved using the shortest path algorithm introduced earlier. Like every Lagrangian-based approach, the proposed scheme results only in solutions that belong to the convex hull of the operational rate distortion function. Hence, we also propose a second algorithm, based on a tree pruning scheme, that can find all optimal solutions. Since both of these algorithms are symmetric in the rate and the distortion, the dual problem, that of finding the polygon that results in the smallest distortion for a given bit rate, can also be solved by the same algorithms. In Section V, we extend the results for the jointly optimal encoding of multiple boundaries. In Section VI, we introduce a vertex encoding scheme that is based on an eight-connect chain code and run-length coding. In Section VII, we present results of the proposed algorithms and in Section VIII, we summarize the paper and present our conclusions.

II. PROBLEM FORMULATION

The main idea behind the proposed approach is to approximate a given boundary by a polygon, and to encode the polygons vertices instead of the original boundary. Since we assume that the original boundary is represented with pixel accuracy, it can be losslessly encoded by an eight-connect chain-code. We propose to approximate the boundary with a low-order polygon that can be encoded efficiently.

The following notation will be used. Let $B = \{b_0, \cdots, b_{N_B-1}\}$ denote the connected boundary, which is an ordered set, where b_j is the jth point of B and N_B is the total number of points in B. Note that in the case of a closed boundary, $b_0 = b_{N_B-1}$. Let $P = \{p_0, \cdots, p_{N_P-1}\}$ denote the polygon used to approximate B, which is also an ordered set, with p_k the kth vertex of P, N_P the total number of vertices in P and the kth edge starts at p_{k-1} and ends at p_k. Since P is an ordered set, the ordering rule and the set of vertices uniquely define the polygon. We will elaborate on the fact that the polygon is an ordered set later on.

We assume that the vertices of the polygon are encoded differentially, which is an efficient method for natural boundaries, since the location of the current vertex is strongly correlated with the location of the previous one. We denote the required bit rate for the differential encoding of vertex p_k given vertex p_{k-1} by $r(p_{k-1}, p_k)$. Hence, the bit rate $R(p_0, \cdots, p_{N_P-1})$ for the entire polygon is

$$R(p_0, \cdots, p_{N_P-1}) = \sum_{k=0}^{N_P-1} r(p_{k-1}, p_k) \qquad (1)$$

where $r(p_{-1}, p_0)$ is set equal to the number of bits needed to encode the absolute position of the first vertex. For a closed boundary, i.e., the first vertex is identical to the last one, the rate $r(p_{N_P-2}, p_{N_P-1})$ is set to zero since the last vertex does not need to be encoded. Note that this rate depends on the specific vertex encoding scheme. We present one such scheme that is a combination of an eight-chain code and a run-length scheme in Section VI.

In general, the polygon used to approximate the boundary could be permitted to place its vertices anywhere on the plane. In this paper, we restrict the vertices to belong to the original boundary ($p_k \in B$), so that we can employ a fast polygon selection algorithm. This restriction results in the following fact, which we employ to derive low complexity optimization algorithms. The kth polygon edge which connects two consecutive vertices, p_{k-1} and p_k, is an approximation to the partial boundary $\{b_j = p_{k-1}, b_{j+1}, \cdots, b_{j+l} = p_k\}$, which contains $l+1$ boundary points. Therefore, we can measure the quality of this approximation by an edge distortion measure which we denote by $d(p_{k-1}, p_k)$. The polygon distortion measure can then be expressed as the sum or the maximum of all edge distortion measures.

There are several different distortion measures that can be employed. One popular distortion measure for curve approximations is the maximum absolute distance, which has also been employed in [2], [6], [7], and [11]. Besides its perceptual relevance, this distortion measure has the advantage that it can be computed efficiently. Let $d(p_{k-1}, p_k, t)$ be the shortest distance between the line which goes through p_{k-1} and p_k and an arbitrary point t. This distance can be expressed as shown in (2), at the bottom of the page, where the subscripts x and y indicate the x and y coordinates of a particular point. Then, the maximum absolute distance between the partial boundary $\{b_j = p_{k-1}, b_{j+1}, \cdots, b_{j+l} = p_k\}$ and the edge (p_{k-1}, p_k) is given by

$$d(p_{k-1}, p_k) = \max_{t \in \{b_j = p_{k-1}, b_{j+1}, \cdots, b_{j+l} = p_k\}} d(p_{k-1}, p_k, t). \tag{3}$$

Another popular distortion measure is the mean squared distance (error), which has been used in [3] and [12] and is of the following form:

$$d(p_{k-1}, p_k) = \frac{1}{N_B} \sum_{t \in \{b_j = p_{k-1}, b_{j+1}, \cdots, b_{j+l} = p_k\}} \cdot d(p_{k-1}, p_k, t)^2. \tag{4}$$

So far, we have only discussed the edge distortion measures, i.e., the measures that judge the approximation of a certain partial boundary by a given polygon edge. In general, we are interested in a polygon distortion measure that can be used to determine the quality of approximation of an entire polygon. We will treat two different classes of polygon distortion measures. The first class is based on the maximum operator (or equivalently, on the minimum operator) and is of the following form:

$$D(p_0, \cdots, p_{N_P-1}) = \max_{k \in [0, \cdots, N_P-1]} d(p_{k-1}, p_k), \tag{5}$$

where $d(p_{-1}, p_0)$ is defined to be zero. We will refer to all distortion measures based on the above definition as class one distortion measures.

The second class of distortion measures is based on the summation operator is of the following form:

$$D(p_0, \cdots, p_{N_P-1}) = \sum_{k=0}^{N_P-1} d(p_{k-1}, p_k) \tag{6}$$

where again $d(p_{-1}, p_0)$ is set equal to zero. We will refer to all distortion measures based on the above definition as class two distortion measures.

The main motivation for considering these two classes of distortion measures stems from the popularity of the maximum absolute distance distortion measure, which is a class one measure, and the mean squared distance distortion measure, which is a class two measure. If we select the maximum absolute distance as the polygon distortion measure, then we have to use (3) for the edge distortion and (5) for the polygon distortion. On the other hand, if we select the mean squared distance as the distortion measure, then we have to use (4) for the edge distortion and (6) for the polygon distortion. Note that there are many other polygon distortion measures that fit into this framework, such as the absolute area or the total number of error pixels between the boundary and the polygon. As we mentioned above, one of the big advantages of restricting the vertices of the polygon to belong to the original boundary is the ability to express the polygon distortion as the sum or the maximum of the edges distortions. We will employ this fact later on to derive fast polygon selection algorithms. These fast algorithms are necessary since the number of possible polygons is extremely high. If we define the smallest possible polygon as a single point, then, given the degree of the polygon (N_P), there are $\binom{N_B}{N_P} = N_B!/(N_B - N_P)! \cdot N_P!$ different selections of N_P vertices from the original boundary. Since we have defined the polygon to be an ordered set, the set of vertices uniquely specifies a polygon. The degree of the polygon (N_P) is also a variable, therefore the total number of possible polygons is equal to

$$\sum_{k=1}^{N_B} \frac{N_B!}{(N_B - k)! \cdot k!}. \tag{7}$$

Clearly, an exhaustive search is not a feasible approach.

In the remainder of the paper we introduce fast and efficient algorithms for both classes of polygon distortion measures that solve the following constrained optimization problem

$$\min_{p_0, \cdots, p_{N_P-1}} D(p_0, \cdots, p_{N_P-1})$$

subject to

$$R(p_0, \cdots, p_{N_P-1}) \le R_{\max} \tag{8}$$

where $R_{\max}$ is the maximum bit rate permitted for the encoding of the boundary. We also present algorithms that solve the

$$d(p_{k-1}, p_k, t) = \frac{|(t_x - p_{k-1,x}) \cdot (p_{k,y} - p_{k-1,y}) - (t_y - p_{k-1,y}) \cdot (p_{k,x} - p_{k-1,x})|}{\sqrt{(p_{k,x} - p_{k-1,x})^2 + (p_{k,y} - p_{k-1,y})^2}} \tag{2}$$

dual problem

$$\min_{p_0,\cdots,p_{N_P-1}} R(p_0,\cdots,p_{N_P-1}),$$

subject to

$$D(p_0,\cdots,p_{N_P-1}) \leq D_{\max} \qquad (9)$$

where $D_{\max}$ is the maximum distortion permitted. Note that there is an inherent tradeoff between the rate and the distortion in the sense that a small distortion requires a high rate, whereas a small rate results in a high distortion. As we will see, the solution approaches for problems (8) and (9) are related in the sense that the algorithms are symmetric with respect to the rate and the distortion or the algorithm developed to solve problem (9) is used iteratively to solve problem (8).

III. DISTORTION MEASURES BASED ON THE MAXIMUM OPERATOR

In this section, we introduce two algorithms to solve the problems stated in (8) and (9) for class one distortion measures, such as the maximum absolute distance.

A. The Minimum Rate Case

First, we consider the minimum rate case stated in (9). The goal of the proposed algorithm is to find the polygon whose vertices can be encoded with the smallest number of bits. This selection is constrained by the fact that the selected polygon must result in a distortion smaller or equal to the maximum distortion.

The key observation for deriving an efficient search is the fact that given a certain vertex of a polygon (p_{k-1}) and the rate which is required to code the polygon up to and including this vertex $(R_{k-1}(p_{k-1}))$, the selection of the next vertex p_k is independent of the selection of the previous vertices $p_0,\cdots,p_{k-2}$. This is true since the rate can be expressed recursively as a function of the segment rates $r(p_{k-1},p_k)$ and the segment distortions $d(p_{k-1},p_k)$. That is,

$$R_k(p_k) = R_{k-1}(p_{k-1}) + w(p_{k-1},p_k) \qquad (10)$$

where

$$w(p_{k-1},p_k) = \begin{cases} \infty: & d(p_{k-1},p_k) > D_{\max} \\ r(p_{k-1},p_k): & d(p_{k-1},p_k) \leq D_{\max} \end{cases}. \qquad (11)$$

This recursion needs to be initialized by setting $R_{-1}(p_{-1})$ equal to zero. Note that by the definition of $w(p_{k-1},p_k)$, the rate for a polygon that does not satisfy the maximum distortion constraint is infinite. Clearly, $R_{N_P-1}(p_{N_P-1}) = R(p_0,\cdots,p_{N_P-1})$, the rate for the entire polygon.

As indicated above, we need to start the search for an optimal polygon at a given vertex. If the boundary is not closed, the first boundary point b_0 has to be selected as the first vertex p_0. For a closed boundary, the selection of the first vertex is less obvious. Ideally, the algorithm should find all the optimal vertices, including the first one. Unfortunately, the above recursion requires a starting vertex. Hence, since

we want to use this recursion to derive a fast algorithm, we need to fix the first vertex, even for a closed boundary. Therefore, the found solution is optimal, given the constraint of the predetermined first vertex. Clearly, we can drop this constraint by finding all optimal approximations using each boundary point as a starting vertex and then selecting the overall best solution. This exhaustive search with respect to the initial vertex is computationally quite expensive. We, therefore, propose to select the point with the highest curvature as the first vertex, since it is the most likely point to be included in any polygonal approximation. This heuristic almost always results in the best possible selection of the initial vertex and if not, the performance difference is negligible.

For future convenience, we relabel the boundary so that the first vertex of the polygon p_0 coincides with the first point of the boundary b_0. Besides fixing the first vertex of the polygon, we also require that the last vertex p_{N_P-1} is equal to the last point of the boundary b_{N_B-1}. This leads to a closed polygonal approximation for a closed boundary. For a boundary which is not closed, this condition, together with the starting condition, makes sure that the approximation starts and ends at the same points as the boundary.

Using (10), the problem stated in (9) can be formulated as a shortest path problem in a weighted directed graph $G = (V, E)$, where V is the sets of graph vertices and E is the set of edges (see Fig. 1). Let $V = B$ since every boundary point can be a polygon vertex. Note that there are two kind of vertices, polygon vertices and graph vertices. In the proposed formulation, each graph vertex represents a possible polygon vertex and, henceforth, we will drop the distinction between these two entities. The edges between the vertices represent the line segments of the polygon. 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. Since every combination of different boundary points can represent a line segment of a valid polygon, the edge set E is defined as follows: $E = \{(b_i, b_j) \in B^2 : \forall i \neq j\}$ (see Fig. 1). A path of order K from vertex u to a vertex u' is an ordered set $\{v_0,\cdots,v_K\}$ such that $u = v_0$, $u' = v_K$ and $(v_{k-1},v_k) \in E$ for $k = 1,\cdots,K$. The order of the path is the number of edges in the path. The length of a path is defined as follows:

$$\sum_{k=1}^{K} w(v_{k-1},v_k) \qquad (12)$$

where $w(u,v) : E \to \mathcal{R}$ is a weight function defined in accordance with (11) as follows:

$$w(u,v) = \begin{cases} \infty: & d(u,v) > D_{\max} \\ r(u,v): & d(u,v) \leq D_{\max} \end{cases}. \qquad (13)$$

Again, note that the above definition of the weight function leads to a length of infinity for every path (polygon), which includes a line segment resulting in an approximation error larger than $D_{\max}$. Therefore, a shortest path algorithm will not select these paths. Every path that starts at vertex p_0 and ends at vertex p_{N_B-1} and does not result in a path length of infinity, results in a path length equal to the rate of the polygon it represents. Therefore, the shortest of all those paths
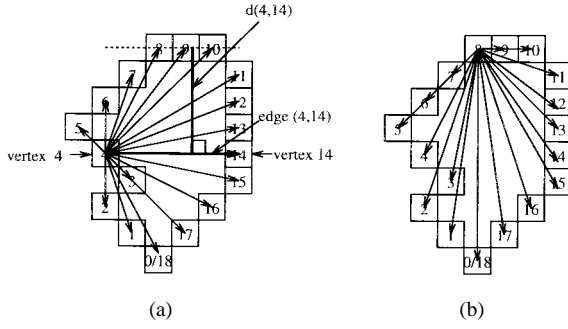
Fig. 1. Interpretation of the boundary and the polygon approximation as a fully connected weighted directed graph. Note that the set of all edges E equals $\{(b_i, b_j) \in B^2: i \neq j\}$. Two representative subsets are displayed: (a) $\{(b_4, b_j) \in B^2: \forall j \neq 4\}$ and (b) $\{(b_8, b_j) \in B^2: \forall j \neq 8\}$.
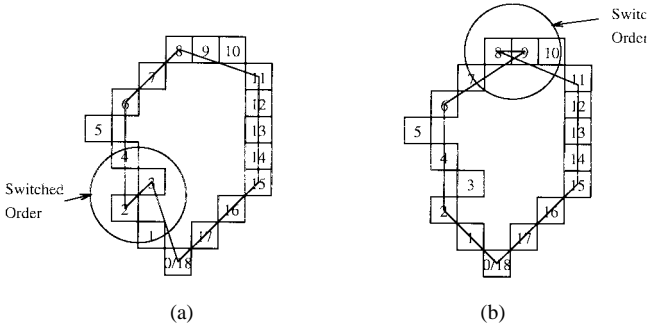

Fig. 2. Examples of polygons with rapid changes in direction.

corresponds to the polygon with the smallest bit rate, which is the solution to the problem in (9).

The classical algorithm for solving such a single-source shortest-path problem, where all the weights are nonnegative, is Dijkstra's algorithm [13] with time complexity $O(|V|^2 + |E|)$. This is a significant reduction compared to the time complexity of exhaustive search. Recall that we defined the polygon as an ordered set for reasons that will now become apparent. We can further simplify the algorithm by observing that it is very unlikely for the optimal path to select a boundary point b_j as a vertex when the last selected vertex was b_i, where $i > j$. In general, we cannot guarantee that the optimal path will not do this, since the selection process depends on the vertex encoding scheme, which we have not specified yet. On the other hand, a polygon where successive vertices are not assigned to boundary points in increasing order can exhibit rapid direction changes, even when the original boundary is quite smooth (see Fig. 2). Therefore, we add the restriction that not every possible combination of (b_i, b_j) represents a valid edge but only the ones for which $i < j$. Hence, the edge set E is redefined in the following way: $E = \{(b_i, b_j) \in B^2: i < j\}$ (see Fig. 3). This restriction results in the fact that a given vertex set uniquely specifies the polygon. We used this fact before to derive the number of possible polygons in an exhaustive search approach.

By defining the edge set E in the above fashion, we achieve two goals simultaneously. First, the selected polygon approximation has to follow the original boundary without rapid direction changes, and second and more important, the
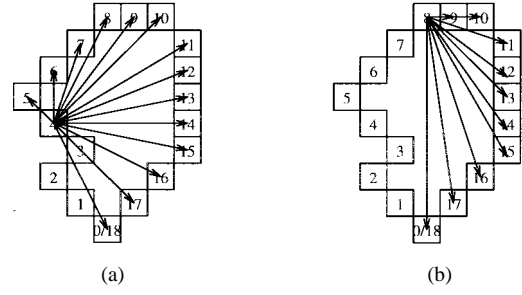

Fig. 3. Interpretation of the boundary and the polygon approximation as a weighted directed graph. Note that the set of all edges E equals $\{(b_i, b_j) \in B^2: i < j\}$. Two representative subsets are displayed: (a) $\{(b_4, b_j) \in B^2: \forall j > 4\}$ and (b) $\{(b_8, b_j) \in B^2: \forall j > 8\}$.

resulting graph is a weighted *directed acyclic graph* (DAG). For a DAG, there exists an algorithm for finding a single-source shortest-path, which is even faster than Dijkstra's algorithm. Following the notation in [13], we call this the *DAG-shortest-path algorithm*. The time complexity for the DAG-shortest-path algorithm is $\Theta(|V| + |E|)$, which means that the asymptotic lower bound $\Omega(|V| + |E|)$ is equal to the asymptotic upper bound $O(|V| + |E|)$.

Let $R^*(b_i)$ represent the minimum rate to reach the boundary point b_i from the source vertex $p_0 = b_0$ via a polygon approximation. Clearly, $R^*(b_{N_B-1})$ is the solution to problem (9). Let $q(b_i)$ be a back pointer which is used to remember the optimal path. Then the proposed algorithm works as follows.

1) $R^*(p_0) = r(p_{-1}, p_0);$
2) for $i = 1, \ldots, N_B - 1;$
3) $\{$
4) $R^*(b_i) = \infty;$
5) $\}$
6) for $i = 0, \ldots, N_B - 2;$
7) $\{$
8)   for $j = i + 1, \ldots, N_B - 1;$
9) $\{$
10)   calculate edge distortion $d(b_i, b_j);$
11)   look up edge rate $r(b_i, b_j);$
12)   assign $w(b_i, b_j)$ based on definition (13);
13)   if $(R^*(b_i) + w(b_i, b_j) < R^*(b_j));$
14) $\{$
15) $R^*(b_j) = R^*(b_i) + w(b_i, b_j);$
16) $q(b_j) = b_i;$
17) $\}$
18) $\}$
19) $\}.$

The optimal path $\{p_0^*, \cdots, p_{N_P-1}^*\}$ can be found by backtracking the pointers $q(b_i)$ in the following recursive fashion (by definition $p_{N_P-1}^* = b_{N_B-1}$ and $p_0^* = b_0$):

$$p_{k-1}^* = q(p_k^*), \qquad k = N_P - 1, \cdots, 2. \qquad (14)$$

The formal proof of the correctness of the DAG-shortest-path algorithm, on which the above scheme is based, can be found in [13]. We will reason more intuitively how this approach works. In line 1, the rate for encoding the starting point of the

boundary is assigned to the minimum rate of the first polygon vertex. In lines 2–5, the minimum rate for reaching any of the boundary points is set to infinity. The "for loop" in line 6 selects the boundary points in sequence as possible vertex points from which a polygon edge starts and the "for loop" in line 8 selects possible vertex points where the polygon edge ends. Hence, these two loops select each edge in the edge set E exactly once. Therefore, the lines 9–18 are processed for every edge. The lines 10–12 are used to calculate the weight of the edge, $w(b_i, b_j)$. The most important part of this algorithm is the comparison in line 13. Here, we test if the new bit rate, $R^*(b_i) + w(b_i, b_j)$, to reach boundary point b_j, given that the last vertex was b_i, is smaller than the smallest bit rate used so far to reach b_j, $R^*(b_j)$. If this bit rate is indeed smaller, then it is assigned as the new smallest bit rate to reach boundary point b_j, $R^*(b_j) = R^*(b_i) + w(b_i, b_j)$. We also assign the back pointer of b_j, $q(b_j)$ to point to b_i since this is the previous vertex used to achieve $R^*(b_j)$. This algorithm leads to the optimal solution because, as stated earlier, when the rate $(R^*(b_i))$ of a vertex (b_i) is given, then the selection of the future vertices $(b_j, i < j < N_B)$ is independent of the selection of the past vertices $(b_k, 0 \leq k < i)$.

The analysis of the above algorithm shows that there are two nested loops, which results in a time complexity of $\Theta(N_B^2)$. We use the number of edge distortion evaluations as measure for the time complexity, since this is the most time-consuming operation. In the case where the edge distortion is the maximum absolute distance, then another loop is required because of the maximum operator in (3). Therefore, the time complexity of the maximum absolute distance algorithm is $\Theta(N_B^3)$ with respect to the distance evaluations in (2).

B. The Minimum Distortion Case

We now consider the minimum distortion case which is stated in (8). The goal of the proposed algorithm is to find the polygon with the smallest distortion for a given bit budget for encoding its vertices. Sometimes this is called a *rate constrained approach*. Recall that for class one distortion measures the polygon distortion is defined as the maximum of the edge distortions [see (5)]. Hence, in this section, we propose an efficient algorithm that finds the polygonal approximation with the smallest maximum distortion for a given bit rate.

We propose an iterative solution to this problem that is based on the fact that we can solve the dual problem stated in (9) optimally. Consider $D_{\max}$ in (9) to be a variable. We derived in Section III-A an algorithm that finds the polygonal approximation which results in the minimum rate for any $D_{\max}$. We denote this optimal rate by $R^*(D_{\max})$. We prove below that the rate $R^*(D_{\max})$ is a nonincreasing function of $D_{\max}$, which means that $D_{\max}^1 < D_{\max}^2$ implies $R^*(D_{\max}^1) \geq R^*(D_{\max}^2)$.

Proof: (By contradiction): Let polygon P_1 and rate $R^*(D_{\max}^1)$ be the solutions to the minimum rate optimization problem 1. Let polygon P_2 and rate $R^*(D_{\max}^2)$ be the solutions to the minimum rate optimization problem 2. Assume that $D_{\max}^1 < D_{\max}^2$ and $R^*(D_{\max}^1) < R^*(D_{\max}^2)$.
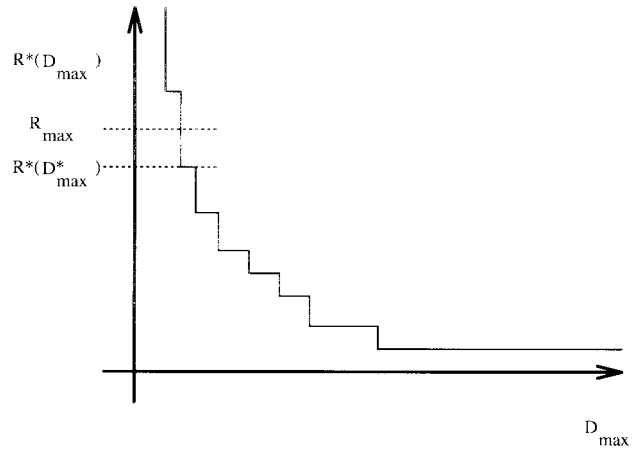

Fig. 4. $R^*(D_{\max})$ function, which is a nonincreasing function exhibiting a staircase characteristic. The selected $R_{\max}$ falls onto a discontinuity and, therefore, the optimal solution is of the form $R^*(D_{\max}^*) < R_{\max}$, instead of $R^*(D_{\max}^*) = R_{\max}$.

Then, P_1 is an admissible polygon for the optimization problem 2, since $D_{\max}^1 < D_{\max}^2$. Since by assumption $R^*(D_{\max}^1) < R^*(D_{\max}^2)$, P_1 is a better solution than P_2, which is a contradiction, since we showed that the selection algorithm employed to find P_2 is optimal. Hence, $D_{\max}^1 < D_{\max}^2$ implies $R^*(D_{\max}^1) \geq R^*(D_{\max}^2)$. ∎

Having shown that $R^*(D_{\max})$ is a nonincreasing function, we can use bisection [14] to find the optimal $D_{\max}^*$ such that $R^*(D_{\max}^*) = R_{\max}$. Since this is a discrete optimization problem, the function $R^*(D_{\max})$ is not continuous and exhibits a staircase characteristic (see Fig. 4). This implies that there might not exist a $D_{\max}^*$ such that $R^*(D_{\max}^*) = R_{\max}$. In that case, the proposed algorithm will still find the optimal solution, which is of the form $R^*(D_{\max}^*) < R_{\max}$, but only after an infinite number of iterations. Therefore if we have not found a $D_{\max}$ such that $R^*(D_{\max}) = R_{\max}$ after a given maximum number of iterations, we terminate the algorithm.

IV. DISTORTION MEASURES BASED ON THE SUMMATION OPERATOR

In this section, we introduce two algorithms to solve the problems stated in (8) and (9) for class two distortion measures, such as the mean squared distance. Both presented algorithms are symmetric in the rate and the distortion and, hence, the same technique can be employed for the minimum distortion case (8) and the minimum rate case (9). We will, therefore, only solve the minimum distortion case, and the minimum rate case can be solved be applying the following relabeling to the function names: $D(p_0, \cdots, p_{N_P-1}) \leftarrow R(p_0, \cdots, p_{N_P-1})$ and $R(p_0, \cdots, p_{N_P-1}) \leftarrow D(p_0, \cdots, p_{N_P-1})$.

The first algorithm we propose is based on the Lagrange multiplier method. Like every Lagrangian-based approach the resulting solutions belong to the convex hull of the operational rate distortion function. For cases where the Lagrangian bound is not tight enough, we propose a tree-pruning-based scheme, which can find all optimal solutions.

A. Lagrange Multiplier Approach

In this section, we derive a solution to problem (8) that is based on the Lagrange multiplier method [15]–[17] and the shortest path algorithm presented in Section III-A. Lagrangian relaxation [18] is a well-known tool in operations research. It is mainly used to relax constraints that make the solution of an integer problem difficult. The relaxed integer program can then be solved more easily, which leads to an efficient method for certain problems. The Lagrange multiplier method is closely related to Lagrangian relaxation, and it is extremely useful for solving constrained resource allocation problems. In this application, we will use the Lagrange multiplier method to relax the constraint, so that the relaxed problem can be solved using the shortest path algorithm proposed in Section III-A.

We first define the Lagrangian cost function

$$J_\lambda(p_0, \cdots, p_{N_P-1}) = D(p_0, \cdots, p_{N_P-1}) + \lambda \cdot R(p_0, \cdots, p_{N_P-1}) \quad (15)$$

where λ is called the Lagrange multiplier. It has been shown in [15] and [16] that if there is a λ^* such that

$$\{p_0^*, \cdots, p_{N_P-1}^*\} = \arg \min_{p_0, \cdots, p_{N_P-1}} J_{\lambda^*}(p_0, \cdots, p_{N_P-1}) \quad (16)$$

and which leads to $R(p_0, \cdots, p_{N_P-1}) = R_{\max}$, then $\{p_0^*, \cdots, p_{N_P-1}^*\}$ is also an optimal solution to (8). It is well known that when λ sweeps from zero to infinity, the solution to problem (16) traces out the convex hull of the operational rate distortion function, which is a nonincreasing function. Hence, bisection [14] or the fast convex search we presented in [19] can be used to find λ^*. Therefore, if we can find the optimal solution to the unconstrained problem (16), then we can find the optimal λ^* and the convex hull approximation to the constrained problem of (8).

The key observation for deriving an efficient search for the polygon that minimizes the unconstrained problem (16) is based on the fact that given a certain vertex of a polygon (p_{k-1}) and the Lagrangian cost function that results by coding the polygon up to and including this vertex $(J_{k-1}(p_{k-1}) = D_{k-1}(p_{k-1}) + \lambda \cdot R_{k-1}(p_{k-1}))$, the selection of the next vertex p_k is independent of the selection of the previous vertices $p_0, \cdots, p_{k-2}$. This is true since the rate and the distortion can be expressed recursively as functions of the segment rates $r(p_{k-1}, p_k)$ and segment distortions $d(p_{k-1}, p_k)$

$$R_k(p_k) = R_{k-1}(p_{k-1}) + r(p_{k-1}, p_k) \quad (17)$$

and

$$D_k(p_k) = D_{k-1}(p_{k-1}) + d(p_{k-1}, p_k). \quad (18)$$

These recursions need to be initialized by setting $R_{-1}(p_{-1})$ and $D_{-1}(p_{-1})$ equal to zero. Since the rate and the distortion can be calculated with a first-order recursion, the Lagrangian cost function can also be calculated recursively as follows:

$$J_k(p_k) = J_{k-1}(p_{k-1}) + \{d(p_{k-1}, p_k) + \lambda \cdot r(p_{k-1}, p_k)\}. \quad (19)$$

Clearly, $R_{N_P-1}(p_{N_P-1}) = R(p_0, \cdots, p_{N_P-1})$, the rate for the entire polygon, $D_{N_P-1}(p_{N_P-1}) = D(p_0, \cdots, p_{N_P-1})$ the distortion for the entire polygon and $J_{N_P-1}(p_{N_P-1}) = J(p_0, \cdots, p_{N_P-1})$, the Lagrangian cost function for the entire polygon.

In Section III-A, we have shown that an optimization problem that has the above-described structure can be solved optimally by a DAG-shortest-path algorithm. To be able to employ the previously proposed shortest path algorithm, we have to redefine the weight function $w(u, v): E \rightarrow \mathcal{R}$ as

$$w(u, v) = d(u, v) + \lambda \cdot r(u, v). \quad (20)$$

Since the shortest path algorithm results in the polygon that minimizes the following sum:

$$\sum_{k=0}^{N_P-1} w(p_{k-1}, p_k) \quad (21)$$

this polygon is the optimal solution to the relaxed problem of (16).

Clearly, the time complexity of the Lagrangian approach for a fixed λ is the same as for the shortest path algorithm. The shortest path algorithm is invoked several times by the bisection algorithm to find the optimal λ^* and, hence, the time complexity is a function of the number of required iterations. As pointed out before, the Lagrangian approach finds optimal solutions that belong to the convex hull of the operational rate distortion curve. Clearly, there are other optimal solutions that are above the convex hull. In the next section, we present a tree-pruning algorithm that finds all optimal solutions.

B. Pruning Approach

As before, for this algorithm we make use of the fact that when the current vertex is selected and we know the rate and the distortion used to encode the polygon up to and including this vertex, the previous vertices do not influence the selection of the future vertices. For a given boundary point (b_i) under consideration to be chosen as a polygon vertex, every previous boundary point $(b_k, 0 \le k < i)$ could have been the last vertex used for the polygon approximation. Therefore, the rate and distortion for every previous boundary point are calculated under the assumption that the previous boundary point was used as the previous vertex. These calculations lead to a set of nodes, each representing the hypothesis that the current boundary point is a vertex but with different boundary points as previous vertices.

We introduce a pruning procedure to reduce the number of nodes for each boundary point. If there are two nodes j and i such that $D(j) \ge D(i)$ and $R(j) \ge R(i)$, where $D(n)$ is the distortion and $R(n)$ the rate up to and including node n, then it is clear that node j cannot belong to the optimal solution. This is because node i has a lower distortion and a lower rate than node j, but both represent the same boundary point as the last selected vertex. Hence, node j is pruned from the decision tree. Since the pruned nodes need not be considered in the future of the optimization process, the more nodes pruned, the faster the algorithm becomes. A straightforward approach to pruning has a quadratic time complexity in the number of nodes to be pruned N. Since N depends on previous pruning results, the time complexity of the entire approach depends on
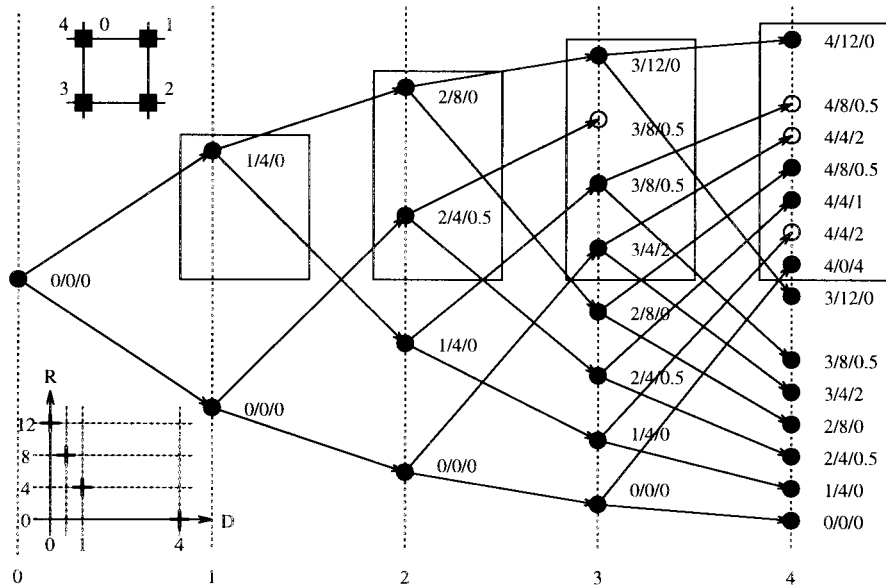
Fig. 5.   Pruned decision tree for the encoding of a boundary. The nodes are labeled as follows: "index/rate/distortion."

the boundary, the distortion measure, and the vertex encoding scheme. As with most integer programming algorithms, one can construct an example where the pruning scheme fails completely, which results in an exponential time complexity. Note that this exponential time complexity is still better than the time complexity of the exhaustive search, since we use the fact about the independence of the future with respect to the past. In general, though, the pruning is extremely efficient in cutting down the complexity of the algorithm and, in fact, this scheme and the previously discussed Lagrangian approach take about the same amount of time for the experimental results we will present in Section VII. If the complete set of optimal rate distortion points is not of interest, but only the problem of (8) needs to be solved, additional pruning can be achieved by removing all nodes that contain a rate higher than $R_{\max}$. This leads effectively to all optimal rate distortion points below and including the line $R(D) = R_{\max}$.

Each of the remaining nodes represents a polygon which has the current boundary point as its last vertex, but with different rate-distortion characteristics. In other words, the remaining nodes represent the set of all optimal solutions for the encoding of the boundary up to and including the current boundary point. These nodes make up the admissible nodes for this boundary point, when this boundary point is considered as a previous vertex in the future of the optimization process.

Fig. 5 shows a simple example of the algorithm. In the left upper corner is the boundary that must be encoded. The adjacent pixels are labeled 0, 1, 2, 3, and 4, and they simply form a square of side length 1. Note that point 4 is the same as point 0 and, therefore, it does not need to be transmitted, but still a distortion occurs between the last vertex of the polygon and point 4 and it is, therefore, included in the closed boundary. Fig. 5 shows the complete decision tree. This tree reflects the fact that given the boundary point used as the previous vertex, and the rate and the distortion for encoding the polygon up to and including that vertex, the selection

of future vertices is independent of the selection made for previous vertices. In Fig. 5, the boundary point index, the rate and the distortion are indicated in the following fashion: "index/rate/distortion." In this example, the sum of the squared distance between the boundary and the polygon is used as the distortion measure and the vertex encoding scheme proposed in Section VI is employed (4 b is required for each transition in this example).

There are two possible transitions from a given node. The upward transition, which indicates that this node is used as a previous vertex, and the downward transition which indicates that this node is not used as a previous vertex. The downward transitions carry a weight of zero (rate = 0, distortion = 0), but the upward transitions result in the addition of r(previous vertex, current vertex) to the rate and d(previous vertex, current vertex) to the distortion. The epochs, which correspond to the boundary points, are indicated at the bottom of the tree. The boxes indicate the new nodes per boundary point and, therefore, the pruning procedure is only applied to those nodes.

Consider the box at epoch 3. There are two nodes (both with description 3/8/0.5) that require the same rate and lead to the same distortion to reach boundary point 3. Therefore, one of the two can be pruned (indicated by an empty circle), since both will lead to the same collection of future paths. By pruning a node, the collection of future paths gets reduced. Clearly, the more nodes that can be pruned, the faster the algorithm is.

At the last epoch, which corresponds to boundary point 4, three nodes can be pruned and only four final nodes remain, which represent the four optimal solutions to the boundary approximation. These four optimal solutions are also displayed as an operational rate distortion function in the lower left corner of Fig. 5. The path (0,1,2,3,4) that leads to 12 b and no distortion is the highest quality approximation which is basically the chain code of the original boundary. The path

(0,1,2,2,4) approximates the box by a triangle, which requires 8 b and leads to a distortion of 0.5. The path (0,0,2,2,4) approximates the box by a diagonal line, which requires 4 b and leads to a distortion of 1. Finally, the path (0,0,0,0,4) does not require any bits, since it approximates the box by its starting point, but it leads to a maximum distortion of 4.

It is interesting to note that this pruning scheme can be easily modified to work with class one distortion measures, since the additivity of the class two distortion measures has not been used in the derivation of this scheme. The only fact employed is the independence of the future from the past, which is also present for class one distortion measures. In general though, the schemes presented for class one distortion measures are faster than the pruning approach.

V. MULTIPLE BOUNDARY ENCODING

In this section, we extend the results of the previous sections for the encoding of multiple boundaries. Assume that M different boundaries have to be encoded and we will adopt the convention that a subscript indicates which boundary is addressed, i.e., B_3 is the third boundary, P_4 is the polygon used to approximate the fourth boundary, etc. Then, the minimum rate optimization problem can be stated as follows:

$$\min_{P_0, \cdots, P_{M-1}} R(P_0, \cdots, P_{M-1})$$

subject to

$$D(P_0, \cdots, P_{M-1}) \le D_{\max} \qquad (22)$$

whereas the minimum distortion problem is of the following form:

$$\min_{P_0, \cdots, P_{M-1}} D(P_0, \cdots, P_{M-1})$$

subject to

$$R(P_0, \cdots, P_{M-1}) \le R_{\max}. \qquad (23)$$

The total rate $R(P_0, \cdots, P_{M-1})$ in the above formulation is defined as

$$R(P_0, \cdots, P_{M-1}) = \sum_{i=0}^{M-1} R_i(p_{i,0}, \cdots, p_{i,N_{P_i}-1}). \quad (24)$$

The total distortion measure $D(P_0, \cdots, P_{M-1})$ is defined for class one distortion measures by

$$D(P_0, \cdots, P_{M-1}) = \max_{i \in [0, \cdots, M-1]} D_i(p_{i,0}, \cdots, p_{i,N_{P_i}-1}) \qquad (25)$$

and for class two distortion measures by

$$D(P_0, \cdots, P_{M-1}) = \sum_{i=0}^{M-1} D_i(p_{i,0}, \cdots, p_{i,N_{P_i}-1}). \quad (26)$$

As we have seen in the previous sections, the two classes of distortion measures require different algorithms. This is also true for the encoding of multiple boundaries.

A. Distortion Measures Based on the Maximum Operator

1) The Minimum Rate Case: Since the total rate $R(P_0, \cdots, P_{M-1})$ is the sum of the individual rates $R_i(p_{i,0}, \cdots, p_{i,N_{P_i}-1})$, and the encoding of the different boundaries is accomplished independently, the minimum total rate is equal to the sum of the minimum individual rates, where the search for the minimum individual rates is also constrained by the maximum distortion $D_{\max}$. Therefore, the following optimization problem is identical to the one in (22):

$$\min_{p_{i,0}, \cdots, p_{i,N_{P_i}-1}} R_i(p_{i,0}, \cdots, p_{i,N_{P_i}-1})$$

subject to

$$D_i(p_{i,0}, \cdots, p_{i,N_{P_i}-1}) \le D_{\max}, \quad \text{for } i = 0, \cdots, M-1 \qquad (27)$$

which shows that the optimal solution to problem (22) can be found by solving the optimization problems for the different boundaries independently using the algorithm developed in Section III-A.

2) The Minimum Distortion Case: We now consider the minimum distortion case of (23). As in Section III-B, we use the fact that we can solve the minimum rate problem optimally, in order to solve the minimum distortion problem by an iterative scheme. By defining $R^*(D_{\max})$ of Section III-B as the minimum total rate needed to encode the M given boundaries with a maximum error of $D_{\max}$, the derivation in Section III-B still applies. Hence, the resulting algorithm is a bisection search and at each iteration the optimization problem of (22) is solved optimally using the above proposed scheme.

B. Distortion Measures Based on the Summation Operator

For class two distortion measures, we have introduced two different algorithms, the Lagrange multiplier approach and the tree-pruning scheme. As we pointed out in Section IV, these algorithms are symmetric in the rate and the distortion and, therefore, we will only discuss the minimum distortion case of (23).

1) Lagrange Multiplier Approach: We define the total Lagrangian cost function $J_\lambda(P_0, \cdots, P_{M-1})$ as follows:

$$J_\lambda(P_0, \cdots, P_{M-1}) = \sum_{i=0}^{M-1} \{ D_i(p_{i,0}, \cdots, p_{i,N_{P_i}-1}) + \lambda \cdot R_i(p_{i,0}, \cdots, p_{i,N_{P_i}-1}) \}. \quad (28)$$

According to Section IV-A, if we can find the global minimum of the total Lagrangian cost function with respect to the approximation polygons

$$\min_{P_0, \cdots, P_{M-1}} J_\lambda(P_0, \cdots, P_{M-1})$$

$$= \min_{P_0, \cdots, P_{M-1}} \left\{ \sum_{i=0}^{M-1} D_i(p_{i,0}, \cdots, p_{i,N_{P_i}-1}) + \lambda \cdot R_i(p_{i,0}, \cdots, p_{i,N_{P_i}-1}) \right\}. \qquad (29)$$

Then we can use an iterative search to solve the constrained problem of (23).

Since the different boundaries are independently encoded, the minimum of the total Lagrangian cost function can be found by minimizing each of the individual boundaries separately. In other words, this problem reduces to the ones studied in [16]. Hence, the following minimization is equivalent to the one in (29):

$$
\min_{P_0, \cdots, P_{M-1}} J_\lambda(P_0, \cdots, P_{M-1})
$$
$$
= \sum_{i=0}^{M-1} \min_{p_{i,0}, \cdots, p_{i,N_{P_i}-1}} \Big\{ D_i(p_{i,0}, \cdots, p_{i,N_{P_i}-1}) +
$$
$$
\lambda \cdot R_i(p_{i,0}, \cdots, p_{i,N_{P_i}-1}) \Big\}. \qquad (30)
$$

Therefore, the multiple boundary encoding problem can be solved with the Lagrange multiplier method using (30), which states that the global minimum of the total Lagrangian cost function is the sum of the global minima of the Lagrangian cost functions for each boundary. Since we derived in Section IV-A an algorithm to find the minimum of a boundary Lagrangian cost function, we can find the minimum of (30) and, hence, we can encode multiple objects using the Lagrangian relaxation scheme.

2) Pruning Approach: The tree-pruning approach proposed in Section IV-B results in the operational rate distortion function (ORDF) for a given boundary. Again, since the boundaries are encoded independently, we first run the pruning algorithm for each of the M boundaries. This results in M different ORDF's. It is interesting to notice that the optimal bit allocation among independent quantizers (characterized by their ORDF's) is commonly solved by the Lagrange multiplier method [16]. Again, the Lagrange multiplier method will only find solutions that belong to the convex hull, but we are interested in all optimal solutions, which is the main reason for introducing the pruning scheme.

We need the total ORDF to be able to solve the minimum distortion problem formulated in (23) optimally. Hence, the problem is to create the total ORDF, using the M ORDF's of the boundaries. In [20] and [21], a dynamic programming-based approach is presented for the case where the ORDF's are defined on the set of positive integers. We introduce a different approach, which does not require that the ORDF's are defined for all the positive integers, but works for ORDF's defined on any finite subset of the real line. The total ORDF can be found by applying a slightly modified version of the pruning scheme to the M different ORDF's of the boundaries. We explain this scheme with the help of the example in Fig. 6. In the rounded boxes on top are the points of the ORDF's of three different boundaries. The notation used in this figure is of the following form: "rate/distortion." The goal is to merge these three ORDF's to find the total ORDF, which is displayed on the right. This is achieved by creating the total ORDF iteratively. First, we generate the combined ORDF of the first and the second boundary. This is achieved by creating all possible rate distortion points, which are inside box number one. Then we apply the same pruning rule we established before, i.e., if there are two nodes j and i such that
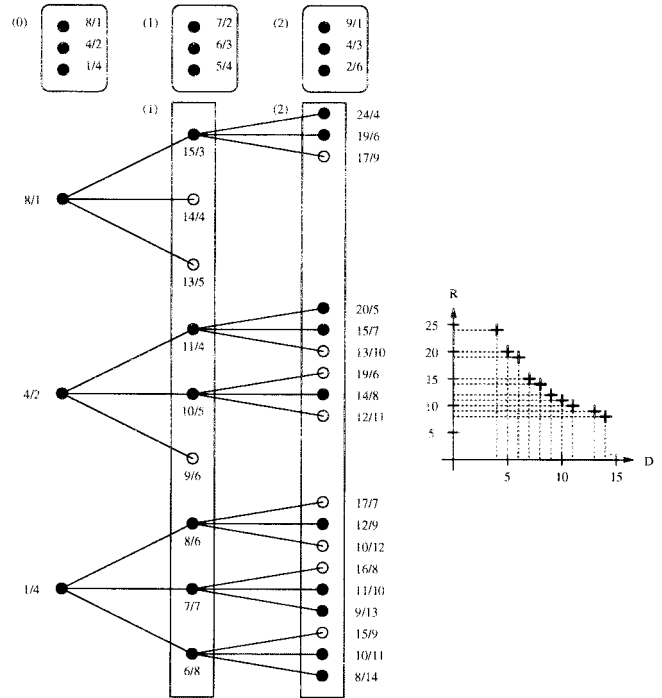

Fig. 6. Pruned decision tree for the optimal encoding of three boundaries. The nodes are labeled as follows: "rate/distortion."

$D(j) \geq D(i)$ and $R(j) \geq R(i)$, then node j cannot belong to the optimal solution. The pruned nodes are indicated with an empty circle and the black nodes represent the new combined ORDF. We then iteratively apply this merging and pruning of two ORDF's to create a combined ORDF, until there is only one ORDF left, which represents the total ORDF and it can directly be used to find the optimal solution to the multiple boundary encoding scheme.

VI. VERTEX ENCODING SCHEME

So far, we have not assumed any specific scheme for encoding the vertices of the polygon. In this section, we present a vertex encoding scheme, which can be considered a combination of an eight-connect chain code and a run-length encoding scheme.

A. Basic Scheme

The chain code and the run-length encoding can be combined by representing the increment between two vertices by an angle α and a run β, which form the symbol (α, β). Therefore, for a run of one, the eight closest neighbors of a given point P are

$$
\begin{array}{ccc}
(3, 1) & (2, 1) & (1, 1) \\
(4, 1) & P & (0, 1) \\
(-3, 1) & (-2, 1) & (-1, 1).
\end{array} \qquad (31)
$$

As an example, $(3, 4)$ represents a straight line of four increments in the $3 \cdot \pi/4$ direction. Each of the possible symbols (α, β) gets a probability assigned and the resulting stream of increments $I = [(\alpha_1, \beta_1), \cdots, (\alpha_{N-1}, \beta_{N-1})]$ can be encoded by an arithmetic or a Huffman code. We use the

following code word assignment. For a given symbol (α, β), the first three bits indicate one of the eight possible values for α followed by $(\beta - 1)$ zeros and a final "1" to encode the number of runs. Clearly, the number of bits used for this uniquely decodable code is equal to $(3 + \beta)$ and the longer the run, the more efficient this code is. Note that this code implies that the lines between the vertices are restricted to intersect the horizontal axis in an angle, which is an integer multiple of $\pi/4$.

B. Generalized Scheme

A generalization of this code is based on the observation that this scheme is optimal in the case when the probability mass function of (α, β) is separable and α is uniformly distributed over all eight α's, whereas β is geometrically distributed $(P(\beta = j) = (1 - \gamma)/\gamma \cdot \gamma^j, \ j \geq 1)$ with parameter $\gamma = 0.5$. The assumptions that the distribution is separable, α is uniformly distributed, and β is geometrically distributed are reasonable, but there might be better choices for γ than 0.5. When an arithmetic coder is used, the resulting bit rate is the entropy based on the probability model the encoder employs (we neglect the renormalization bits used in practical implementations). Therefore, a probability model that leads to a smaller entropy than the above one can be used, even though this leads to fractional bit assignments per symbol. For example, only six out of the eight α's need to be considered since the next α cannot be equal to the current one (if so, this would be coded by an additional run), nor can it be equal to the direct opposite one (if so, one less run would have been coded). Hence, there are only six possible α's, and instead of using 3 b to encode them, only $\log_2 (1/6) \approx 2.6$ b are needed.

The question, therefore, is: Which γ leads to the smallest bit rate for the encoding of a particular polygon? It can be shown that the maximum likelihood estimate γ_{ML} also leads to the minimum entropy and, hence, to the smallest bit rate. Since we assume that the runs β_i for the encoding of vertices p_i are independent of each other and have the same geometric probability mass function, the likelihood function can be written as follows:

$$P(\beta_1, \cdots, \beta_{N_P-1}) = \prod_{i=1}^{N_P-1} \frac{1-\gamma}{\gamma} \cdot \gamma^{\beta_i} \qquad (32)$$

which leads to the following maximum likelihood estimate of γ:

$$\gamma_{ML} = 1 - \frac{N_P - 1}{\sum_{i=1}^{N_P-1} \beta_i}. \qquad (33)$$

So far, we have considered the case where γ_{ML} has been given (in other words, the code word assignment has been given) and then the optimal polygon approximation is found. The question arises of how to jointly select γ_{ML} and the polygon approximation optimally. In fact γ_{ML} has to be quantized since it needs to be sent for every boundary and, in the current scheme, an 8-b uniform quantizer with a range from zero to one is used for that purpose. Therefore, 256 different γ's exist,

and one solution is to run the optimal polygon approximation 256 times and pick the quantized γ, which leads to the smallest rate.

We propose a much faster, but suboptimal iterative procedure to estimate γ_{ML}. This procedure can be applied to all schemes presented, which do not employ an iteration whose convergence is based on the global optimality of the solution. In other words, this scheme can be applied to the minimum rate case for distortion measures of class one and for the pruning scheme introduced for class two distortion measures in Section IV-B.

The iteration works as follows. First an initial quantized $\gamma_{ML,i}^Q$ is used, where i indicates the iteration number and Q the fact that this is a quantized value, and the $(i+1)$th optimal polygon approximation is found. Then, this approximation is used to estimate $\gamma_{ML,i+1}$ based on the distribution of the runs and the quantized $\gamma_{ML,i+1}^Q$ is derived from $\gamma_{ML,i+1}$. These three steps are repeated until the minimum rate for the polygon of iteration $(i+1)$, $R_{i+1}^*(\gamma_{ML,i}^Q)$ does not decrease any further, which usually happens after two to three iterations. Since the minimum rate of the polygonal approximation is bounded from below by zero, we can prove that this iteration converges to a local minimum by showing that $R_{i+1}^*(\gamma_{ML,i}^Q)$ is a nonincreasing function of $\gamma_{ML,i}^Q$.

Proof: Clearly, $R_{i+1}^*(\gamma_{ML,i}^Q) \geq R_{i+1}^*(\gamma_{ML,i+1})$ since $\gamma_{ML,i+1}$ is estimated using the runs of the optimal polygon of iteration $(i+1)$. Since the likelihood function used to find $\gamma_{ML,i+1}$ is concave, $\gamma_{ML,i+1}^Q$ can be found by evaluating the likelihood function for the two reconstruction levels, which are the closest to $\gamma_{ML,i+1}$ and setting $\gamma_{ML,i+1}^Q$ equal to the one which results in the higher score. Note that this is a special case where the optimal solution to a discrete problem (finding $\gamma_{ML,i+1}^Q$) can be inferred from the solution of a continuous problem (finding $\gamma_{ML,i+1}$). Therefore, $R_{i+1}^*(\gamma_{ML,i}^Q) \geq R_{i+1}^*(\gamma_{ML,i+1}^Q) \geq R_{i+2}^*(\gamma_{ML,i+1}^Q)$ which proves the convergence to a local minimum. ∎

Using an arithmetic coder and this iterative scheme, the efficiency of the minimum rate approach for class one distortion measures can be improved by about 15%.

VII. EXPERIMENTAL RESULTS

In this section, we present experimental results of the proposed algorithms using object boundaries from the Miss America sequence. For the presented experiments, we use the vertex encoding scheme with $\gamma = 0.5$ and the Huffman code proposed in Section VI.

We first present results for class one distortion measures, where the employed distortion measure is the minimum absolute distance. In Fig. 7, we compare the original segmentation, which is displayed in the left figure, versus the optimal segmentation for a maximum distortion $D_{\max}$ of one pixel, which is displayed in the right figure. The two objects in the original segmentation require 468 b if encoded by an eight-connect chain code, whereas the optimal segmentation can be encoded with only 235 b. By introducing a permissible maximum error of one pixel, we are able to reduce the total bit rate by about 50%. As expected, some of the details have been
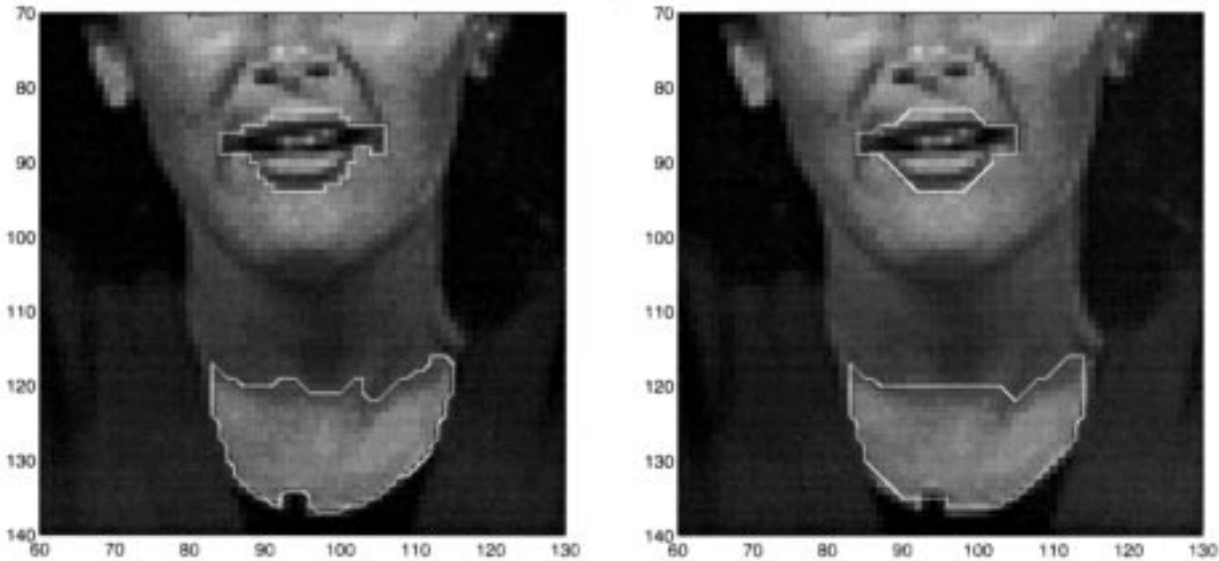
Fig. 7.   Left: original segmentation, which requires 468 b using the eight-connect chain code. Right: optimal segmentation with $D_{\max} = 1$ pixel, which requires a rate of 235 b and results in a distortion of one pixel.
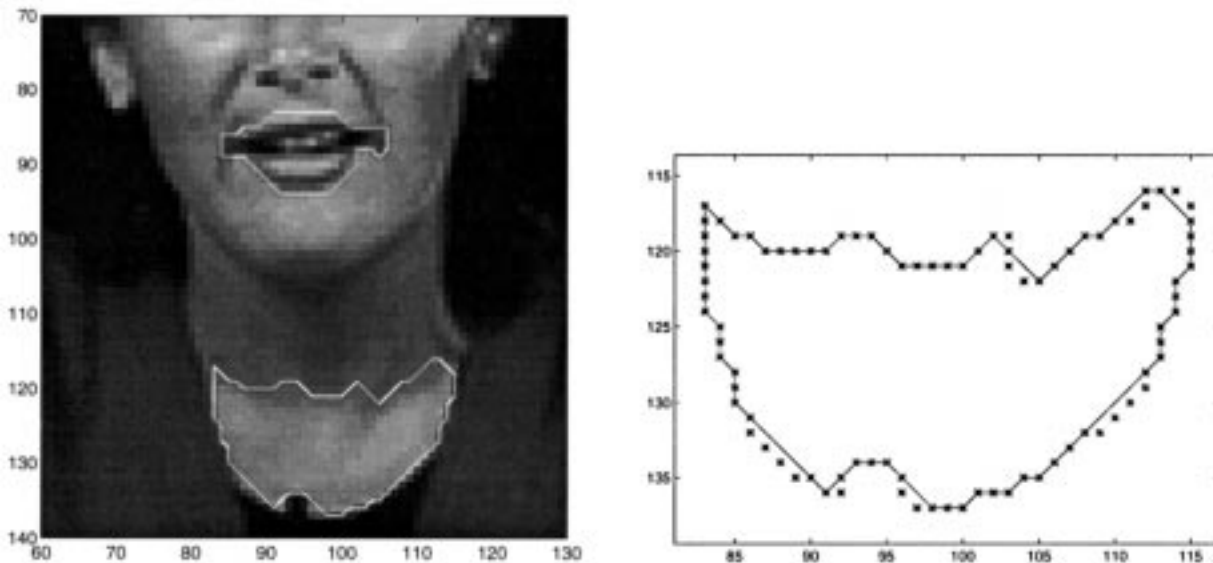

Fig. 8.   Left: optimal segmentation with $R_{\max} = 280$ b, which results in a distortion of 0.71 pixels and a bit rate of 274 b. Right: closeup of the lower boundary; the stars indicate the original boundary and the line represents the polygonal approximation. The upper left corner has been selected as the first vertex.

lost, i.e., the boundary has been "straightened." This smoothing of the boundary might be desired, since most segmentation algorithms result in noisy boundaries. In Fig. 8, we show the resulting segmentation for the minimum distortion case for multiple boundaries. The maximum rate $R_{\max}$ has been set to 280 b, and the optimal solution, which uses 274 b for a $D_{\max} = 0.71$ pixels, is displayed in the left figure. The right figure is a close-up of the lower boundary in the left figure, and the stars indicate the original boundary with the polygonal approximation drawn on top of it.

In Fig. 9, we present results for distortion measures of class two, where the employed distortion measure is the mean squared distance. To highlight the difference between the Lagrange multiplier scheme and the pruning scheme, we set $R_{\max}$ from (8) to 200 b. In Fig. 10, a close-up of the boundary is shown, and the operational rate distortion curve is displayed. Note the possible Lagrangian solutions, which are indicated by circles around the vertices of the convex hull. The Lagrangian solution that satisfies $R \leq 200$ b results in $R = 169$ b and $D = 0.1$, (for a $\lambda^* = 0.002$), whereas the pruning approach results in $R = 200$ b and $D = 0.05$. Both solutions are shown in Fig. 9, and it is clear that the pruning approach results in a better approximation of the original boundary.

VIII. SUMMARY AND CONCLUSIONS

We presented fast and efficient methods for the lossy encoding of object boundaries which are given as eight-connect chain codes. The boundary is approximated by a polygon, and
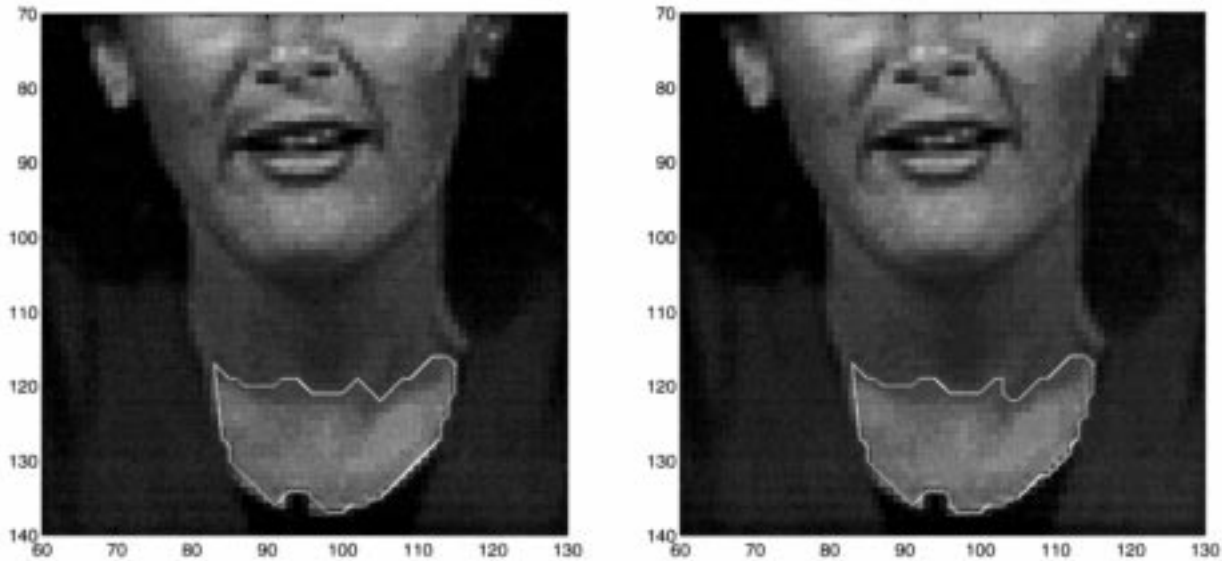
Fig. 9. Comparison between the Lagrangian relaxation approach and the pruning approach for $R_{\max} = 200$ b, for distortion measure the mean squared distance. Left: Lagrange multiplier approach, $R = 169$ b, $D = 0.1$. Right: Pruning approach, $R = 200$ b, $D = 0.05$.
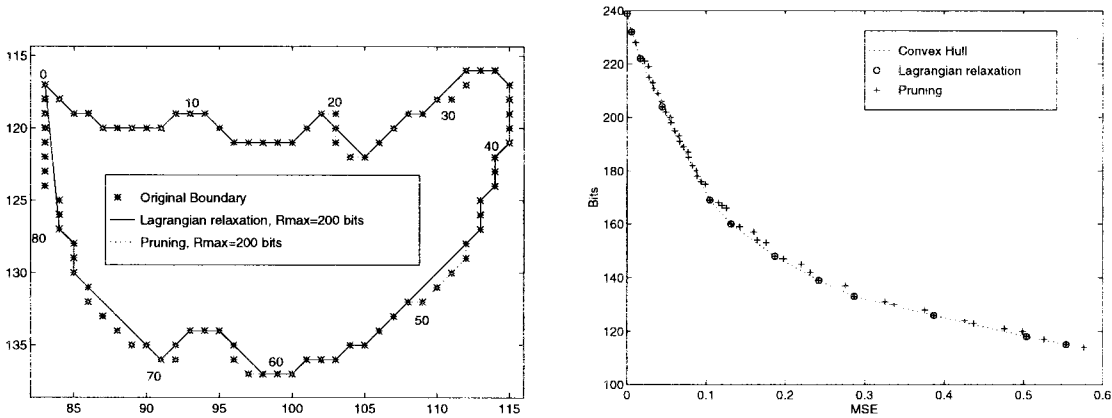

Fig. 10. Comparison between the Lagrangian relaxation approach and the pruning approach. Left: close-up of the boundary and the two different approximations. Right: the operational rate distortion function and its convex hull.

we considered the problem of finding the polygon, which leads to the smallest distortion for a given number of bits. The dual problem of finding the polygon that leads to the smallest bit rate for a given distortion was also addressed. We considered two different classes of distortion measures, where the first class is based on the maximum operator and the second class is based on the summation operator. For the first class, we derived a scheme that is based on a shortest-path algorithm for a weighted directed acyclic graph. For the second class, we proposed a Lagrange multiplier-based approach, which employs the shortest path algorithm iteratively. Lagrangian schemes can only find solutions that belong to the convex hull of the operational rate distortion function; therefore, we also proposed a tree-pruning algorithm that can find all optimal solutions. We extended all proposed schemes to the jointly optimal encoding of multiple boundaries. We finally introduced a vertex encoding scheme, which is a combination of an eight-connect chain code and a run-length scheme.

Experimental results of the proposed schemes were presented using objects from the Miss America sequence.

In conclusion, we compare the different approaches and how they might be applied for the encoding of object boundaries. The Lagrangian-based approaches (class two distortion measures) are iterative schemes, and so is the minimum distortion approach for class one distortion measures. Even though these schemes converge to the optimal solution, several iterations might be required. The pruning-based approaches (class two distortion measures) are one-pass approaches, and so is the minimum rate approach for class one distortion measures. Unfortunately, the efficiency of the pruning schemes cannot be guaranteed. This is in contrast to the minimum rate approach for class one distortion measures. This one-pass method has a time complexity of $\Theta(N_B^2)$ and it is the fastest of all proposed methods. By selecting the edge distortion to be the maximum distance, this algorithm efficiently finds the smallest rate polygonal approximation to a given boundary, which

stays within a maximum error of $D_{\max}$. Because of its speed and its perceptual relevance, this is our preferred approach for encoding object boundaries in a variable bit rate coding framework.

In [23], the vertices of the polygon that approximates the given boundary are allowed to be outside the set of boundary points but inside a predefined distortion band. In [24], second-order B-splines are used for the encoding of the boundary, whose control points lie within a predefined distortion band. Finally, in [25], the various optimal boundary encoding approaches we have developed are reviewed, along with the methods developed toward the solution of the same problem, by the MPEG-4 standardization effort.

REFERENCES

[1] H. Musmann, M. Hötter, and J. Ostermann, "Object-oriented analysis-synthesis coding of moving images," *Signal Process.: Image Commun.,* vol. 1, pp. 117–138, Oct. 1989.

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

[3] A. K. Jain, *Fundamentals of Digital Image Processing.* Englewood Cliffs, NJ: Prentice-Hall, 1989.

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

[5] J. Saghri and H. Freeman, "Analysis of the precision of generalized chain codes for the representation of planar curves," *IEEE Trans. Pattern Anal. Machine Intell.,* vol. PAMI-3, pp. 533–539, Sept. 1981.

[6] J. Koplowitz, "On the performance of chain codes for quantization of line drawings," *IEEE Trans. Pattern Anal. Machine Intell.,* vol. PAMI-3, pp. 180–185, Mar. 1981.

[7] D. Neuhoff and K. Castor, "A rate and distortion analysis of chain codes for line drawings," *IEEE Trans. Inform. Theory,* vol. IT-31, pp. 53–68, Jan. 1985.

[8] T. Kaneko and M. Okudaira, "Encoding of arbitary curves based on the chain code representation," *IEEE Trans. Commun.,* vol. COMM-33, pp. 697–707, July 1985.

[9] R. Prasad, J. W. Vieveen, J. H. Bons, and J. C. Arnbak, "Relative vector probabilities in differential chain coded line-drawings," in *Proc. IEEE Pacific Rim Conf. Communication, Computers, and Signal Processing,* Victoria, B.C., Canada, June 1989, pp. 138–142.

[10] T. Özçelik, "A very low bit rate video codec," Ph.D. dissertation, Dept. Electr. Eng. Comput. Sci., Northwestern Univ., Evanston, IL, Dec. 1994.

[11] G. M. Schuster and A. K. Katsaggelos, "An optimal lossy segmentation encoding scheme," in *SPIE Proc. Conf. Visual Communications and Image Processing,* Mar. 1996, pp. 1050–1061.

[12] ——, "An optimal segmentation encoding scheme in the rate-distortion sense," in *Proc. Int. Symp. Circuits and Systems,* Atlanta, GA, May 1996, vol. 2, pp. 640–643.

[13] T. Cormen, C. Leiserson, and R. Rivest, *Introduction to Algorithms.* New York: McGraw-Hill, 1991.

[14] C. F. Gerald and P. O. Wheatley, *Applied Numerical Analysis,* 4th ed. Reading, MA: Addison, 1990.

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

[16] 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.

[17] K. Ramchandran, A. Ortega, and M. Vetterli, "Bit allocation for dependent quantization with applications to multiresolution and MPEG video coders," *IEEE Trans. Image Processing,* vol. 3, pp. 533–545, Sept. 1994.

[18] M. L. Fisher, "The Lagrangian relaxation method for solving integer programming problems," *Manage. Sci.,* vol. 27, pp. 1–18, Jan. 1981.

[19] 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 Communications and Image Processing,* Mar. 1996, pp. 784–795.

[20] A. V. Trushkin, "Bit number distribution upon quantization of a multivariate random variable," *Problems Inform. Transmission,* vol. 16, pp. 76–79, 1980, trans. from Russian.

[21] ——, "Optimal bit allocation algorithm for quantizing a random vector," *Problems Inform. Transmission,* vol. 17, pp. 156–161, 1981, trans. from Russian.

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

[23] ——, "An efficient boundary encoding scheme which is optimal in the rate distortion sense," in *Proc. 1996 Int. Conf. Image Processing,* Lausanne, Switzerland, Sept. 1996, pp II-77–80.

[24] F. W. Meier, G. M. Schuster, and A. K. Katsaggelos, An efficient boundary encoding scheme using B-spline curves which is optimal in the rate-distortion sense," in *Proc. 2nd Erlangen Symp., Advances in Digital Image Communication*, Erlangen, Germany, April 25, 1997, pp. 75–84.

[25] A. K. Katsaggelos *et al.*, "MPEG-4 and rate-distortion based shape coding techniques," *Proc. IEEE Special Issue Multimedia Signal Processing,* to be published.

Guido M. Schuster received the Ing HTL degree in elektronik, mess- und regeltechnik in 1990 from the Neu Technikum Buchs (NTB), Buchs, St. Gallen, Switzerland, and the M.S. and Ph.D. degrees, both in electrical engineering, from Northwestern University, Evanston, IL, in 1992 and 1996, respectively.

In 1996, he joined the Network Systems Division of U.S. Robotics (now the Carrier Systems Business Unit of 3COM), Mount Prospect, IL, where he co-founded the Advanced Technologies Research Center. His current research interests are operational rate-distortion theory and networked multimedia. He is co-author of *Rate-Distortion Based Video Compression* (Boston, MA: Kluwer). He has filed and holds several patents in fields ranging from adaptive control over video compression to internet telephony.

Dr. Schuster was awarded the gold medal for academic excellence and was the winner of the first annual Landis & Gyr Fellowship Competition, both at NTB.

Aggelos K. Katsaggelos (SM'92–F'98) received the Dipl. degree in electrical and mechanical engineering from the Aristotelian University of Thessaloniki, Thessaloniki, Greece, in 1979, and the M.S. and Ph.D. degrees, both in electrical engineering, from the Georgia Institute of Technology, Atlanta, in 1981 and 1985, respectively.

In 1985, he joined the Department of Electrical Engineering and Computer Science at Northwestern University, Evanston, IL, where he is currently a Professor holding the Ameritech Chair of Information Technology. During the 1986–1987 academic year, he was an Assistant Professor at Polytechnic University, Department of Electrical Engineering and Computer Science, Brooklyn, NY. His current research interests include image recovery, processing of moving images (motion estimation, enhancement, very low bit rate compression), computational vision, and multimedia signal processing. He is a member of the Associate Staff, Department of Medicine, at Evanston Hospital.

Dr. Katsaggelos is on the steering committees of the IEEE TRANSACTIONS ON MEDICAL IMAGING and the IEEE TRANSACTIONS ON IMAGE PROCESSING. He is a member of the the IEEE Technical Committees on Visual Signal Processing and Communications, Image and Multidimensional Signal Processing, and Multimedia Signal Processing; the Technical Chamber of Commerce of Greece; and Sigma Xi. He has served as an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING (1990–1992), an Area Editor for *Graphical Models and Image Processing* (1992–1995), and is currently the Editor-in-Chief of the *IEEE Signal Processing Magazine*. He is the Editor of *Digital Image Restoration* (New York: Springer-Verlag, 1991), and co-author of *Rate-Distortion Based Video Compression* (Boston, MA: Kluwer, 1997). He has served as the General Chairman of the 1994 Visual Communications and Image Processing Conference, Chicago, IL, and will serve as the Technical Program Co-Chair of the 1998 IEEE International Conference on Image Processing, Chicago. He is an Ameritech Fellow and a member of SPIE.