An optimal lossy segmentation encoding scheme

*Guido M. Schuster and †Aggelos K. Katsaggelos

Northwestern University
Department of Electrical Engineering and Computer Science
McCormick School of Engineering and Applied Science
Evanston, IL 60208-3118
Email: *gschuster@nwu.edu, †aggk@eecs.nwu.edu

ABSTRACT

In this paper, we present a fast and optimal method for the lossy encoding of object boundaries which are given as 8-connect chain codes. We approximate the boundary by a polygon and consider the problem of finding the polygon which can be encoded with the smallest number of bits for a given maximum distortion. To this end, we derive a fast and optimal scheme which is based on a shortest path algorithm for a weighted directed acyclic graph. We further investigate the dual problem of finding the polygonal approximation which leads to the smallest maximum distortion for a given bit rate. We present an iterative scheme which employs the above mentioned shortest path algorithm and prove that it converges to the optimal solution. We then extend the proposed algorithm to the encoding of multiple object boundaries and introduce a vertex encoding scheme which is a combination of an 8-connect chain code and a run-length code. We present results of the proposed algorithm using objects from the "Miss America" sequence.

Keywords: Segmentation Encoding, Chain Codes, Object Oriented Video Coding.

1 INTRODUCTION

A major problem in object oriented video coding[1] is the efficient encoding of object boundaries. There are two common approaches for encoding the segmentation information. A lossy approach, which is based on a spline approximation of the boundary,[2] and a lossless approach which is based on chain codes.[3] The proposed boundary encoding scheme is a lossy scheme which 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 approach, whereas the encoding of the vertices is achieved with a chain code-like scheme.

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. Chain coding was originally proposed by Freeman[3] and has attracted considerable attention over the last thirty years.[4–8] 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[3] 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 8 grid neighbors, and hence an increment can be represented by 3 bits. There have been many extensions to this basic scheme such as the generalized chain codes,[4] where the coding efficiency has been improved by using links of different length and different angular resolution. In Ref.[7] a scheme is presented which utilizes patterns in a chain code string to increase the coding efficiency and in Ref.[8] 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 Ref.[5] the performance of different quantization schemes is compared, whereas in Ref.[6] the rate distortion characteristics of certain chain codes are studied.

In this paper, the quantization of the continuous curve is not an issue, since we assume that the segmentation is given with pixel accuracy. According to the 8-connect chain code encoding scheme, a pixel on a boundary is selected as the starting point of the chain code. Every future pixel can then be described as being one of the 8 closest neighbors of the current pixel. Hence 3 bits per boundary pixel are required to encode this incremental information between two pixels. The string of these increments is considered the 8-connect chain code of the boundary. 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 8-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[9] which are used to "straighten" the boundary, i.e., these algorithms lead to a new boundary which can be encoded more efficiently. Clearly this preprocessing introduces an error in the boundary representation, but as long as the error is small enough, the visual distortion is considered insignificant. Such approaches, however have at best an indirect control of the amount of the distortion introduced in the representation of the boundary.

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

In section 2 we define the problem and introduce the required notation. In section 3 we derive an algorithm to find the polygon with the smallest bit rate for a given maximum distortion. In section 4 we address the dual problem of finding the polygon with the smallest distortion for a given maximum bit rate. In section 5 we extend the results for the jointly optimal encoding of multiple boundaries. In section 6 we introduce a vertex encoding scheme which is based on an 8-connect chain code and run-length coding. In section 7 we present results of the proposed algorithms and in section 8 we summarize the paper.

2 PROBLEM FORMULATION

The main idea behind the proposed approach is to approximate the 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 8-connect chain-code. We propose to approximate the boundary with a low order polygon which can be encoded efficiently and also reduces some of the noise along the object boundary which is a result of the segmentation algorithm.

The following notation will be used. Let $B = \{b_0, \ldots, b_{N_B-1}\}$ denote the connected boundary which is an ordered set, where b_j is the j-th 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 $P = \{p_0, \ldots, p_{N_P-1}\}$ denote the polygon used to approximate B which is an ordered set, where p_k is the k-th vertex of P and N_P is the total number of vertices in P.

We assume that the vertices of the polygon are encoded differentially which is for example the case when a single ring chain code or a generalized chain code[4] is used. In other words, only the difference between two consecutive vertices is encoded, where we denote the required bit rate as $r(p_{k-1}, p_k)$. This rate depends on the specific vertex encoding scheme and we will present one such scheme in section 6. Note that the presented theory can be used with other vertex encoding schemes such as the ones mentioned above. In general the polygon which is used to approximate the boundary should be permitted to place its vertices anywhere in the plane. In this paper, we will restrict the location of the vertices so that we can employ a fast polygon selection algorithm. We restrict the vertices to belong to the original boundary ($p_k \in B$). This restriction results in the following fact, which we employ to derive a low complexity optimization algorithm.

The straight line connecting two consecutive vertices p_{k-1} and p_k is an approximation to the partial boundary $\{b_j = p_{k-1}, b_{j+1}, \ldots, b_{j+l} = p_k\}$ which contains $l+1$ boundary points. Therefore, we can measure the quality of this approximation by a segment distortion measure which we denote by $d(p_{k-1}, p_k)$. A popular distortion measure for line approximations is the maximum absolute distance between a boundary point and the approximation,[5,6] which we will also employ in this paper. Besides its perceptual relevance, this distortion measure has the advantage that it can be computed efficiently by the following formula,

$$d(p_{k-1}, p_k) = \max_{t \in \{b_j = p_{k-1}, b_{j+1}, \ldots, b_{j+l} = p_k\}} \frac{|(t_x - p_{k-1,x}) * (p_{k,y} - p_{k-1,y}) - (t_y - p_{k-1,y}) * (p_{k,x} - p_{k-1,x})|}{\sqrt{(p_{k,x} - p_{k-1,x})^2 + (p_{k,y} - p_{k-1,y})^2}}, \quad (1)$$

where the subscripts x and y indicate the x and y coordinate of a particular point. Note that because of the restriction that the vertices need to belong to the boundary, we are able to formulate the distortion for every line segment of the polygon independently.

3 MINIMUM RATE CASE

In this section we derive an efficient algorithm for selecting the optimal polygonal approximation for a given boundary in the sense that the encoding of this polygon requires the fewest number of bits for a given maximum distortion. This objective can be stated as follows,

$$\min_{[p_0, \ldots, p_{N_P-1}] \in B^{N_P}} R(p_0, \ldots, p_{N_P-1}), \quad \text{subject to:} \quad D(p_0, \ldots, p_{N_P-1}) \leq D_{max}, \quad (2)$$

where $R(p_0, \ldots, p_{N_P-1})$ is the rate required to encode all the vertices of the polygon, $D(p_0, \ldots, p_{N_P-1})$ is the resulting distortion and D_{max} is the maximum permissible distortion. The rate $R(p_0, \ldots, p_{N_P-1})$ can be expressed as the sum of the segment rates $r(p_{k-1}, p_k)$,

$$R(p_0, \ldots, p_{N_P-1}) = \sum_{k=0}^{N_P-1} r(p_{k-1}, p_k), \quad (3)$$

where $r(p_{-1}, p_0)$ is set equal to the number of bits needed to encode the absolute position of the first vertex. In the case of 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. The distortion $D(p_0, \ldots, p_{N_P-1})$ can be expressed as the maximum of the segment distortions $d(p_{k-1}, p_k)$,

$$D(p_0, \ldots, p_{N_P-1}) = \max_{k \in [0, \ldots, N_P-1]} d(p_{k-1}, p_k), \quad (4)$$

where $d(p_{-1}, p_0)$ is defined to be zero. 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.

If we define the smallest possible polygon as a single point, then, given the degree of the polygon (N_P), there are $\frac{N_B!}{(N_B - N_P)! * N_P!}$ different polygons which only use vertices from the original boundary. Since the degree of the

polygon (N_P) is also a variable, the total number of possible polygons is,

$$\sum_{k=1}^{N_B} \frac{N_B!}{(N_B-k)!*k!}. \tag{5}$$

Clearly an exhaustive search is not a feasible approach.

The key observation for deriving an efficient search is based on the fact that given a certain vertex of a polygon (p_{k-1}), the rate which is required to code the polygon up to and including this vertex ($R_{k-1}(p_{k-1})$), and the distortion of this polygonal approximation up to and including this vertex ($D_{k-1}(p_{k-1})$), the selection of the next vertex p_k is independent of the selection of the previous vertices $p_0, \ldots, p_{k-2}$. 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 the segment distortions $d(p_{k-1}, p_k)$,

$$R_k(p_k) = R_{k-1}(p_{k-1}) + r(p_{k-1}, p_k), \tag{6}$$

and

$$D_k(p_k) = \max\{D_{k-1}(p_{k-1}), d(p_{k-1}, p_k)\}. \tag{7}$$

This recursion needs to be initialized by setting $R_{-1}(p_{-1})$ and $D_{-1}(p_{-1})$ equal to zero. Clearly $R_{N_P-1}(p_{N_P-1}) = R(p_0, \ldots, p_{N_P-1})$, the rate for the entire polygon and $D_{N_P-1}(p_{N_P-1}) = D(p_0, \ldots, p_{N_P-1})$ the distortion 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, we can select any boundary point and a good choice is the point with the highest curvature since it is the most likely point to be included in any polygonal approximation. 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. Note that our experiments have shown that the efficiency of the solution is not sensitive to the choice of the initial starting point. 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 and 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 Eqs. (6) and (7), the problem stated in Eq. (2) 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, $E = \{(b_i, b_j) \in B^2 : \forall i \neq j\}$. A path of order K from vertex u to a vertex u' is an ordered set $\{v_0, \ldots, v_K\}$ such that $u = v_0$, $u' = v_K$ and $(v_{k-1}, v_k) \in E$ for $k = 1, \ldots, K$. The order of the path is the number of edges in the path. The length of a path is defined as follows,

$$\sum_{k=1}^{K} w(v_{k-1}, v_k), \tag{8}$$

where $w(u, v) : E \rightarrow \mathcal{R}$ is a weight function defined as follows,

$$w(u, v) = \begin{cases} \infty & : \quad d(u, v) > D_{max} \\ r(u, v) & : \quad d(u, v) \leq D_{max} \end{cases}. \tag{9}$$

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 which 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 corresponds to the polygon with the smallest bit rate which is the solution to the problem in Eq. (2).

The classical algorithm for solving such a single-source shortest-path problem, where all the weights are non-negative, is Dijkstra's algorithm[10] and its time complexity is $O(|V|^2 + |E|)$. This is a significant reduction compared to the time complexity of the exhaustive search. 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 increasing boundary points can exhibits 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).

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 more importantly, the 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 Ref.,[10] 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 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 (2). Let $q(b_i)$ be a back pointer which is used to remember the optimal path. Then the proposed algorithm works as follows:

$R^*(p_0) = r(p_{-1}, p_0)$;
for $i = 1, \ldots, N_B - 1$;
{
    $R^*(b_i) = \infty$;
}
for $i = 0, \ldots, N_B - 2$;
{
    for $j = i + 1, \ldots, N_B - 1$;
    {
        calculate $d(b_i, b_j)$ using Eq. (1);
        look up $r(b_i, b_j)$;
        assign $w(b_i, b_j)$ based on definition (9);
        if $(R^*(b_i) + w(b_i, b_j) < R^*(b_j))$;
        {
            $R^*(b_j) = R^*(b_i) + w(b_i, b_j)$;
            $q(b_j) = b_i$;
        }
    }
}

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

$$p_{k-1}^* = q(p_k), \qquad k = N_P - 1, \ldots, 2. \tag{10}$$

The proof of the correctness of the DAG-shortest-path algorithm, on which the above scheme is based, can be found in Ref..[10] The analysis of the above algorithm shows that inside the two nested loops, a distortion

is calculated, which requires another loop because of the maximum operator in Eq. (1). Therefore the time complexity of this algorithm is $\Theta(N_B^3)$.

4 MINIMUM DISTORTION CASE

In the previous section we presented the solution to the minimum rate problem stated in Eq. (2). In this section we study the dual problem, that of finding the polygonal approximation with the minimum distortion for a given maximum bit rate R_{max}. This can be expressed as follows,

$$\min_{[p_0,\ldots,p_{N_P-1}]\in B^{N_P}} D(p_0,\ldots,p_{N_P-1}), \quad \text{subject to:} \quad R(p_0,\ldots,p_{N_P-1}) \leq R_{max}. \tag{11}$$

We propose an iterative solution to this problem which is based on the fact that we can solve the dual problem stated in Eq. (2) optimally. Consider D_{max} in Eq. (2) to be a variable. We derived in section 3 an algorithm which finds the polygonal approximation which results in the minimum rate for any D_{max}. We denote this optimal rate by $R^*(D_{max})$. We claim that the rate $R^*(D_{max})$ is a non-increasing function of D_{max}, which means that $D^1_{max} < D^2_{max}$ implies $R^*(D^1_{max}) \geq R^*(D^2_{max})$.

We will prove this claim by contradiction. Assume that $D^1_{max} < D^2_{max}$ and $R^*(D^1_{max}) < R^*(D^2_{max})$. In this case, the polygon which resulted in D^1_{max} and $R^*(D^1_{max})$ is also an admissible polygon for the second optimization, since $D^1_{max} < D^2_{max}$. Since by assumption $R^*(D^1_{max}) < R^*(D^2_{max})$, this polygon is a better solution than the one selected during the second optimization process, which is a contradiction since the above algorithm is optimal. Hence $D^1_{max} < D^2_{max}$ implies $R^*(D^1_{max}) \geq R^*(D^2_{max})$.

Having proven that $R^*(D_{max})$ is a non-increasing function, we can use bisection[11] 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 following 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 stop the algorithm.

5 MULTIPLE BOUNDARY ENCODING

In this section we extend the results of sections 3 and 4 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 and P_4 is the polygon used to approximate the fourth boundary etc.. Then the minimum rate optimization problem can be stated as follows,

$$\min_{P_0,\ldots,P_{M-1}} R(P_0,\ldots,P_{M-1}), \quad \text{subject to:} \quad D(P_0,\ldots,P_{M-1}) \leq D_{max}, \tag{12}$$

where $R(P_0,\ldots,P_{M-1})$ is the total rate which is defined as follows,

$$R(P_0,\ldots,P_{M-1}) = \sum_{i=0}^{M-1} R_i(p_{i,0},\ldots,p_{i,N_{P_i}-1}), \tag{13}$$

and $D(P_0,\ldots,P_{M-1})$ is the total distortion which is defined as follows,

$$D(P_0,\ldots,P_{M-1}) = \max_{i\in[0,\ldots,M-1]} D_i(p_{i,0},\ldots,p_{i,N_{P_i}-1}). \tag{14}$$

Since the total rate $R(P_0,\ldots,P_{M-1})$ is the sum of the individual rates $R_i(p_{i,0},\ldots,p_{i,N_{P_i}-1})$, 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 Eq. (12),

$$\sum_{i=0}^{M-1}\left\{\min_{[p_{i,0},\ldots,p_{i,N_{P_i}-1}]\in B_i^{N_{P_i}}} R_i(p_{i,0},\ldots,p_{i,N_{P_i}-1}),\quad \text{subject to:}\quad D_i(p_{i,0},\ldots,p_{i,N_{P_i}-1})\le D_{max}\right\}, \tag{15}$$

which shows that the optimal solution to problem (12) can be found by solving the optimization problems for the different boundaries independently using the algorithm developed in section 3.

As in section 4 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. The minimum distortion problem can be stated as follows,

$$\min_{P_0,\ldots,P_{M-1}} D(P_0,\ldots,P_{M-1}),\qquad \text{subject to:}\qquad R(P_0,\ldots,P_{M-1})\le R_{max}. \tag{16}$$

By defining $R^*(D_{max})$ of section 4 as the minimum total rate needed to encode the M given boundaries with a maximum error of D_{max}, the derivation in section 4 still applies. Hence the resulting algorithm is a bisection search and at each iteration the optimization problem of Eq. (12) is solved optimally using the above proposed scheme.

6 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 8-connect chain code and a run-length encoding 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 1, the 8 closest neighbors of a given point P are:

$$\begin{array}{ccc}(3,1) & (2,1) & (1,1)\\ (4,1) & P & (0,1)\\ (-3,1) & (-2,1) & (-1,1).\end{array} \tag{17}$$

As an example, $(3,4)$ represents a straight line of 4 increments in the $3*\pi/4$ direction. Each of the possible symbols (α,β) gets a probability assigned and the resulting stream of increments $I=[(\alpha_1,\beta_1),\ldots,(\alpha_{N-1},\beta_{N-1})]$ 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 3 bits indicate one of the 8 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 $\pi/4$.

A generalization of this code is based on the observation that this scheme is optimal in the case where the probability mass function of (α,β) is separable and α is uniformly distributed over all 8 α's whereas β is geometrically distributed ($P(\beta=j)=\frac{1-\gamma}{\gamma}*\gamma^j, j\ge 1$) with parameter $\gamma=0.5$. The assumptions that the distribution is separable, that α is uniformly distributed and that 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 which 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 6 out of the 8 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 6 possible α's, and instead of using 3 bits to encode them only $\log_2(1/6) \approx 2.6$ bits are needed.

The question is therefore, 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,

$$P(\beta_1, \ldots, \beta_{N_P-1}) = \prod_{i=1}^{N_P-1} \frac{1-\gamma}{\gamma} * \gamma^{\beta_i}, \tag{18}$$

which leads to the following maximum likelihood estimate of γ,

$$\gamma_{ML} = 1 - \frac{N_P - 1}{\sum_{i=1}^{N_P-1} \beta_i}. \tag{19}$$

In section 3 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 on 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 bit 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 iterative procedure to estimate γ_{ML} which can be applied for the minimum rate case, but not for the minimum distortion case since we can only prove that it converges to a local minimum. The iteration works as follows: first an initial quantized $\gamma^Q_{ML,i}$, where i indicates the iteration number and Q the fact that this is a quantized value, is used 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. Then the quantized $\gamma^Q_{ML,i+1}$ is derived from $\gamma_{ML,i+1}$. These three steps are repeated until the minimum rate for the polygon of iteration $i+1$, $R^*_{i+1}(\gamma^Q_{ML,i})$ 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 0, we can prove that this iteration converges to a local minimum by showing that $R^*_{i+1}(\gamma^Q_{ML,i})$ is a non-increasing function of $\gamma^Q_{ML,i}$. Clearly $R^*_{i+1}(\gamma^Q_{ML,i}) \geq R^*_{i+1}(\gamma_{ML,i+1})$ 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, $\gamma^Q_{ML,i+1}$ can be found by evaluating the likelihood function for the two reconstruction levels which are the closest to $\gamma_{ML,i+1}$ and setting $\gamma^Q_{ML,i+1}$ 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 $\gamma^Q_{ML,i+1}$) can be inferred from the solution of a continuous problem (finding $\gamma_{ML,i+1}$). Therefore $R^*_{i+1}(\gamma^Q_{ML,i}) \geq R^*_{i+1}(\gamma^Q_{ML,i+1}) \geq R^*_{i+2}(\gamma^Q_{ML,i+1})$ which proves the convergence to a local minimum. Using an arithmetic coder and this iterative scheme, the efficiency of the minimum rate approach can be improved by about 15%.

7 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 $\gamma = 0.5$. First we present results for the minimum rate case for multiple objects. In Fig. 5 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 original segmentation requires 468 bits if encoded by an 8-connect chain code whereas the optimal segmentation can be encoded with only 235 bits. 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 lost, i.e., the boundary has been "straightened". In Fig. 6 we show the resulting segmentation for the minimum distortion case for multiple boundaries. The maximum rate R_{max} has been set to 280 bits and the optimal solution, which uses 274 bits for a $D_{max} = 0.71$ pixels, is displayed in the left figure. The right figure is a closeup of the lower boundary in the left figure and the stars indicate the original boundary with the polygonal approximation drawn on top of it.

8 SUMMARY

We presented a fast and optimal method for the lossy encoding of object boundaries which are given as 8-connect chain codes. We approximated the boundary by a polygon and derived a fast algorithm for finding the polygon which can be encoded with the smallest bit rate for a given maximum distortion. We investigated the dual problem of finding the polygonal approximation which leads to the smallest distortion for a given maximum bit rate. We presented an iterative scheme and proved that this scheme converges to the optimal solution. We extended the proposed algorithms to the encoding of multiple object boundaries and introduced a new vertex encoding scheme. Finally we presented results of the proposed algorithms using objects from the "Miss America" sequence.

9 REFERENCES

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

[2] A. K. Jain, *Fundamentals of Image Processing.* Prentice-Hall, 1989.

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

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

[5] J. Koplowitz, "On the performance of chain codes for quantization of line drawings," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-3, pp. 180–185, Mar. 1981.

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

[7] T. Kaneko and M. Okudaira, "Encoding of arbitary curves based on the chain code representation," *IEEE Transations on Communications*, vol. COM-33, pp. 697–707, July 1985.

[8] 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 Conference on Communication, Computers and Signal Processing*, (Victoria, Canada), pp. 138–142, June 1989.

[9] T. Özçelik, *A Very Low Bit Rate Video Codec.* PhD thesis, Dept. EECS, Northwestern University, Dec. 1994.

[10] T. Cormen, C. Leiserson, and R. Rivest, *Introduction to Algorithms.* McGraw-Hill Book Company, 1991.

[11] C. F. Gerald and P. O. Wheatley, *Applied Numerical Analysis.* Addison Wesley, fourth ed., 1990.

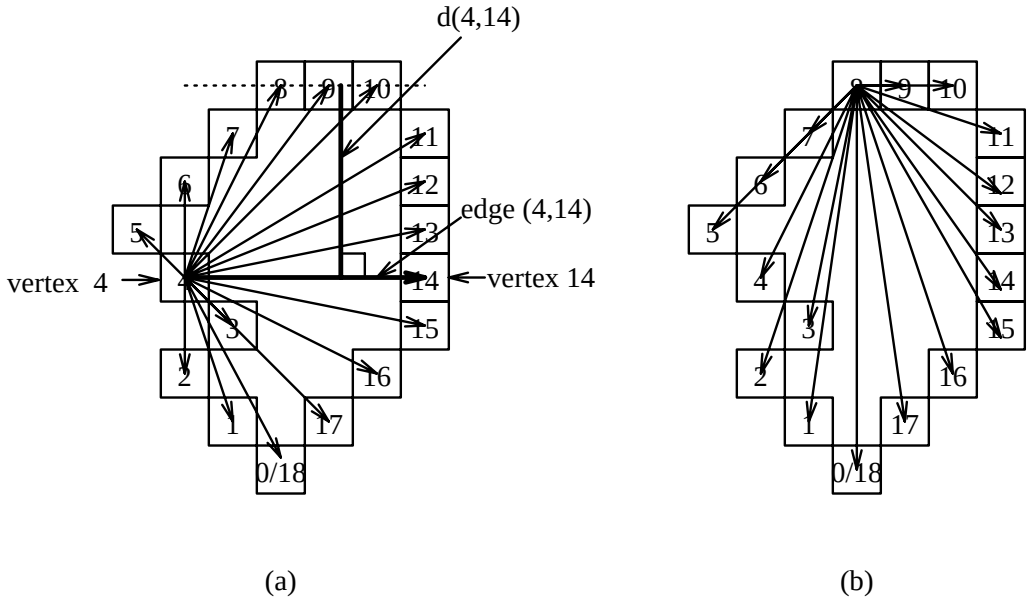

Figure 1: Interpretation of the boundary and the polygon approximation as a fully connected weighted directed graph. Note that the set of all edges E equals $\{(b_i, b_j) \in B^2 : i \neq j\}$. Two representative subsets are displayed: (a) $\{(b_4, b_j) \in B^2 : \forall j \neq 4\}$ and (b) $\{(b_8, b_j) \in B^2 : \forall j \neq 8\}$.

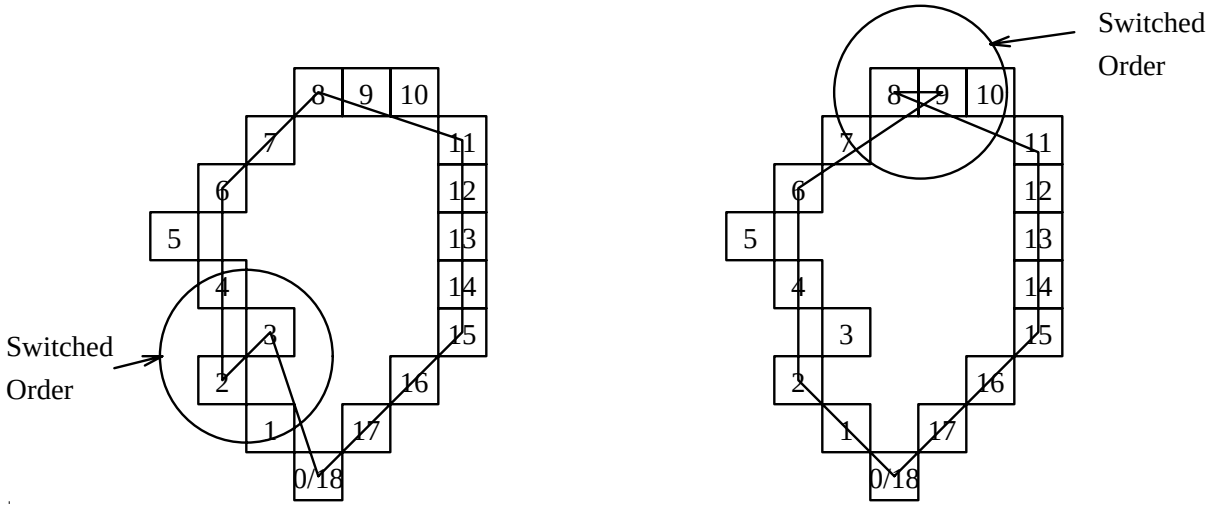

Figure 2: Examples of polygons with rapid changes in direction.

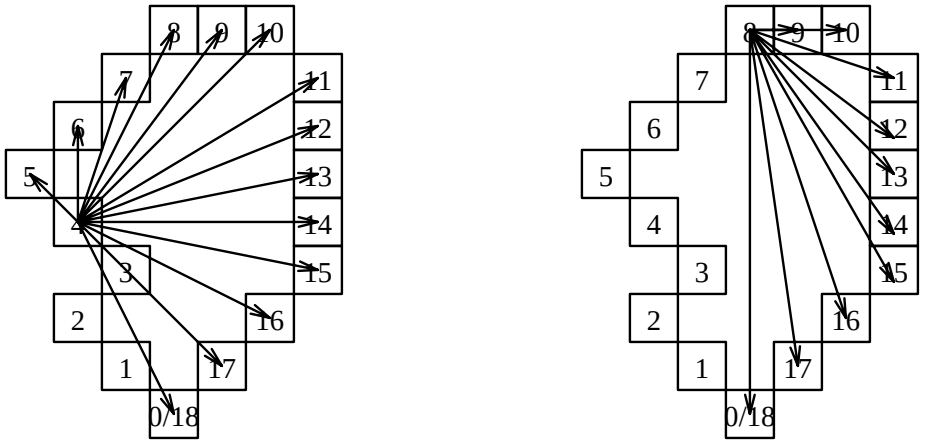

Figure 3: Interpretation of the boundary and the polygon approximation as a weighted directed graph. Note that the set of all edges E equals $\{(b_i, b_j) \in B^2 : i < j\}$. Two representative subsets are displayed: (a) $\{(b_4, b_j) \in B^2 : \forall j > 4\}$ and (b) $\{(b_8, b_j) \in B^2 : \forall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

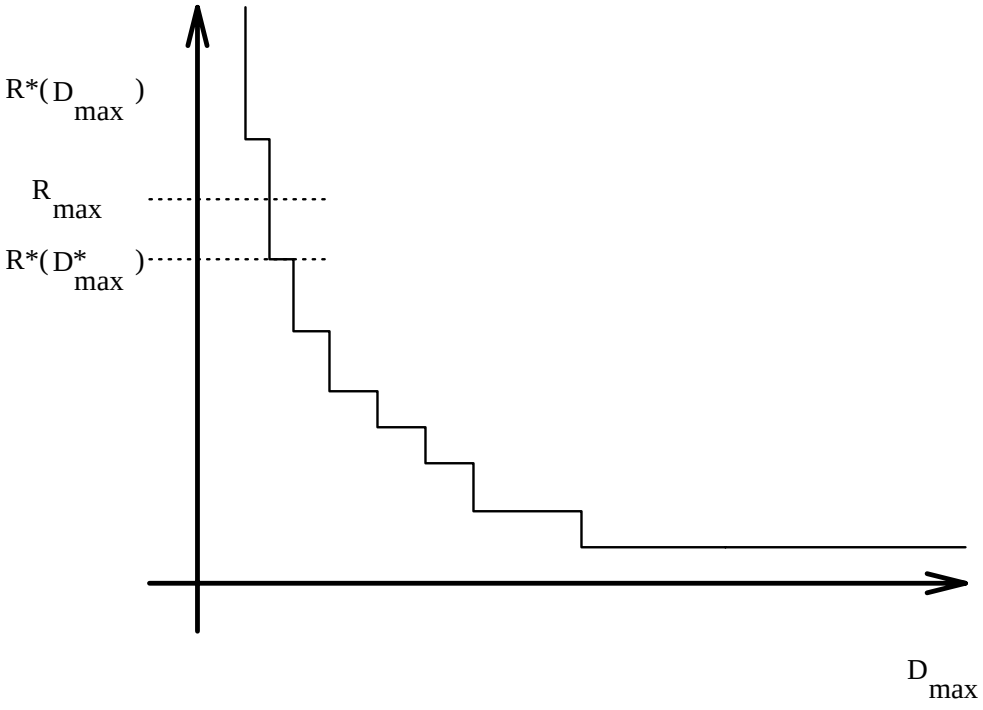

Figure 4: The $R^*(D_{max})$ function, which is a non-increasing function exhibiting a staircase characteristic. The selected R_{max} falls onto a discontinuity and therefore the optimal solution is of the form $R^*(D^*_{max}) < R_{max}$, instead of $R^*(D^*_{max}) = R_{max}$.

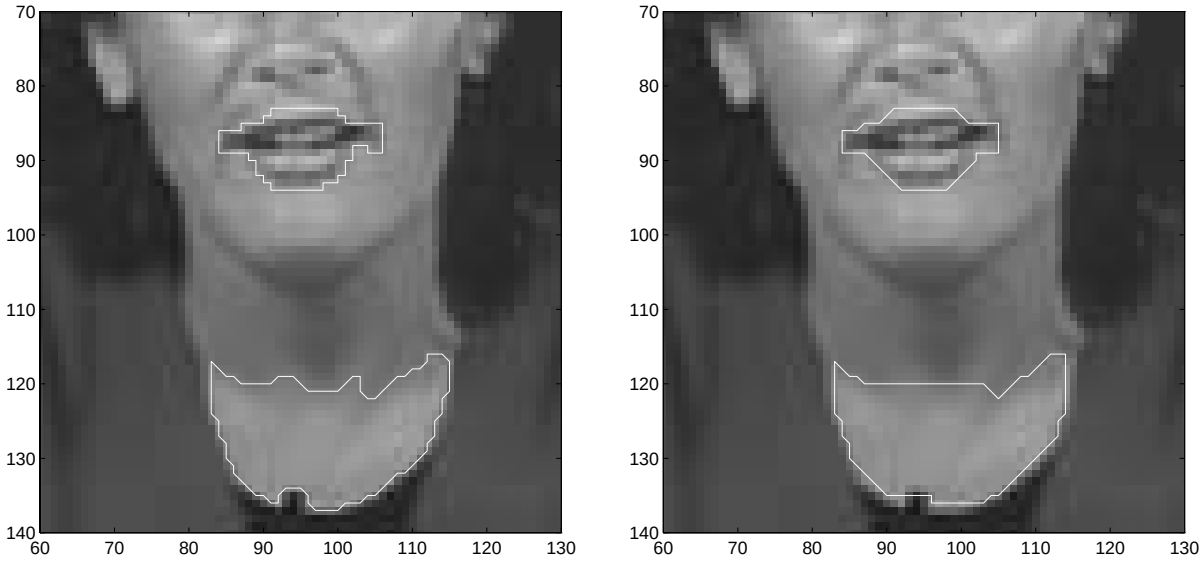

Figure 5: Left figure: original segmentation which requires 468 bits using the 8-connect chain code. Right figure: optimal segmentation with $D_{max} = 1$ pixel which requires a rate of 235 bits and results in a distortion of 1 pixel.

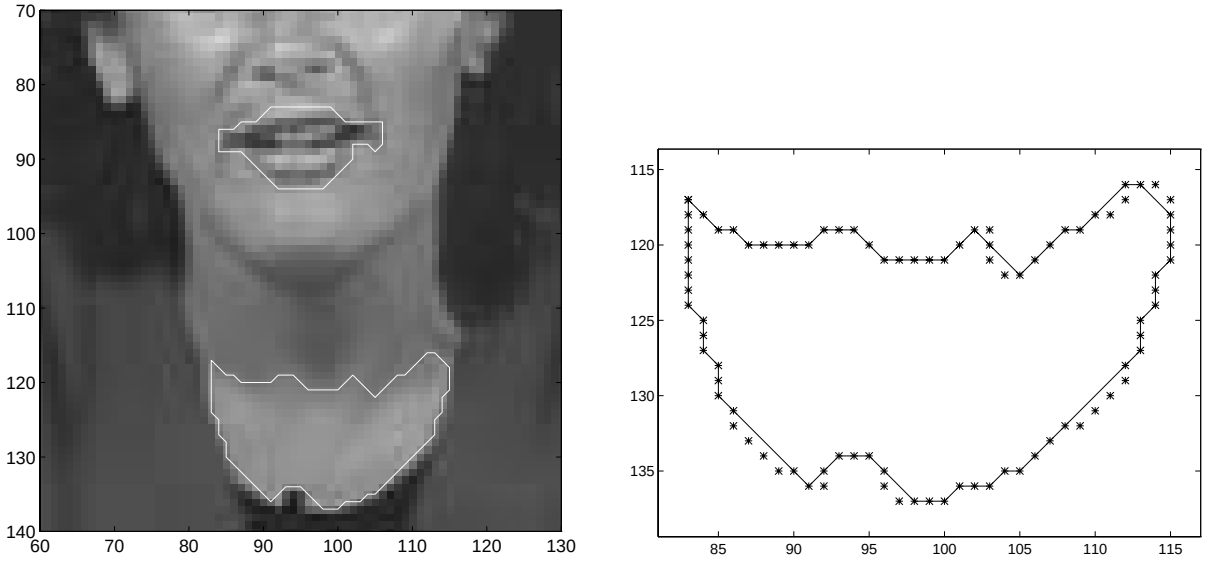

Figure 6: Left figure: optimal segmentation with $R_{max} = 280$ bits which results in a distortion of 0.71 pixels and a bit rate of 274 bits. Right figure: closeup of the lower boundary; the stars indicate the original boundary and the line represents the polygonal approximation. The upper left corner has been selected as the first vertex.