# Operationally Optimal Vertex-Based Shape Coding

Guido M. Schuster

3COM, Carrier Systems Business Unit
Advanced Technologies Research Center
1800 W. Central Rd.
Mount Prospect, Illinois 60056-2293
Tel: (847) 222-2486
Fax: (847) 222-0573
E-mail: guido_schuster@3com.com

Gerry Melnikov and Aggelos K. Katsaggelos

Northwestern University
Department of Electrical and Computer Engineering
McCormick School of Engineering and Applied Science
Evanston, Illinois 60208-3118
Tel: (847) 491-7164
Fax: (847) 491-4455
E-mail: {gerrym,aggk}@ece.nwu.edu

**Abstract**

In this paper, we present a review of our work on rate-distortion based operationally optimal vertex-based lossy shape encoding techniques. We approximate the boundary of a given shape by a low order curve, such as a polygon, and consider the problem of finding the approximation which leads to the smallest distortion for a given number of bits. We also address the dual problem of finding the approximation 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 (such as the maximum distance between a boundary and its approximation) and the second class is based on the summation operator (such as the total number of error pels between a boundary and its approximation). For the first class, we derive a fast and operationally optimal scheme which is based on a shortest path algorithm for a weighted directed acyclic graph. For the second class, we propose a solution approach which is based on the Lagrange multiplier method, which uses the above mentioned shortest path algorithm. Examples illustrate the effectiveness of the various techniques which are presented.

# List of Figures

# 1 Introduction

Multimedia communications has become one of the fastest growing sectors of the communications industry. While the current drivers are primarily video streaming and video conferencing, a number of new applications are in the horizon. Such applications, like multimedia libraries and interactive chat rooms, will need access to video data on an object by object basis, where video objects are defined by their shape, texture and motion.

Current video coding standards, such as MPEG-1, MPEG-2, H.261, and H.263 are block-based codecs. The scene is divided into equal-sized square blocks for which the texture and motion are encoded. No shape encoding is necessary, since the block size is known *a priori*. Unfortunately, a bit stream of such a coder does not lend itself to object based interaction with the data, since the shape of the objects (and therefore the objects themselves) is not defined.

Object based description of video is quite natural and there is a large amount of previous work. For example, most of the source material generated in a broadcast studio is based on the so-called chroma-keying technique, where an actor is filmed in front of a blue screen and then moved in front of any desired background. In [1] a second generation image coding scheme was formulated, where the segmentation of the image is transmitted explicitly. The underlying assumption is that the contours of regions are very important for subjective image quality, whereas the texture of the regions is of lower importance. Object-based analysis-synthesis coding (OBASC) [2], which is a video coding paradigm, also follows the idea that the shape of moving objects is more important than the texture and that geometric distortions are less noticeable for a human observer than the coding artifacts of traditional block coders.

Object based treatment of video sequences, necessitated by the emerging content driven applications, requires efficient representation of object boundaries. The ultimate goal is to allocate an available bit budget optimally among the video scene components (shape, texture, motion) and within each component. Here, we address the issue of operationally optimal shape encoding, which is a step in the direction of globally optimal resource allocation in the object-oriented video.

As already mentioned, the current video standards do not have the capability of encoding the shape of a video object. This is one of the aspects the upcoming MPEG-4 Visual standard will address [3, 4, 5, 6]. For the first time an international standard will enable the transmission of arbitrarily shaped video objects (VO). For a review of the upcoming MPEG-4 shape coding standard and a comparison with the

operationally optimal shape coding techniques discussed in this paper, the reader is referred to [7].

It is common practice in computer graphics to define the shape of an object using a so called $\alpha$ map, $M_k$, given by

$$M_k = \{m_k(x,y) | 0 \leq x \leq X,\ 0 \leq y \leq Y\},\ 0 \leq m_k \leq 255. \tag{1}$$

This alpha map is of the same size as the frame, which is of dimension $X \times Y$ pels. It specifies whether a certain pel is part of the visual object (VO) ($m_k(x,y) > 0$) or not ($m_k(x,y) = 0$), and whether the pel is transparent ($m_k(x,y) < 255$) or opaque ($m_k(x,y) = 255$). While in computer graphics transparent objects are important, for the purpose of video description and coding, opaque objects are more common, and hence most shape coders are designed for opaque objects.

There are two major classes of shape coders, bitmap based coders and contour based coders. Bitmap based coders look at the $\alpha$ map as just another black and white (or gray scale) image which needs to be encoded efficiently. One can argue that to some degree these coders break the object oriented paradigm. Bitmap-based shape coders are used in the very popular group 3 fax standard [8] and in the emerging bi-level standard JBIG [9] as well as in MPEG-4. The contour-based coders, on the other hand, try to encode the boundary of the object, by following its outline. The proposed operationally optimal techniques belong to this class of shape coders. In the following we provide a brief review of contour-based coders. Various bitmap-based and contour-based shape coders are discussed in detail in [7].

One of the first contour coders was proposed by Freeman [10]. He suggested the use of chain coding for boundary quantization and encoding, which has attracted considerable attention over the last thirty years [11, 12, 13, 14, 15]. The curve is quantized using the grid intersection scheme [10] 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. For lossless encoding of boundary shapes, an average 1.2 bits/boundary pel and 1.4 bits/boundary pel are required respectively for a 4 and an 8 neighbor grid [16]. There have been many extensions to this basic scheme such as the generalized chain codes [11], where the coding efficiency has been improved with the use of links of different length and different angular resolution. In [14] a scheme is presented which utilizes patterns in a chain code string to increase the coding efficiency, and in [15] 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 [12] the performance of

different quantization schemes is compared, whereas in [13] 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 pel accuracy.

A polygon-based shape representation was developed for OBASC [17, 18]. As a quality measure, the Euclidean distance $D_{max}$ between the original and the approximated contour was used. During subjective evaluations of CIF (352×288 pels) video sequences, it was found that allowing a peak distance of $D^*_{max} = 1.4$ pel is sufficient to allow proper representations of objects in low bitrate applications. Hence the lossy polygon approximation was developed. The polygon approximation is computed using the two contour points with the maximum distance between them as the starting point. Then, points are added to the polygon where the approximation error between the polygon and the contour is the highest. This is repeated until the shape approximation error is less than $D^*_{max}$. In a last step, splines are defined using the polygon points. If the spline approximation does not result in a larger approximation error between two neighboring polygon points, the spline approximation is used. This leads to a smoother representation of the shape. Vertex coordinates and the curve type between two vertices are arithmetically encoded.

Several other optimal, as well as, greedy non-optimal vertex-based curve approximation algorithms have been proposed in the literature. With a fixed number of approximating line segments and no restriction on their connectivity, an optimal (with respect to distortion) segment placement problem was solved by Bellman in [19]. In [20, 21] a *fan-based* approach, based on the concept of overlapping visibility regions, was used in which a greedy minimax strategy arrived at a polygonal approximation with a small number of connected line segments and end-points on the original curve. The same problem was solved optimally in [22] with the use of dynamic programming. The rate for encoding the vertices of the polygons was not taken into account, therefore these approaches are not optimal in the rate-distortion sense. In [23] this approach was extended to allow the line end-points to be off the original boundary and DPCM encoding of the control points was taken into account in the cost minimization process.

In [24, 25] 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 between the boundary and the approximation. This is an appropriate procedure when the smoothing of the boundary is the main objective. When, however, the encoding of the resulting control points is taken into account, the tradeoff between the encoding cost and the resulting distortion needs to be considered. By selecting the mean squared error 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.

The above methods for polygon/spline representation achieve good results but they cannot claim optimality. In this paper we propose a framework for the operationally optimal contour encoding of object boundaries in the intra mode. The techniques discussed here, even though dealing with still boundary approximations, are designed to be a part of a video codec. Hence, throughout the paper, we refer to video coding as the target application, but we concentrate on the intra mode of operation and do not utilize the temporal correlation between frames for shape prediction. While optimal shape coding techniques are able to outperform existing shape coders, the proposed framework also allows for additional capabilities. For example, these techniques allow for the optimal bit allocation among objects in a scene and/or sequence [26], simple rate control, combination of automatic segmentation and compression [27], etc.

The techniques presented in this paper for the operationally optimal boundary encoding can also be applied to the near-lossless encoding of one-dimensional signals. An example is represented by the work in [28, 29], where ECG signals are compressed using a min-max approach. They can even be applied to the near-lossless encoding of two-dimensional signals. In [23] a piecewise linear approximation was used by converting an image into a 1-D signal with a zig-zag or a Hilbert scan. In this framework, a near-lossless image compression problem is easily formulated using the minimum maximum criterion. This may be the coding technique of choice for applications requiring a guaranteed accuracy of approximation at every pixel, as opposed to a global distortion measure, like the mean-squared error.

The rest of the paper is organized as follows. In section 2 we define the problem mathematically and introduce the necessary notation. In section 3 we present the basic idea behind the proposed algorithms. In section 4 we discuss the constraints imposed on the code used to encode the approximation. In section 5 we introduce a definition of distortion which fits into the proposed framework. In section 6 we introduce the directed acyclic graph (DAG) formulation of the problem, which results in a fast solution approach. In section 7 we show how the DAG algorithm can be used to find the approximation with the minimum maximum segment distortion for a given rate. In section 8 we show how the DAG algorithm can be used to find the approximation with the smallest total distortion for a given rate. In section 9 we present

experimental results and in section 10 we conclude the paper and point out directions for future research.

## 2 Problem Formulation

In this section we formulate the problem mathematically and introduce the necessary notation. For the purpose of the discussion, we focus on the approximation of a given boundary by a polygon, as depicted in Fig. 1. We note here, that the proposed shape encoding techniques are not restricted to polygons. In fact they can be extended to any order curve. We use second order B-spline curves to illustrate this point later on. As can be seen in Fig. 1, there are three sets involved in the proposed framework. The boundary set $B$, the control point set $P$ and the admissible control point set $A$. While $B$ is given, we are trying to find $P$, which must be a subset of $A$, such that the resulting approximation curve is as close as possible to $B$, (measured by a distortion measure, discussed in section 5), for a given bit rate necessary to encode $P$ (using a code satisfying the constraints discussed in section 4).

We use the notation shown in Fig. 1. We denote by $B = \{b_0, \ldots, b_{N_B-1}\}$ 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, as shown in Fig. 1, $b_0 = b_{N_B-1}$. We denote by $P = \{p_0, \ldots, p_{N_P-1}\}$ the curve control points used to approximate $B$, which is also an ordered set. Note that for a polygon, $p_k$ is the $k$-th vertex of $P$ and $N_P$ the total number of vertices in $P$. Since $P$ is an ordered set, the ordering rule and the set of control points uniquely define the approximation.

Having established the above notation, operationally optimal shape coding can now be formalized as the solution to the following discrete, constrained optimization problem:

$$\min_{p_0,\ldots,p_{N_P-1}} D(p_0, \ldots, p_{N_P-1}), \quad \text{subject to:} \quad R(p_0, \ldots, p_{N_P-1}) \leq R_{max}, \tag{2}$$

where $D(\cdot)$ is the distortion measure, quantifying the error between the approximation and the original, $R(\cdot)$ is the bit rate required to encode the control points $P$, and $R_{max}$ is the maximum bit rate permitted for the encoding of the boundary. We also present algorithms which solve the dual problem, that is,

$$\min_{p_0,\ldots,p_{N_P-1}} R(p_0, \ldots, p_{N_P-1}), \quad \text{subject to:} \quad D(p_0, \ldots, p_{N_P-1}) \leq D_{max}, \tag{3}$$

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 (2) and (3) 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 (3) is used iteratively to solve problem (2).

### 2.1 Definition of the admissible control point set $A$

In general, the curve used to approximate the boundary could be permitted to place its control points ($P$) anywhere on the plane. The optimization algorithm for this problem to be presented later, however, has a time complexity which is quadratic (for the polygon case) in the number of admissible control points. Hence we would like to keep this number as small as possible, without sacrificing coding efficiency. We therefore introduce the set of all admissible control points $A$, defined as the set of all the pels which are within a given maximum distance $DM$ from a boundary point (see Fig. 1). Beside keeping the cardinality of $A$ small, this also forces the approximation to closely follow the boundary, since all admissible control points are quite close to the original boundary. Hence the set of admissible control points consists of a band of width $2 * DM$ around the original boundary. Clearly, the bit rate for the operationally optimal approximation is a non-increasing function of the size of the proposed band. On the other hand, as mentioned above, the time complexity of the optimization algorithm is quadratic (for the polygon case) in the number of the admissible control points and the larger the band, the bigger the set of admissible control points.

We propose the following procedure for defining the set of admissible control points $A$ and for assigning control points to boundary points. Since the boundary set $B$ is ordered, this assignment is used to order the admissible control point set $A$.

1) for $j = 0, \ldots, N_B - 1$;
2) $\quad a_{j,0} = b_j$;
3) $i = 1$;
4) while ( $||v_i|| \leq DM$ )
5) $\quad$ for $j = 1, \ldots, N_B - 2$;
6) $\quad\quad$ if $\{b_j + v_i\} \notin A$
7) $\quad\quad\quad a_{j,i} = b_j + v_i$;
8) $\quad i = i + 1$;

The $v_i$'s in the above algorithm are offset vectors, which are displayed in Fig. 2. The offset vector $v_j$ starts at 0 and ends at $j$. Note that the distance $||v_j||$ is a non-decreasing function of $j$. The basic idea behind the proposed algorithm is to draw spirals around the boundary points $b_j$, where the spirals are drawn in discrete increments of pels. After a spiral is advanced by one increment, the new pel $b_j + v_i$ is an

admissible control point. If this point is not already in the admissible control point set $A$, it is added as the element $a_{j,i}$. Note that the subscripts identify the nearest boundary point, and its relative offset. This is used later on to define the segment distortion. This procedure is repeated for the next boundary point until there are no more boundary points. Then the procedure starts with the second boundary point again until the distance is greater than $DM$. Note that the first and last boundary points are the first and the last admissible control points. This forces the approximation to start and end exactly where the original boundary starts and ends.

In the above algorithmic description of this scheme, the inner loop assigns one admissible control point $a_{j,i}$=$b_j$+$v_i$ to each boundary point $b_j$. This loop is repeated for the next vector increment $v_{i+1}$ as long as the admissible control points are within the band defined by the maximum distance $DM$. In Fig. 3 different stages of the algorithm are shown. In this example, the maximum distance $DM$ is set to one pel and therefore only the vectors $v_1$, $v_2$, $v_3$ and $v_4$ are used in the algorithm, since all other vectors result in a distance larger than one pel. Figure 3a shows the admissible control points and their assignment to the boundary points after the vector increment $v_1$ has been used for all boundary points. Figure 3b shows the admissible control points and their assignment after the vector increment $v_2$ has been used for all boundary points. Figure 3c shows the admissible control points and their assignment after the vector increment $v_3$ has been used for all boundary points, and Fig. 3d shows the admissible control points and their assignment after the vector increment $v_4$ has been used for all boundary points, which is the final admissible control point set. This set then has the following structure, where the order is imposed by the original boundary set order, $A = \{a_{0,0}, a_{1,0}, a_{1,1}, a_{1,2}, a_{2,0}, a_{2,1}, a_{2,2}, a_{2,3}, a_{3,0}, a_{3,1}, \ldots, a_{18,0}\}$.

## 3 Proposed framework

In this section, we discuss the basic idea behind the proposed shape encoding approaches. The goal is to give the reader some insight into the development of the schemes, instead of just presenting the resulting algorithms. We use the definitions which were introduced in the previous section and concentrate on polygons. It is clear that the number of possible polygons is finite, but extremely large. If we define the smallest possible polygon as a single point, then, given the degree of the polygon ($N_P$), there are $\binom{N_A}{N_P} = \frac{N_A!}{(N_A-N_P)!\cdot N_P!}$ different selections of $N_P$ vertices from the admissible control points set $A$. 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

$$\sum_{k=1}^{N_A} \frac{N_A!}{(N_A - k)! \cdot k!}. \tag{4}$$

Clearly an exhaustive search is not a feasible approach and we need to look for a fast formulation of the solution to the problem.

The fast graph algorithm we introduce in section 6 is based on the following observation. If we fix the current vertex of a polygon, then changing future vertices does not change the polygon up to and including the current vertex (see Figure 4a). In other words, knowing the position of the current vertex, makes the past independent of the future. Clearly, the knowledge of the current vertex represents a state of this system. Therefore, given we have an optimal approximation for all the previous vertices, finding the optimal approximation for the current vertex simply requires the selection of the optimal approximation between all previous vertices and the current one.

In the case of a polygonal approximation, the curve between two state values is simply a straight line, as can be seen in Figure 4a. For higher order curves, such as a B-spline, the situation is a bit more complicated. Before continuing, it is worthwhile to present a review of B-splines. These are a family of parametric curves which have proven to be very useful in boundary encoding [7]. In the following discussion, we concentrate on second order B-splines. However, the presented theory can be generalized to higher order curves. Also, it should be noted that first order B-splines are equivalent to polygons.

A B-spline is a specific curve type from the family of parametric curves [30]. A parametric curve consists of one or more curve segments. Each curve segment is defined by $(n + 1)$ control points where $n$ defines the degree of the curve. The control points are located around the curve segment and together with a constant base matrix $M$ they solely define the shape of the curve. A two dimensional curve segment $Q_u$ with control points $(p_{u-1}, p_u, p_{u+1})$ is defined in the $x - y$ plane as follows:

$$Q_u(p_{u-1}, p_u, p_{u+1}, t) = \left[ \; x(t), \quad y(t) \; \right], \text{ for } 0 \leq t \leq 1, \text{ and 0 otherwise.} \tag{5}$$

The points at the beginning and the end of a curve segment are called knots and can be found by setting $t = 0$ and $t = 1$. The following is the definition for a second degree curve segment, with $u$ the index for the different curve segments, and $p_{u,x}$ and $p_{u,y}$, respectively, the horizontal and vertical coordinates of control

point $p_u$,

$$Q_u(p_{u-1}, p_u, p_{u+1}, t) = T \cdot M \cdot P$$
$$= \begin{bmatrix} t^2 & t & 1 \end{bmatrix} \cdot \begin{bmatrix} m_{1,1} & m_{1,2} & m_{1,3} \\ m_{2,1} & m_{2,2} & m_{2,3} \\ m_{3,1} & m_{3,2} & m_{3,3} \end{bmatrix} \cdot \begin{bmatrix} p_{u-1,x} & p_{u-1,y} \\ p_{u,x} & p_{u,y} \\ p_{u+1,x} & p_{u+1,y} \end{bmatrix}. \quad (6)$$

Both, the base matrix $M$, with specific constant parameters for each specific type of a parametric curve, and the control point matrix $P$, with $(n+1)$ control points, define the shape of $Q_u$ in a two dimensional plane. Every point of the curve segment can be calculated with Eq. (6) by letting $t$ vary from 0 to 1. Every curve segment can be calculated independently in order to calculate the entire curve $Q$, consisting of $N_P$ curve segments, which is of the following form,

$$Q(t') = \sum_{u=1}^{N_P} Q_u(p_{u-1}, p_u, p_{u+1}, t' - u + 1), \text{ with } 0 \leq t' \leq N_P + 1. \quad (7)$$

Among common parametric curves are the Bezier curve and the B-spline curve. For the presentation in this paper we chose a second order (quadratic) uniform non-rational B-spline curve [30] with the following base matrix,

$$M = \begin{bmatrix} 0.5 & -1.0 & 0.5 \\ -1.0 & 1.0 & 0.0 \\ 0.5 & 0.5 & 0.0 \end{bmatrix}. \quad (8)$$

Figure 5 shows such a second order B-spline curve. The shape coding method presented in this paper is independent of the matrix $M$ and degree $n$, that is, parametric curves of higher order can be used.

The beginning and the end of the boundary approximation have to be treated as special cases, if the first curve segment should start exactly at the first boundary point and the last curve segment should end exactly at the last boundary point. When we use a double control point (such as $p_{u-1} = p_u$) the curve segment $Q_u$ will begin exactly at the double control point (see Figure 5). We apply this property to the beginning and the end of the curve, so that $p_0 = p_1$ and $p_{N_P} = p_{N_P+1}$. These two special cases can easily be incorporated into the boundary approximation algorithm.

Recall the argument made for the polygon case, which is based on the fact that knowledge of the location of the current vertex, decouples the future from the past. A similar argument can be made for higher order curves, such as the above introduced B-spline curves. Consider Fig. 4b which shows a second order B-spline curve. As mentioned above, three control points completely define a curve segment. Hence if we know the last two control points, then the future is independent from the past. In other words, when the last two control points are fixed, then the approximation up to and including these two control points does not change if future control points are changed. Clearly in a second order B-spline curve approximation,

a state of the system contains the last two control points. In general, the order of the curve is equivalent to the dimensionality of the state space. As we will see later on, the dimensionality is in the exponent of the time complexity of the proposed algorithms, and hence it should be kept as small as possible.

In summary, the basic idea behind the proposed algorithms is the fact that one can define a state (of small dimensionality) which makes the future of the optimization process independent from its past. Our definition of the state varies according to the order of the curve used to generate approximating segments. Thus, for example, with a second-order B-spline approximation, a state consists of two consecutive control points, along with cost information. Since dimensionality of the state space directly depends on the curve used for the approximation, we will carefully constrain the code used for encoding the control points, so that the order of prediction in a DPCM scheme does not require knowledge of control points not in the state. Hence the state space dimension does not expand due to encoding. With this formulation, we will be able to write the total rate as a sum of the segment rates. A similar concept applies to the definition of distortion, which we will also select such that the state space does not expand. In other words, we will be able to write the distortion of the approximation as a function of segment distortions. Note the importance of the segment concept. A segment is the curve unit between two consecutive state values. In the case of a polygon, a segment is just an edge between two vertices (see the line with the two arrows in Fig. 4a). In the case of a second order B-spline curve, a segment is defined in Eq. 6, and drawn with arrows in Fig. 4b. We will define the segment rate and the segment distortion in the next two sections.

# 4 Rate

In this section we discuss the constraints imposed on the code structure by the proposed framework. As we pointed out in the previous section, the order of the approximation curve requires a certain dimensionality of the state space. In general, there is a one to one correspondence between the order of the curve and the required state space dimension. Note that the smaller the state space, the faster the optimal solution can be found.

A straightforward way of encoding the control points $P$ is with the use of a fixed length code for each point. Clearly this is not very efficient, since consecutive control points are correlated. This correlation suggests the use of differential pulse code modulation (DPCM) encoding of the points. In other words, the current point is predicted, using previous points, and only the difference between the prediction and the current point is encoded. Because of the correlation among neighboring control points, the prediction

error will be small in general, and can be coded efficiently using an entropy code. Assuming that the order of the predictor is $o$, then the rate for encoding the $k$-th segment is denoted by $r(p_{k-o}, \ldots, p_k)$. In other words, the rate for encoding the $k$-th segment is defined as the rate of encoding the $k$-th control point (see Figure 4), which depends on the position of the last $o$ control points, since those are used in the prediction of $p_k$. Clearly for a first order prediction, the rate to encode segment $k$ only depends on the current and previous control points. Furthermore, for a second order prediction, the segment rate only depends on the current, and the two previous control points. The total rate can now be written as the sum of the individual segment rates, that is,

$$R(p_0, \ldots, p_{N_P-1}) = \sum_{k=0}^{N_P-1+o} r(p_{k-o}, \ldots, p_k), \tag{9}$$

where all control points $p_i, i < 0$ are equal to $p_0$, and all control points $p_j, j > N_P - 1$ are equal to $p_{N_P-1}$. This forces the higher order curves to start at $p_0$ and stop at $p_{N_P-1}$. Note that the rate $r(p_{-o}, \ldots, p_0)$ is the rate used to encode the first point $p_0$. Further note that it is already known at the receiver that the last point is repeated $o$ times. Hence the rates $r(p_{k-o}, \ldots, p_k)$ for $k > N_P - 1$ are zero.

By using DPCM which is of the same order as the curve used for the approximation, the total rate in Eq. (9) and the curve share the same state. In other words, the same arguments on the independence of the future of the past, given knowledge of the present, also applies to the above total rate expression. Again, we are able to write the total rate as a function of its segment rates, which allows us to define a compact state, to be used for the formulation of a fast optimization algorithm.

# 5 Distortion

In this section we discuss various ways for defining the distortion between the original boundary and its approximation. Ideally, the definition of distortion should correspond to the visual quality of the approximation perceived by a human observer. Unfortunately, it is not well understood how a human observer judges the quality of a boundary approximation, especially within the framework of a video scene, with many different objects moving at different speeds in different directions.

Nevertheless, there are mathematical distortion measures which are very meaningful, such as the maximum distance between the original boundary and the approximation and/or the total area between the original boundary and the approximation [26]. Note that similar to the rate, we propose to express the total distortion as a function of the segment distortions. By doing so, we will be able to formulate the total

distortion using the same state, used for the rate and the approximation curve. Before we can define the total distortion, we need to define the segment distortions. Again, the decomposition of the total distortion into segment distortions is crucial for our ability to formulate a fast optimization procedure.

Consider Figure 6, which shows three popular segment distortions for polygons. Figure (a) shows the maximum segment distance between the original boundary and the polygon edge. Note that each admissible control point is associated with its nearest boundary point ($p_k$ with $b_e$ and $p_{k-1}$ with $b_a$). This was accomplished by the admissible control point set construction algorithm we introduced in section 2.1. Besides the fact that the maximum distance has perceptual relevance, it is also easy to calculate for a polygonal approximation [26]. Figure (6b) shows an area-based segment distortion measure. The area is defined by the boundary between $b_a$ and $b_e$, the line between $b_e$ and $p_k$, the polygon edge between $p_k$ and $p_{k-1}$ and the line between $p_{k-1}$ and $b_a$. There are different ways one can calculate this area, which are either based on a continuous or a discrete model [31]. In the continuous case, the boundary is thought of as a polygon and hence the area can be found by computational geometry. Clearly, the boundary is discrete and usually the polygon edge is also quantized to pel accuracy when displayed, hence one can count the pels of the error area and use this count as a distortion measure. This approach will lead to the same distortion measure as the one used in MPEG-4 [7]. Figure (6c) shows a different approach which is based on the idea of a distortion band. Basically, we draw a band of a given maximum distortion around the original boundary, and if an edge stays inside this band, then its maximum distortion is acceptable. From a computational point of view, this is a fast method of defining the segment (and ultimately the total) distortion, but special care needs to be taken (using a sliding window) so that no trivial solutions are selected [32].

While Figure 6 shows three different segment distortion measures for a polygon approximation, Figure 7 shows them for a second order B-spline approximation. Note that the definition of a segment is slightly more complicated for higher order curves, since the control points do not directly correspond to the beginning and end of the segment. In other words, we need to find the boundary points which correspond to the beginning ($t = 0$) and end ($t = 1$) of the curve segment. For example, for $t = 0$, this can be accomplished by testing each boundary point between $b_a$ and $b_c$, and picking the one which is the closest to $t = 0$. We name that boundary point $b_b$. In a similar fashion, we find $b_d$. Clearly, having defined the curve segment and its associated boundary points, all three previous polygon segment distortion measures can be similarly defined for higher order curves. Note that the area for a second order B-spline approximation is outlined

as follows: The boundary between $b_b$ and $b_d$, the line from $b_d$ to $t = 1$, the B-spline between $t = 1$ and $t = 0$ and the line between $t = 0$ and $b_b$.

Independently of which of the above segment distortion measures is selected, they all depend directly on the order $o$ of the selected curve approximation, i.e., $d(p_{k-o}, \ldots, p_k)$. Clearly this was the goal of breaking the approximation into segments for which we can define a segment distortion. As mentioned above, we can now use these segment distortions in two different ways to calculate the total distortion.

We will treat two different classes of total distortion measures. The first class is based on the maximum operator (or equivalently, on the minimum operator) and is of the following form,

$$D(p_0, \ldots, p_{N_P-1}) = \max_{k \in [0,\ldots,N_P-1+o]} d(p_{k-o}, \ldots, p_k), \tag{10}$$

where $d(p_{-o}, \ldots, p_0)$ is defined to be equal to 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 and is of the following form,

$$D(p_0, \ldots, p_{N_P-1+o}) = \sum_{k=0}^{N_P-1+o} d(p_{k-o}, \ldots, p_k), \tag{11}$$

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

In the remainder of this paper we will show how both classes of distortion measures can be treated, using a directed acyclic graph formulation with a slightly different definition of the graph weights.

# 6 Directed acyclic graph (DAG) formulation

In this section we show how a directed acyclic graph (DAG) can be used to formulate the problem at hand. We first concentrate on the polygon approximation, where the set of admissible control points $A$ is equal to the original boundary $B$, which implies that $a_{j,0} = b_j$. Since the second subscript will always be zero for $A$=$B$, we will drop it in the notation. In this special case, there is a simple, one-to-one correspondence between graph vertices and control points (polygon vertices) and between graph edges and polygon edges. Recall that for a polygon approximation, the state we selected consists of the last control point. Then we showed that DPCM encoding of the control points, using a first order predictor, shares the same state as the polygon approximation. We then carefully defined the approximation distortion, such that it is perceptually relevant and uses that same state as the rate and the polygon. In this section we now use the

fact that these three entities share the same state to show how the selection of the optimal polygon can be formulated using a DAG. Having formulated the problem as a shortest path problem through a DAG allows for the use of a fast dynamic programming algorithm to find the operationally optimal solution.

For the proposed DAG formulation, we need to start the search for an optimal approximation at a given control point. If the boundary is not closed, clearly the first boundary point $a_0 = b_0$ has to be selected as the first control point $p_0$. For a closed boundary, there is freedom in selecting the first control point. Experiments have shown that the performance of the algorithm is insensitive to the selection of the first control point, and hence, we always use the first boundary point (which is arbitrary) as the first control point, even for a closed boundary.

Besides fixing the first control point of the approximation, we also require that the last control point $p_{N_P-1}$ be equal to the last point of the boundary $a_{N_A-1} = b_{N_B-1}$. This leads to a closed 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.

The optimization problems stated in Eqs. (2) and (3) can be formulated as a shortest path problem in a weighted directed graph $G = (V, E)$, where $V$ is the set of graph vertices, which is equivalent to the admissible control point set $A$, and $E$ is the set of edges (see Fig. 8). As stated above, in this example $A = B$ and hence $V = B$, since every boundary point can be a control point. 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 control points can represent a line segment of a valid polygon, the edge set $E$ is defined as follows, $E = \{(a_i, a_j) \in A^2 : \forall i \neq j\}$ (see Fig. 8). 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,

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

where $w(u, v) : E \rightarrow \mathcal{R}$ is a weight function, specifying the cost of traversing this edge from $u$ to $v$. By defining this weight function in different ways, we will be able to find the optimal approximation for class one and class two distortion measures. In all cases, the directed edge $(u, v)$ is associated with its segment rate $r(u, v)$ and its segment distortion $d(u, v)$, as can be seen in Fig. 9. The weight $w(u, v)$ is then a function of the segment rate and the segment distortion. The classical algorithm for solving such

a single-source shortest-path problem, where all the weights are non-negative, is Dijkstra's algorithm [33], which has time complexity $O(|V|^2 + |E|)$. This is a significant reduction compared to the time complexity of the exhaustive search. Recall that we defined the control point set as an ordered set for reasons which will now become apparent. We can further simplify the algorithm by observing that we would not like the optimal path to select an admissible control point $a_j$ as a control point when the last selected control point was $a_i$, where $i > j$. For example, 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. 10). Therefore we add the restriction that not every possible combination of $(a_i, a_j)$ represents a valid edge but only the ones for which $i < j$. Hence the edge set $E$ is redefined in the following way, $E = \{(a_i, a_j) \in A^2 : i < j\}$ (see Fig. 11). This restriction results in the fact that a given control point 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 approximation has to follow the original boundary without rapid direction changes, and second, 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 [33], 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 $O(|V| + |E|)$. Recall that in general $V$ equals $A$ and hence we would like to keep the set of admissible control points as small as possible, as discussed in section 2.1.

We now introduce the DAG-shortest-path algorithm. Let $C^*(a_i)$ represent the minimum cost to reach the admissible control point $a_i$, (which in this case is identical to the boundary point $b_i$, since $A = B$), from the source vertex $a_0$ via an approximation. Clearly $C^*(a_{N_B-1})$ is the cost of the shortest path from the source vertex $a_0$ to the sink vertex $a_{N_B-1}$. Let $q(a_i)$ be a back pointer which is used to remember the optimal path. Consider Fig. 12 which shows a simple DAG for a polygon approximation. To find the shortest path through this graph, (which selects the optimal control points) we use the algorithm below.

1) $C^*(a_0) = w(a_{-1}, a_0)$;
2) for $i = 1, \ldots, N_A - 1$;
3) $\quad C^*(a_i) = \infty$;
4) for $i = 0, \ldots, N_A - 2$;
5) $\quad$ for $j = i + 1, \ldots, N_A - 1$;
6) $\quad\quad$ calculate segment distortion $d(a_i, a_j)$;

7) calculate segment rate $r(a_i, a_j)$;
8) calculate $w(a_i, a_j)$ as a function of $d(a_i, a_j)$ and $r(a_i, a_j)$;
9) if $(C^*(a_i) + w(a_i, a_j) < C^*(a_j))$;
10) $C^*(a_j) = C^*(a_i) + w(a_i, a_j)$;
11) $q(a_j) = a_i$;

The optimal path (the control point set) $\{p_0^*, \ldots, p_{N_P-1}^*\}$ can be found by back tracking the pointers $q(a_i)$ in the following recursive fashion (by definition $p_{N_P-1}^* = a_{N_A-1} = b_{N_B-1}$ and $p_0^* = a_0 = b_0$),

$$p_{k-1}^* = q(p_k^*), \tag{13}$$

starting with $p_{N_P-1}^*$ and stoping when $p_0^*$ is reached. The formal proof of the correctness of the DAG-shortest-path algorithm, on which the above scheme is based, can be found in [33]. We will reason more intuitively how this approach works. In line (1) the cost for encoding the starting point of the boundary is assigned to the minimum cost of the first control point. In lines (2) and (3) the minimum cost for reaching any of the admissible control points is set to infinity. The "for loop" in line (4) selects the admissible control points in sequence as possible control points from which a polygon edge starts and the "for loop" in line (5) selects possible control points where the polygon edge ends. Hence these two loops select each edge in the edge set $E$ exactly once. Therefore the lines (6) to (11) are processed for every edge. The lines (6) to (8) are used to calculate the weight of the edge, $w(a_i, a_j)$. The most important part of this algorithm is the comparison in line (9). Here we test if the new cost, $C^*(a_i) + w(a_i, a_j)$, to reach admissible control point $a_j$, given that the last control point was $a_i$, is smaller than the smallest cost used so far to reach $a_j$, $C^*(a_j)$. If this cost is indeed smaller, then it is assigned as the new smallest cost to reach admissible control point $a_j$, $C^*(a_j) = C^*(a_i) + w(a_i, a_j)$. We also assign the back pointer of $a_j$, $q(a_j)$ to point to $a_i$ since this is the previous control point used to achieve $C^*(a_j)$. This algorithm leads to the operationally optimal solution because, as stated earlier, when the cost $(C^*(a_i))$ of a vertex $(a_i)$ is given, then the selection of the future vertices $(a_j,\ i < j < N_A)$ is independent of the selection of the past vertices $(a_k,\ 0 \leq k < i)$.

The analysis of the above algorithm shows that there are two nested loops which results in a time complexity of $\Theta(N_A^2)$. We use the number of edge distortion evaluations as measure for the time complexity, since this is the most time consuming operation.

The above derivation is focused on the polygon approximation where $A = B$. As we have shown, in this special case, there exists a natural connection between the polygon approximation and the required graph. We now extend this to the case where $A$ is constructed according to the algorithm presented in

section 2.1. Basically the same graph (see Fig. 12) can be used, but we need to extend the vertices of the graph to include all the admissible control points which are associated with the same boundary point, i.e., $a_j$ is replaced by the set $\{a_{j,0}, a_{j,1}, a_{j,2}, \ldots\}$. The extended graph is displayed in Fig. 13, where $a_0$ and $a_4$ were not extended since they are the source and sink vertices, equivalent to the beginning and end of the original boundary. In Fig. 13, we only show the edges originating in $a_{1,i}$ and terminating in $a2, j$, to keep the graph readable. Note that the edge $(a_1, a_2)$ in Fig. 12 is replaced by the shown set of edges $(a_{1,i}, a_{2,j})$, $\forall i, j$. To fill the entire graph with edges, this replacement is done for each edge in the original graph in Fig. 12.

The next extension of the above algorithm is for higher order curves. Recall that we defined the order of the curve to be equal to the order of prediction $o$ and we are using a second order B-spline as an example of a higher order curve. As discussed in section 3, the state of an order $o$ curve includes the last $o$ control points. We then showed that the same state can be used for encoding these control points with an order $o$ DPCM scheme. We proposed to define the total distortion as a function of segment distortions, such that the total distortion measure shares again the same state. We now discuss for a second order curve how the DAG needs to be changed to be able to find the optimal solution. To simplify the graph, we again assume $A = B$, and after we develop this graph it can be extended to include $A \supseteq B$, as was done above. As pointed out, the state consists the knowledge of the last two admissible control points $(p_{k-2} = a_i, p_{k-1} = a_j), j > i$, and hence all feasible values this state can take on are the vertices in the graph shown in Fig. 14. The edges between the vertices then represent a segment of the curve, for which we have defined a segment rate $(r(p_{k-2}, p_{k-1}, p_k))$ and a segment distortion $(d(p_{k-2}, p_{k-1}, p_k))$. These edges are than labeled with an edge weight $(w(p_{k-2}, p_{k-1}, p_k))$, (for clarity reasons, in Fig. 14, we only labeled five edges with their respective weight, but clearly, each edge has a weight associated with it) which is a function of the segment distortion and the segment rate. In the next section, we will discuss how to select the function so that the DAG shortest path algorithm can be used to find the optimal approximation for distortion measures of both classes. Note the difference in connectivity in Figs. 12 and 14. In Fig. 12, each vertex has an edge for each future vertex, while in Fig. 14, each vertex has an edge for each future vertex, sharing the middle control point. In other words, two consecutive vertices define three control points, which in turn is enough to define a segment for a second order curve. The rate and distortion of this segment is then assigned to the edge between the two vertices. Again, the DAG-shortest-path algorithm is used to find the optimal approximation, where for this example the order of the state is two, which results in a time complexity

of $\Theta(N_A^3)$. In general the time complexity of a curve approximation of order $o$ is $\Theta(N_A^{o+1})$. Therefore, the smaller the set of admissible control points, and the smaller the order of the approximation curve, the faster the algorithm. On the other hand, the larger the admissible control point set and the higher the order of the curve, the (potentially) better the approximation of the original boundary.

# 7 Distortion measures based on the maximum operator

In this section we introduce two algorithms to solve the problems stated in Eq. (2) and (3) for class one distortion measures, such as the maximum absolute distance. These algorithms are specific implementations of the general optimal bit allocation algorithm among dependent quantizers for the minimum maximum distortion criterion we proposed in [34]. Again, we first focus on the polygon case and then extend the result to higher order approximations. As pointed out before, we will employ the proposed DAG formulation to find the optimal approximation. The key to using this formulation is the definition of the weight function.

## 7.1 The minimum rate case

First we consider the minimum rate case which is stated in Eq. (3). The goal of the proposed algorithm is to find the approximation whose control points can be encoded with the smallest number of bits. This selection is constrained by the fact that the selected approximation must result in a distortion smaller or equal to the maximum distortion. To this end, we propose the following definition of the weight function

$$w(p_{k-1}, p_k) = \begin{cases} \infty & : \quad d(p_{k-1}, p_k) > D_{max} \\ r(p_{k-1}, p_k) & : \quad d(p_{k-1}, p_k) \leq D_{max} \end{cases} . \tag{14}$$

The above definition of the weight function leads to a length of infinity for every path which includes a segment resulting in an approximation error larger than $D_{max}$. Therefore the DAG shortest path algorithm will not select these paths. Every path which starts at vertex $a_0$ and ends at vertex $a_{N_A-1}$ and does not result in a path length of infinity, results in a path length equal to the rate of the approximation it represents. Therefore the shortest of all those paths corresponds to the approximation with the smallest bit rate which is the solution to the problem in Eq. (3).

Clearly, the extension of this to higher order schemes is straightforward, since, in previous sections, we took great care of defining the segment rate and segment distortion. Hence, in general, we propose the following definition of the weight function

$$w(p_{k-o}, \ldots, p_k) = \begin{cases} \infty & : \quad d(p_{k-o}, \ldots, p_k) > D_{max} \\ r(p_{k-o}, \ldots, p_k) & : \quad d(p_{k-o}, \ldots, p_k) \leq D_{max} \end{cases} . \tag{15}$$

### 7.2 The minimum distortion case

We now consider the minimum distortion case which is stated in Eq. (2). The goal of the proposed algorithm is to find the approximation with the smallest distortion for a given bit budget for encoding its control points. Sometimes this is also called a rate constrained approach. Recall that for class one distortion measures the total distortion is defined as the maximum of the segment distortions. Hence in this section we propose an efficient algorithm which finds the approximation with the smallest maximum distortion for a given bit rate.

We propose an iterative solution to this problem which is based on the fact that we can solve the dual problem stated in Eq. (3) optimally. Consider $D_{max}$ in Eq. (3) to be a variable. We derived in section 7.1 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 proved in [34, 26] that the rate $R^*(D_{max})$ is a non-increasing function of $D_{max}$, which means that $D^1_{max} < D^2_{max}$ implies $R^*(D^1_{max}) \geq R^*(D^2_{max})$.

Since the $R^*(D_{max})$ is a non-increasing function, we can use bisection [35] 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. 15). This implies that there might not exist a $D^*_{max}$ such that $R^*(D^*_{max}) = R_{max}$. In this 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 a $D_{max}$ such that $R^*(D_{max}) = R_{max}$ is not found after a given maximum number of iterations, the algorithm is terminated. In this case, due to the discrete nature of optimization, after a number of iterations further bisecting the range of $D_{max}$ does not lead to corresponding refinements in the vaues of $R^*(D_{max})$, in which case the algorithm terminates. It also terminates as soon as $R_{max}$ is approximated within specified accuracy.

Clearly, the time complexity of this approach for a fixed $D^*_{max}$ is the same as for the DAG shortest path algorithm. The shortest path algorithm is invoked several times by the bisection algorithm to find an $D^*_{max}$ which results in $R^*(D^*_{max}) = R_{max}$. Hence the total time complexity is a linear function of the number of required iterations.

# 8 Distortion measures based on the summation operator

In this section we introduce an algorithm to solve the problems stated in Eqs. (2) and (3) for class two distortion measures, such as the total number of error pels. The presented algorithm is symmetric in the rate and the distortion and hence the same technique can be employed for the minimum distortion case (Eq. (2)) and the minimum rate case (Eq. (3)). 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: $D(p_0, \ldots, p_{N_P-1}) \leftarrow R(p_0, \ldots, p_{N_P-1})$ and $R(p_0, \ldots, p_{N_P-1}) \leftarrow D(p_0, \ldots, p_{N_P-1})$. Again, we first focus on the polygon case and then extend the result to higher order approximations. As pointed out before, we will employ the proposed DAG formulation, to find the optimal approximation. Again, the key to using this formulation is the definition of the weight function.

The 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, in [34, 26] we have proposed a tree-pruning based scheme, which can find all optimal solutions.

## 8.1 Lagrange multiplier approach

In this section we derive a solution to problem (2) which is based on the Lagrange multiplier method [36, 37, 38] and the shortest path algorithm presented in section 7.1. We use the Lagrange multiplier method to relax the constraint so that the unconstrained problem can be solved using the DAG shortest path algorithm.

We first define the Lagrangian cost function

$$J_\lambda(p_0, \ldots, p_{N_P-1}) = D(p_0, \ldots, p_{N_P-1}) + \lambda \cdot R(p_0, \ldots, p_{N_P-1}), \tag{16}$$

where $\lambda$ is the Lagrange multiplier. It has been shown in [36, 37] that if there is a $\lambda^*$ such that,

$$\{p_0^*, \ldots, p_{N_P-1}^*\} = \arg \min_{p_0, \ldots, p_{N_P-1}} J_{\lambda^*}(p_0, \ldots, p_{N_P-1}), \tag{17}$$

and which leads to $R(p_0, \ldots, p_{N_P-1}) = R_{max}$, then $\{p_0^*, \ldots, p_{N_P-1}^*\}$ is also an optimal solution to (2). It is well known that when $\lambda$ sweeps from zero to infinity, the solution to problem (17) traces out the convex hull of the operational rate distortion function, which is a non-increasing function. Hence bisection [35] or the very fast convex search we presented in [34, 39, 40] can be used to find $\lambda^*$. Therefore, if we can find

the optimal solution to the unconstrained problem (17), then we can find the optimal $\lambda^*$ and the convex hull approximation to the constrained problem of Eq. (2).

To be able to employ the above proposed DAG shortest path algorithm, we define the weight function as follows,

$$w(p_{k-1}, p_k) = d(p_{k-1}, p_k) + \lambda \cdot r(p_{k-1}, p_k). \tag{18}$$

Since the shortest path algorithm results in an approximation which minimizes the following sum,

$$\sum_{k=0}^{N_P-1} w(p_{k-1}, p_k), \tag{19}$$

this approximation is the optimal solution to the unconstrained problem of Eq. (17).

The extension of this to higher order schemes is again straightforward. Hence, in general, the following definition of the weight function is proposed

$$w(p_{k-o}, \ldots, p_k) = d(p_{k-o}, \ldots, p_k) + \lambda \cdot r(p_{k-o}, \ldots, p_k). \tag{20}$$

Clearly, the time complexity of the Lagrangian approach for a fixed $\lambda$ is the same as for the DAG shortest path algorithm. The shortest path algorithm is invoked several times by the bisection algorithm to find the optimal $\lambda^*$ and hence the time complexity is a linear function of the number of required iterations.

Now that we have discussed the different approaches for class one and class two distortion measures, it becomes clear that they are strongly related [41]. The class two approaches are based on the Lagrange multiplier method. The Lagrange multiplier method is used to transform the constrained optimization problem into a set of unconstrained optimization problems parameterized by the Lagrangian multiplier $\lambda$. These unconstrained problems are then solved optimally using a shortest path algorithm. The optimal $\lambda^*$, which results in the solution of the original constrained problem, is then found using an iterative approach, such as bisection, where for each iteration the unconstrained problem needs to be solved. For the class two approaches, both the minimum rate and the minimum distortion problems are solved by the same algorithm. This is one of the main differences between the class two and the class one approaches. For the class one approaches, the minimum rate problem, which is a constrained optimization problem, can be transformed into an unconstrained problem using an elegant definition of the edge weights. Then this unconstrained problem can be solved directly using the same shortest path algorithm as above. In other words, no iteration is necessary to solve the minimum rate problem. The minimum distortion problem is

then solved using the fact that the we can find the optimal solution to the minimum rate problem, which results in a non-increasing operational rate distortion function. The solution to the minimum rate problem is also found by an iterative search for the optimal $D^*_{max}$ using bisection. For each iteration, the minimum rate problem (i.e., the unconstrained problem) is solved using the shortest path algorithm.

# 9 Experimental results

In this section we present coding results for a polygonal and a second order B-spline based implementations of the presented algorithms. We use the first 100 frames of the MPEG-4 test sequence "Kids". All the results to be shown represent averages over these 100 frames. The sequence is in SIF format ($352 \times 240$ pels) and the binary $\alpha$ plane is provided, i.e., no segmentation step is necessary. Since the $\alpha$ plane represents the video objects as a bit plane, we first extract the boundaries from the $\alpha$ plane. We define the boundary as the outermost pels of an object. Note that one could define the boundary as the line between the background pels and the outermost object pels, and the proposed algorithms would still work. So far we have only defined the constraints a particular control point encoding scheme needs to satisfy. For the reported results, we used two simple first and second order DPCM schemes for polygonal and B-spline approximations, respectively. Consider Fig. 16, which shows the situation at hand, for the B-spline case. The segment rate $r(p_{k-2}, p_{k-1}, p_k)$, which is required to encode control point $p_k$, given the control points $p_{k-2}$ and $p_{k-3}$, consists of 2 bits for encoding the angle $\alpha$ (+45,-45,+90,-90) and 2 to 5 bits for encoding the length $\beta$ $(1, \ldots, 15)$ [34]. For the first order prediction, the absolute angle is used instead, requiring 3 bits (0,+45,-45,+90,-90,+135,-135,180). Furthermore we need to encode the starting point for each boundary, which in SIF requires 17 bits (9+8 bits for the two dimensions). Note that these bits are not included in the following rate distortion plots. In the experiments presented below, we fixed the width of the admissible control point set to 1.5 pels, which results in the fact that all eight neighboring pels of a boundary point are included in the set.

In the first experiment, we use a distortion measure based on the maximum operator. We define the segment distortion using the band concept introduced earlier. Hence we draw a band of distance $D_{max}$ around the original boundary and each segment which leaves this band has a segment distortion of infinity assigned to it. Figure 17 demonstrates this concept for $D_{max} = 3$ pels and shows two optimal approximations (polygonal and B-spline) within the distortion band. This figure also shows the effect of upper-limiting the run lengths in the VLC table used for encoding control points. Since in the implemen-

tation only runs up to 15 are allowed, with some run lengths smaller than 15 also excluded, some long straight line approximations were composed of consecutive runs of 15 along the same direction. We then use the above proposed DAG algorithm to find the set of control points which results in the smallest bit rate for this given maximum distortion $D_{max}$. We do this for different $D_{max}$ = 0.8, 1, 1.5, 2, 2.5 and 3 pels. The resulting averaged operational rate distortion curves are displayed in Fig. 18 for polygonal and B-spline approximations. For comparison purposes, we also show in Fig. 18 the averaged operational rate distortion curve for the case when the set of admissible control points $A$ is equal to the set of the original boundary points $B$ ($\Box$ marks). Clearly choosing a wider admissible control point band leads to better performance in the operational rate distortion sense. Note that we find the optimal polygonal and second order B-spline approximations such that the rate required to encode the control points is minimized for a given maximum distortion $D_{max}$. While the employed maximum distortion measure is a very useful objective criterion, we also display a single $\alpha$ plane for visual inspection. Figure 19 shows the original $\alpha$ plane for the 17-th frame and Fig. 20 shows the same plane after encoding and decoding, using a $D_{max}$ of 0.8. The required bit rate is 858 bits and the resulting $d_n$ is 0.0257.

In the second experiment, we use a distortion measure which is based on the summation operator. To create the resulting averaged operational rate distortion curves, shown in Fig. 21, we swept the Lagrangian multiplier $\lambda$ from zero to infinity in discrete steps. For each $\lambda$ we encoded each frame in the sequence and each boundary in each frame. We then recorded the resulting average rate and distortion (the average is with respect to the 100 frames). Therefore the resulting bit allocation is optimal among all objects and all frames. Figure 22 shows the resulting polygonal and B-spline approximations when this distortion measure was employed.

Finally in Fig. 23 we combine all the results from the previous two experiments, using the $d_n$ distortion metric. The reason for doing this is that $d_n$ is the metric used by MPEG-4 for evaluating the various shape encoding techniques. As expected, each minimum summation technique outperforms its corresponding minimum maximum one. In addition, in Fig. 23, a result is shown (line with $\diamond$) obtained using a different VLC table than the one used in the rest of the curves. This was done to point to the potential benefits of using more efficient VLCs.

## 10 Summary and Future Directions

In this paper we proposed a framework for the rate distortion operationally optimal encoding of shape information in the intra mode. We have shown that each curve approximation has a natural order. If the control point encoding scheme is matched to this order and the distortion is carefully defined, then the optimal approximation can be found using a directed acyclic graph (DAG) shortest path algorithm. We have shown that the minimum maximum distortion optimization problem and the minimum total (average) distortion optimization problem can both be solved by similar means, using an appropriate definition of the DAG weight function.

The proposed framework is very powerful, in that it can be extended to solve various formulations of the shape coding problem also optimally. For example, in [26] we have shown in detail that the jointly optimal encoding of several shapes is a direct extension of the work presented here. In fact we have used this optimal encoding of several shapes in the second experiment reported above (minimum average distortion). While we have shown fixed order optimization in this paper, a meaningful extension of this work is in the direction of mixed order approximations. Additional work has been done in including the so called lexicographic [42] optimality criterion in the minimum maximum distortion optimization. This can be achieved easily in the proposed framework. The extension of the methodology used here to find the operationally optimal shape approximations in the intra mode to the inter mode is a promising direction of future work and is currently under investigation. In [27] we have extended the shape encoding scheme to include the segmentation algorithm. In other words, we jointly find and encode optimally the shape of a given object. Figure 24 illustrates how a variable-width admissible control point band is used for that purpose. While the proposed algorithms are already able to outperform any other lossy intra-mode shape encoding scheme in the rate distortion sense, these schemes can still improve significantly. We are currently focusing on the control point encoding scheme, which is a very simple DPCM scheme. We expect that other optimized schemes will result in further performance gains [43].

## References

[1] M. Kunt, "Second-generation image coding techniques," *Proceedings of the IEEE*, vol. 73, pp. 549–574, Apr. 1985.

[2] 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.

[3] L. Chiariglione, "MPEG and multimedia communications," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, pp. 5–18, Feb 1997.

[4] T. Sikora, "The MPEG-4 video standard verification model," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, pp. 19–31, Feb 1997.

[5] J. Ostermann and A. Puri, "Natural and synthetic video in MPEG-4," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, May 1998.

[6] R. Schäfer and T. Sikora, "Digital video coding standards and their role in video communications," *Proceedings of the IEEE*, vol. 38, pp. 907–924, June 1995.

[7] A. K. Katsaggelos, L. P. Kondi, F. W. Meier, J. Ostermann, and G. M. Schuster, "MPEG-4 and rate-distortion-based shape-coding techniques," *Proceedings of the IEEE,* special issue on *Multimedia Signal Processing*, vol. 86, no. 6, pp. 1126–1154, June, 1998.

[8] ITU-T, Telecommunication standardization sector of ITU, *T.4, Standardization of group 3 facsimile terminals for document transmission.*

[9] ITU-T, Telecommunication standardization sector of ITU, *T.82, Information technology, coded representation of picture and audio information, progressive bi-level image compression.*

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

[11] 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.

[12] 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.

[13] 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.

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

[15] 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.

[16] M. Eden and M. Kocher, "On the performance of contour coding algorithm in the context of image coding. part i: Contour segment coding," *Signal Processing*, vol. 8, pp. 381–386, July 1985.

[17] M. Hötter, "Optimization and efficiency of an object-oriented analysis-synthesis coder," *IEEE Transactions on Circuits and Systems*, vol. 4, pp. 181–194, Apr. 1993.

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

[19] R. Bellman, "On the approximation of curves by line segments using dynamic programming," *Communications of the ACM*, vol. 4, p. 284, Apr. 1961.

[20] V. Bhaskaran, B. K. Natarjan, and K. Konstantinides, "Optimal piecewise-linear compression of images," in *Proceedings Data Compression Conference* (J. A. Storer and M. Cohn, eds.), 1993.

[21] R. A. Fowell and D. D. McNeil, "Faster plots by fan data-compression," *IEEE Computer Graphics and Applications*, vol. 9, no. 2, pp. 58–66, 1989.

[22] J. Dunham, "Optimum uniform piecewise linear approximation of curves," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, pp. 67–75, Aug. 1986.

[23] D. Saupe, "Optimal piecewise linear image coding," in *Proceedings of the Conference on Visual Communications and Image Processing*, vol. 3309, pp. 747–760, SPIE, 1997.

[24] A. K. Jain, *Fundamentals of digital image processing.* Prentice-Hall, 1989.

[25] R. L. Lagendijk, J. Biemond, and C. P. Quist, "Low bit rate video coding for mobile multimedia communications," *Proceedings of EUSIPCO*, vol. 1, pp. 435–438, 1996.

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

[27] L. P. Kondi, F. W. Meier, G. M. Schuster, and A. K. Katsaggelos, "Joint optimal object shape estimation and encoding," in *Proceedings of the Conference on Visual Communications and Image Processing*, (San Jose, California), pp. 14–25, Jan. 1998.

[28] D. Haugland, J. G. Heber, and J. H. Husoy, "Optimization algorithms for ECG data compression," *Medical and Biomedical Engineering and Computing*, vol. 35, pp. 420–424, 1997.

[29] R. Nygaard and D. Haugland, "Compressing ECG signals by polynomial approximation," *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, June 1998.

[30] Foley, vanDam, Feiner, and Hughes, *Computer graphics: Principles and practice.* Addison-Wesley, 1990.

[31] G. Melnikov, P. V. Karunaratne, G. M. Schuster, and A. K. Katsaggelos, "Rate-distortion optimal boundary encoding using an area distortion measure," *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, June 1998.

[32] F. W. Meier, "An efficient shape coding scheme using B-splines which is optimal in the rate-distortion sense," Master's thesis, Northwestern University, Evanston, Illinois, USA, June 1997. Department of Electrical and Computer Engineering.

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

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

[35] C. F. Gerald and P. O. Wheatley, *Applied numerical analysis.* Addison Wesley, fourth ed., 1990.

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

[37] Y. Shoham and A. Gersho, "Efficient bit allocation for an arbitrary set of quantizers," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 36, pp. 1445–1453, Sept. 1988.

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

[39] G. M. Schuster and A. K. Katsaggelos, "An optimal quad-tree-based motion estimation and motion compensated interpolation scheme for video compression," *IEEE Transactions on Image Processing*, to appear.

[40] G. M. Schuster and A. K. Katsaggelos, "Fast and efficient mode and quantizer selection in the rate distortion sense for H.263," in *Proceedings of the Conference on Visual Communications and Image Processing*, pp. 784–795, SPIE, Mar. 1996.

[41] G. M. Schuster and A. K. Katsaggelos, "The minimum-average and minimum-maximum criteria in lossy compression," *Vistas in Astronomy*, to appear.

[42] D. T. Hoang, E. L. Linzer, and J. S. Vitter, "Lexicographic bit allocation for MPEG video," *Journal of visual communication and image representation*, vol. 8, Dec. 1997.

[43] G. Melnikov, G. M. Schuster, and A. K. Katsaggelos, "Simultaneous optimal boundary encoding and variable-length code selection," *Proceedings of the International Conference on Image Processing*, to appear 1998.

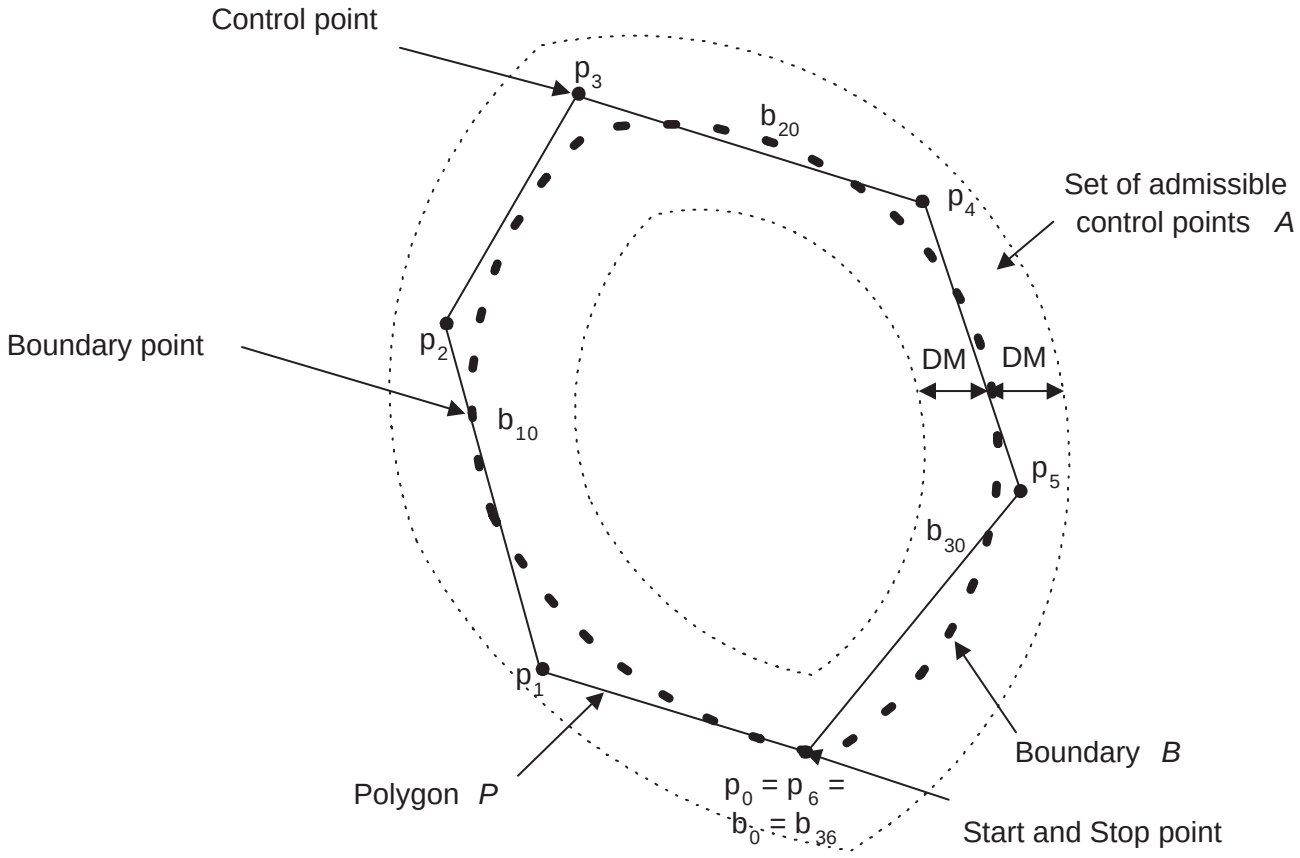


Figure 1: Overview of the proposed approximation of an original boundary B by a polygon P.

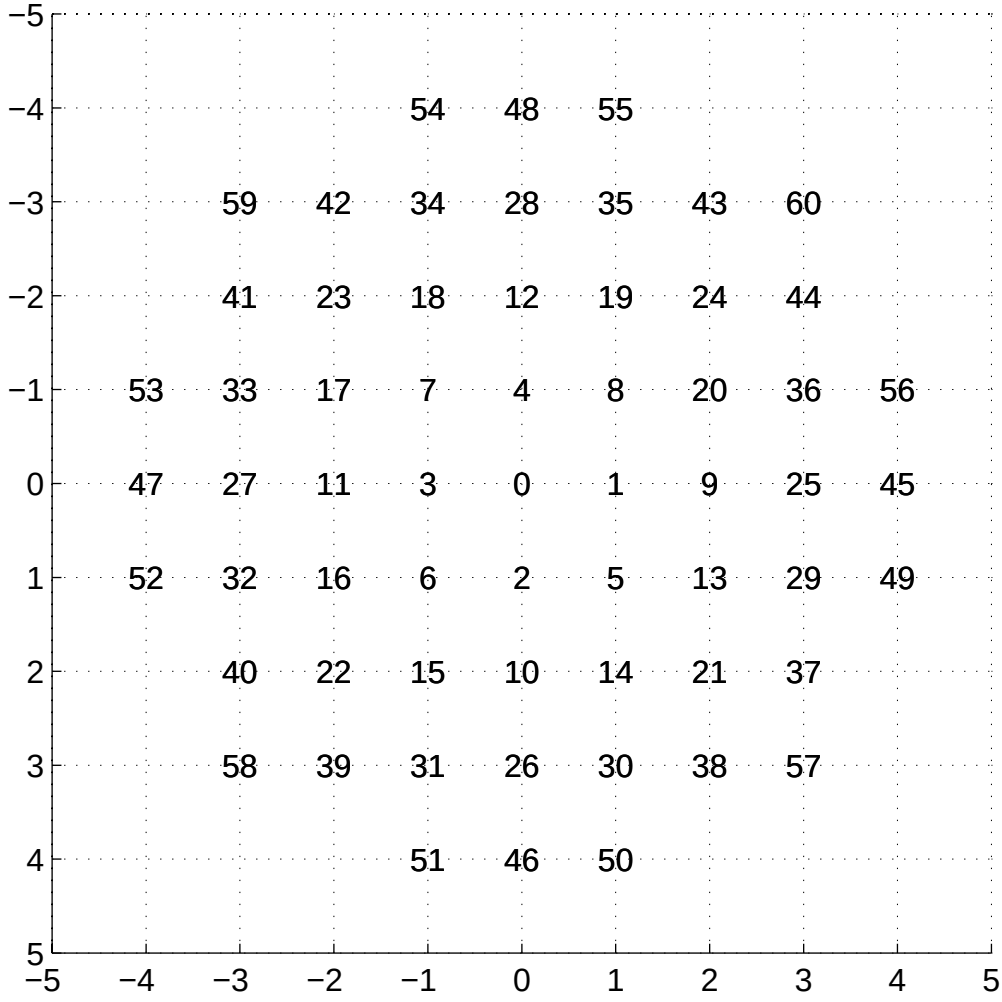


Figure 2: Vector increments for the definition and ordering of the admissible control point set. Note that vector $v_j$ starts at 0 and ends at $j$.

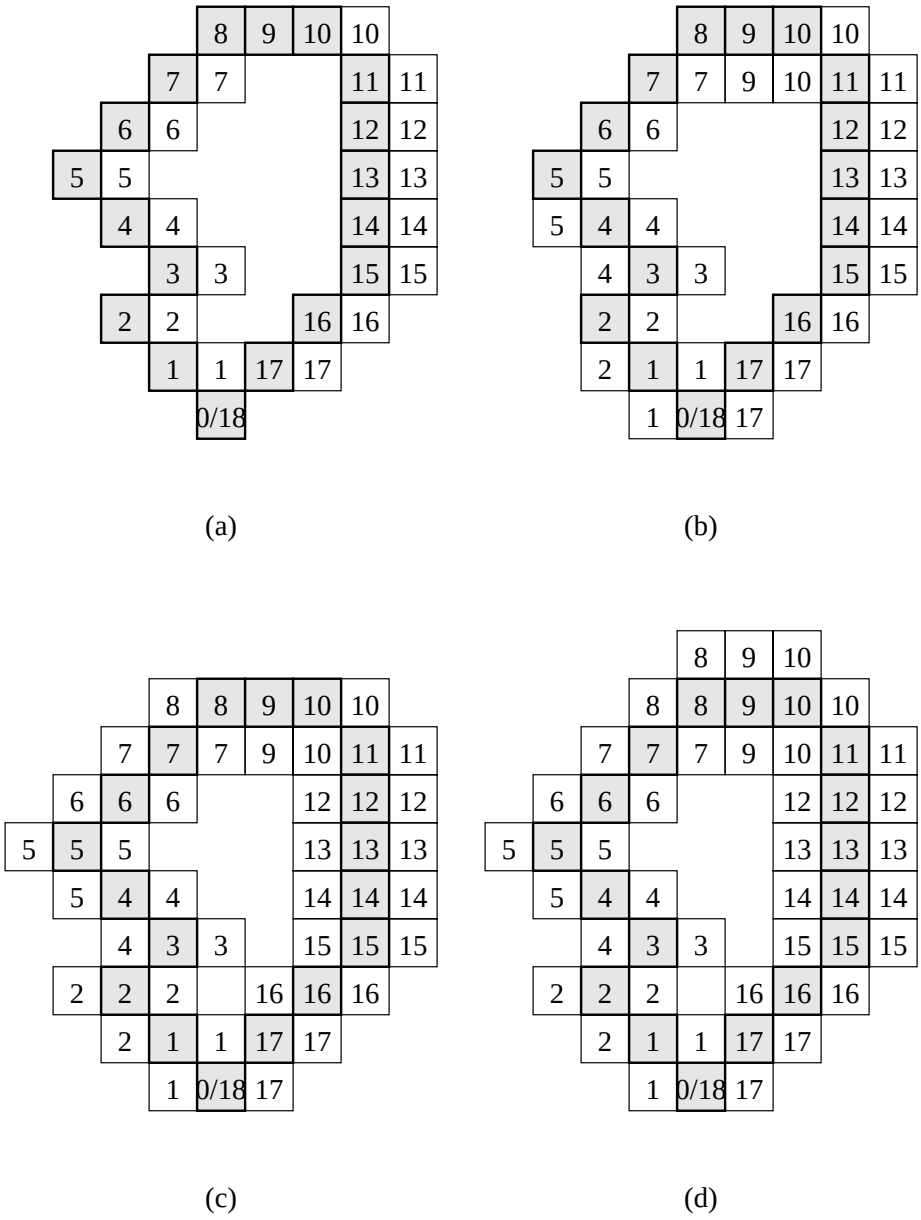


Figure 3: Figures a, b, c, and d show the final result after the inner loop has been finished for $v_1$, $v_2$, $v_3$ and $v_4$, for a maximum distance $DM$ of one pel. The pels with a bold border indicate the original boundary and the other pels are the admissible control points, where the number of the associated boundary point is written inside it.

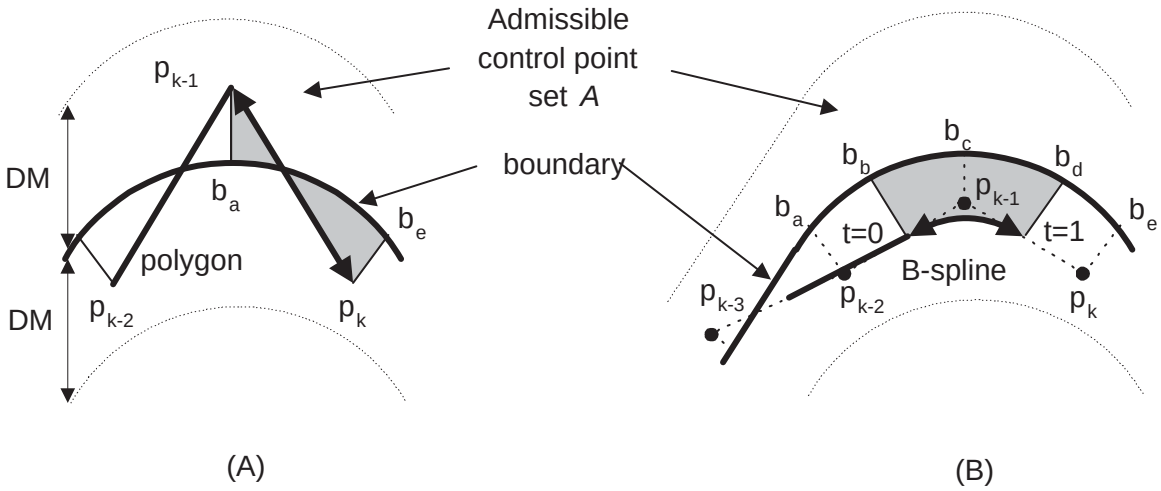


Figure 4: The definition of a state for a polygon and a second order B-spline curve. Figure (a) shows that for a polygon, knowing $p_{k-1}$ makes the search for future control points independent from the control points already selected. Figure (b) shows that for a second order B-spline, knowing $p_{k-2}$ and $p_{k-1}$ makes the search for future control points independent from the control points already selected.

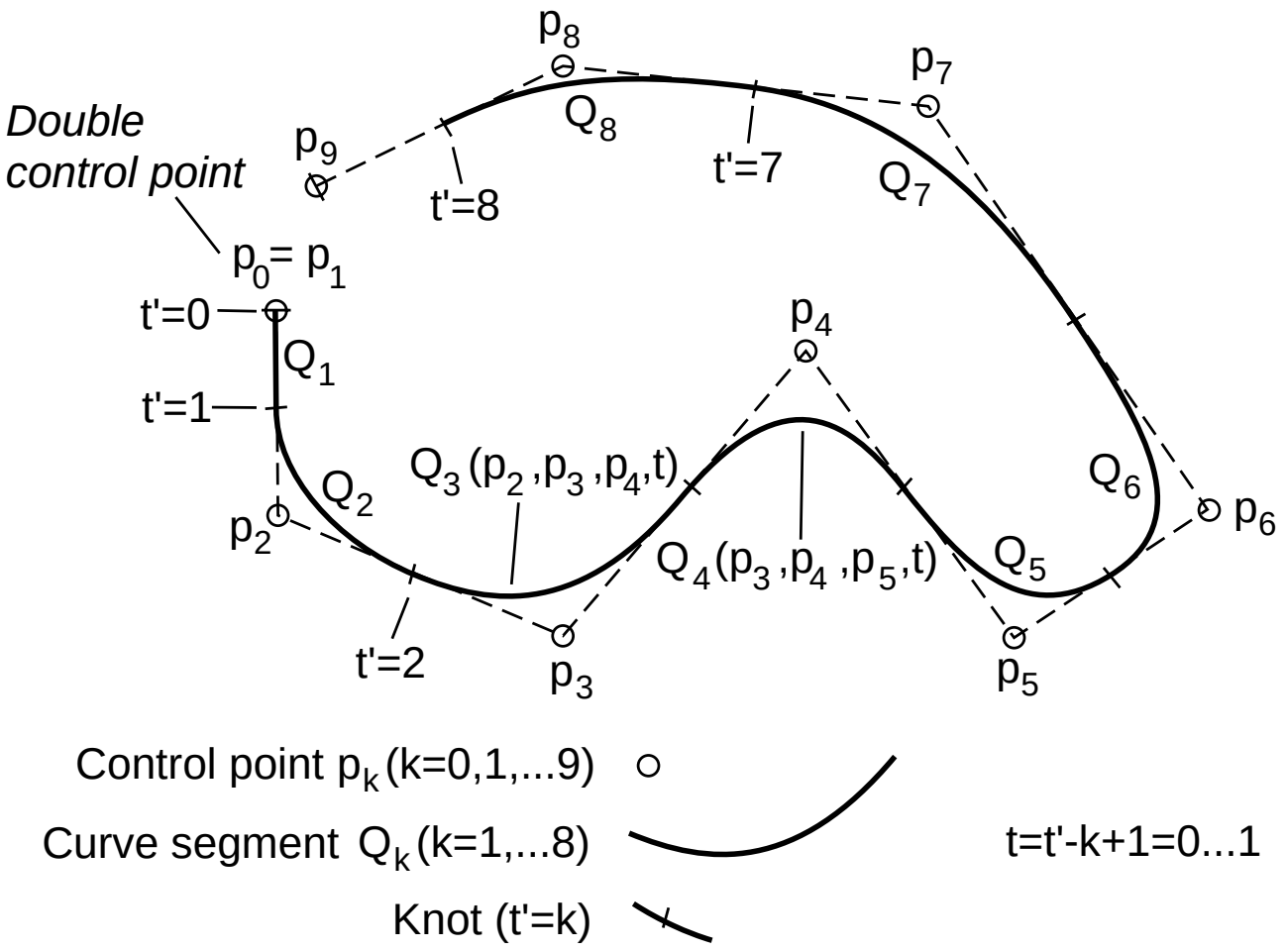


Figure 5: A second degree B-spline curve with 8 curve segments $Q_k$ and a double control point at the beginning of the curve.

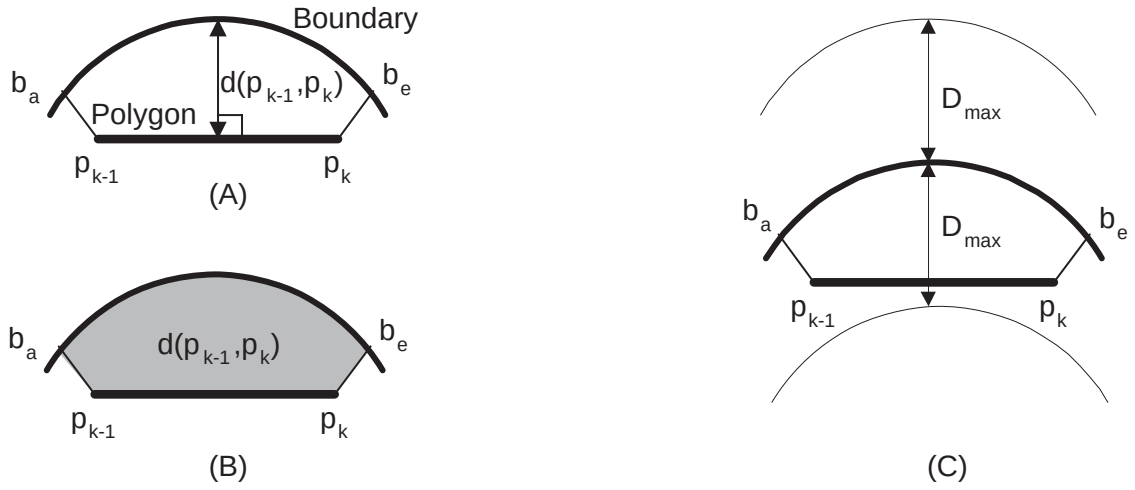


Figure 6: Three different ways for defining the segment distortion for a polygon-based approximation of the original boundary. Figure (a) shows the maximum distance, Figure (b) the error area, and Figure (c) the maximum distance band concept.

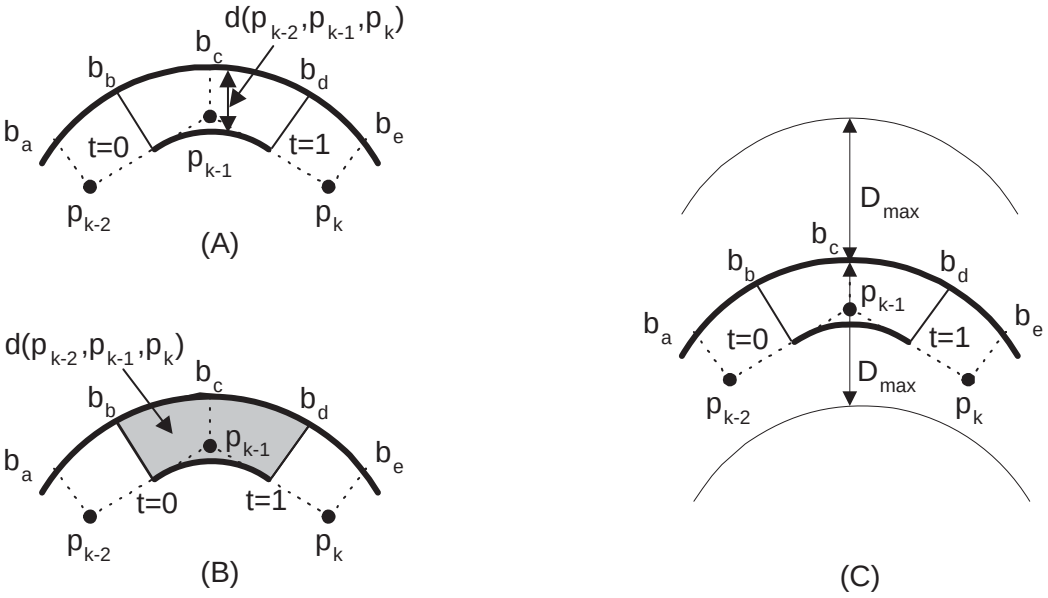


Figure 7: Three different ways for defining the segment distortion for a second order B-spline curve-based approximation of the original boundary. Figure (a) shows the maximum distance, Figure (b) the error area, and Figure (c) the maximum distance band concept.

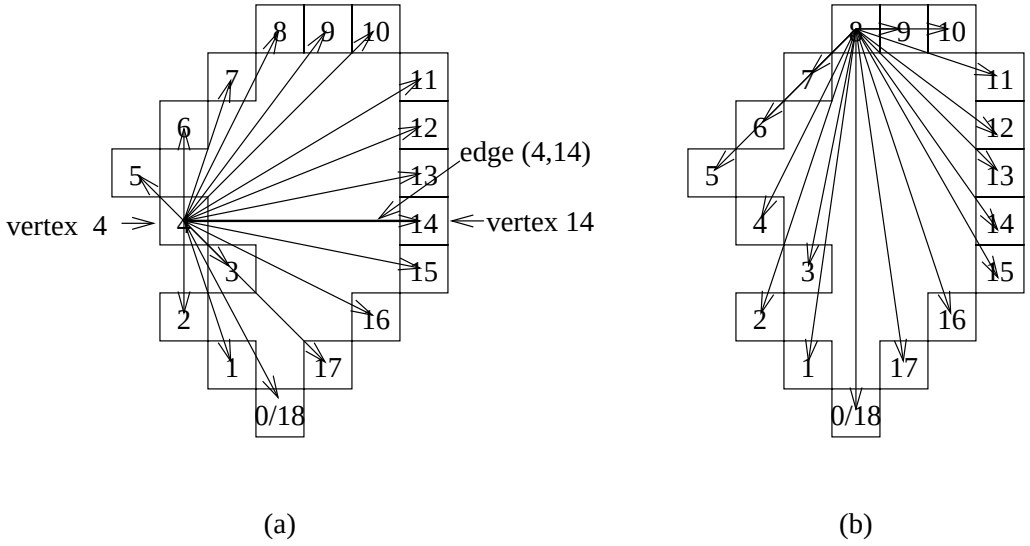


Figure 8: Interpretation of the boundary and the polygon approximation as a fully connected weighted directed graph. Note that the set of all edges $E$ equals $\{(a_i, a_j) \in A^2 : i \neq j\}$. Two representative subsets are displayed: (a) $\{(a_4, a_j) \in A^2 : \forall j \neq 4\}$ and (b) $\{(a_8, a_j) \in A^2 : \forall j \neq 8\}$.

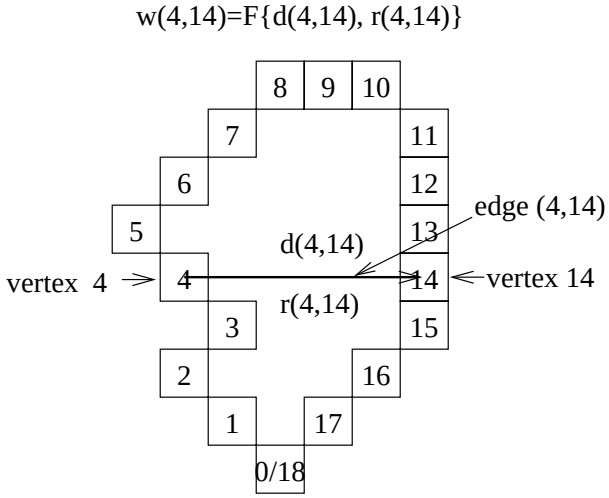


Figure 9: Each edge has a segment rate and a segment distortion associated with it. The edge weight is then a function (F$\{\cdot\}$) of these rates and distortion. For class one distortion measures, this function is different than for class two distortion measures.

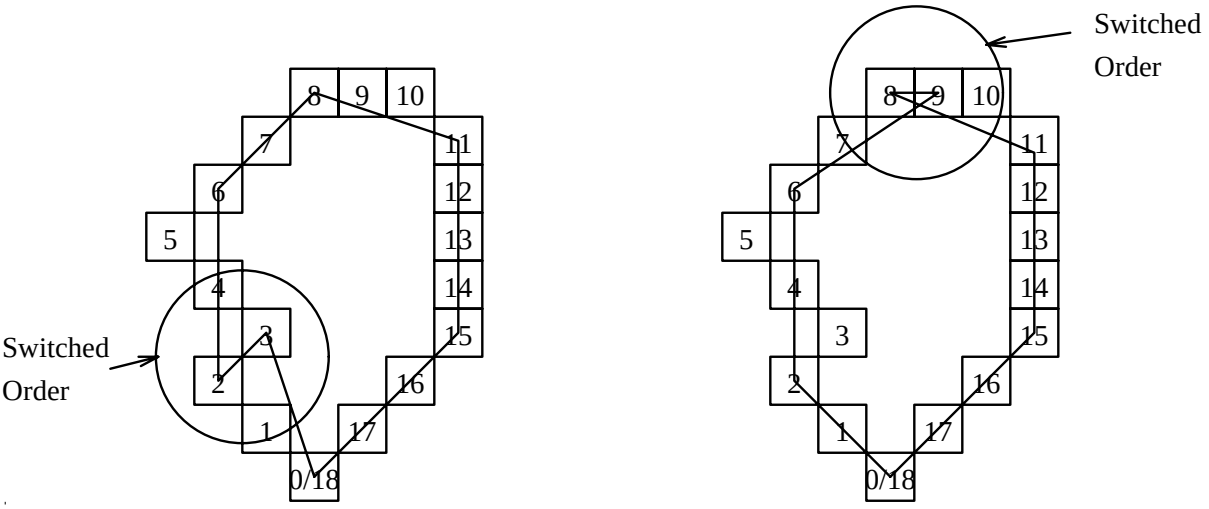


Figure 10: Examples of polygons with rapid changes in direction.

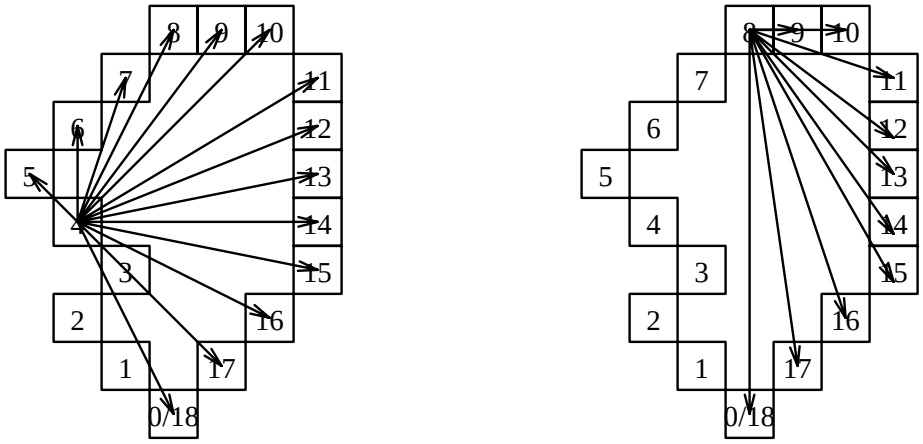


Figure 11: Interpretation of the boundary and the polygon approximation as a weighted directed acyclic graph. Note that the set of all edges $E$ equals $\{(a_i, a_j) \in A^2 : i < j\}$. Two representative subsets are displayed: (a) $\{(a_4, a_j) \in A^2 : \forall j > 4\}$ and (b) $\{(a_8, a_j) \in A^2 : \forall j > 8\}$.

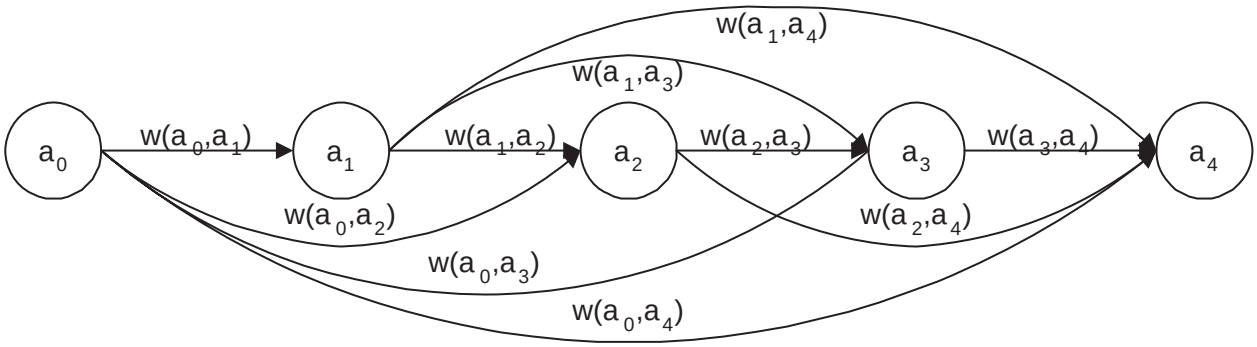


Figure 12: A simple DAG for a polygon approximation where $A = B$. The weights are defined as a function of the segment rate and the segment distortion.

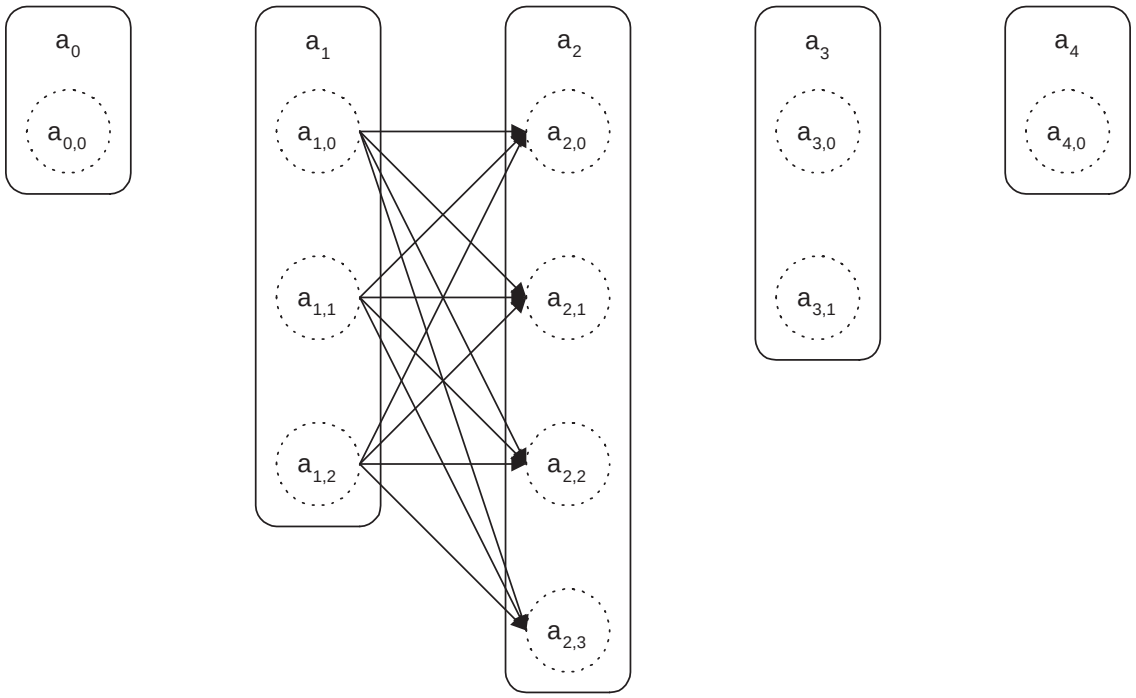


Figure 13: A simple DAG for a polygon approximation, where $A \supseteq B$. For this example we selected $A = \{a_{0,0}, a_{1,0}, a_{1,1}, a_{1,2}, a_{2,0}, a_{2,1}, a_{2,2}, a_{2,3}, a_{3,0}, a_{3,1}, a_{4,0}\}$. Furthermore, we only show the edges originating in $a_{1,i}$ and terminating in $a_{2,j}$, to keep the graph readable. Note that the edge $(a_1, a_2)$ (for $A = B$) is replaced by the shown set of edges $(a_{1,i}, a_{2,j}), \forall i, j$. To fill the entire graph with edges, this replacement is done for each edge in the original graph.

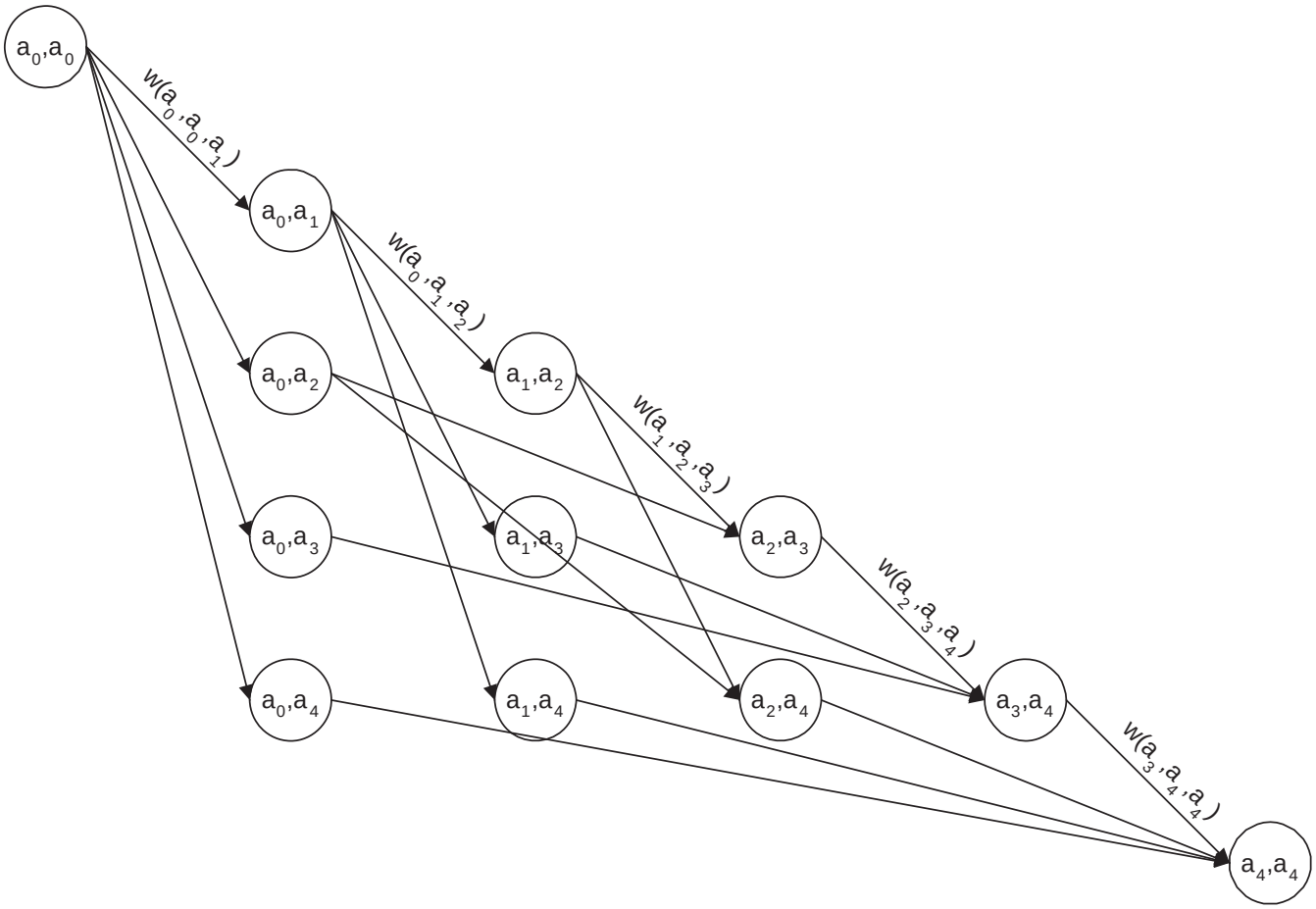


Figure 14: A simple DAG for a second order B-spline approximation, where $A = B$. We only label the five top edges with their respective weight function, but clearly, each edge has a weight function associated.

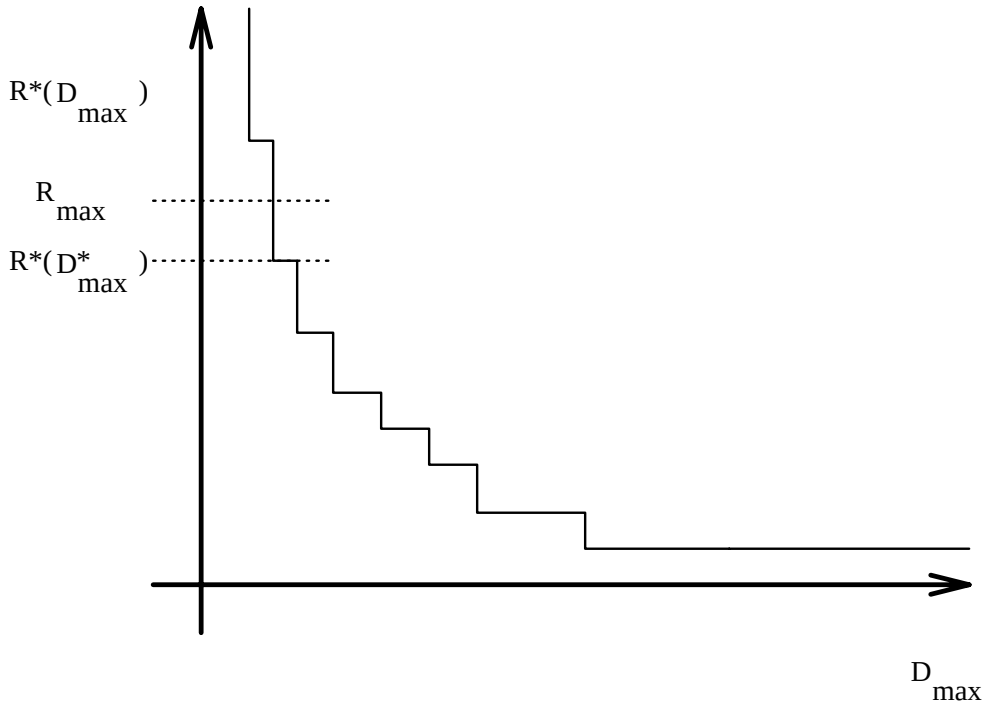


Figure 15: The $R^*(D_{max})$ function, which is a non-increasing function exhibiting a staircase characteristic. The selected $R_{max}$ falls onto a discontinuity and therefore the optimal solution is of the form $R^*(D^*_{max}) < R_{max}$, instead of $R^*(D^*_{max}) = R_{max}$.

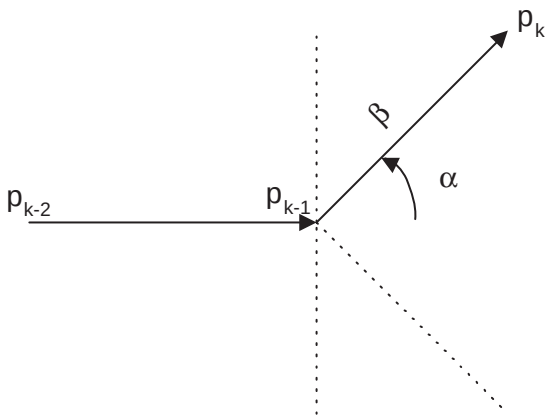


Figure 16: A simple second order DPCM scheme is used to encode the control points. The angle $\alpha$ must take a value out of the following set: $\{-90, -45, 45, 90\}$. This is encoded with two bits. The length $\beta$ is a number between 1 and 15 and is encoded using between 2 and 5 bits.

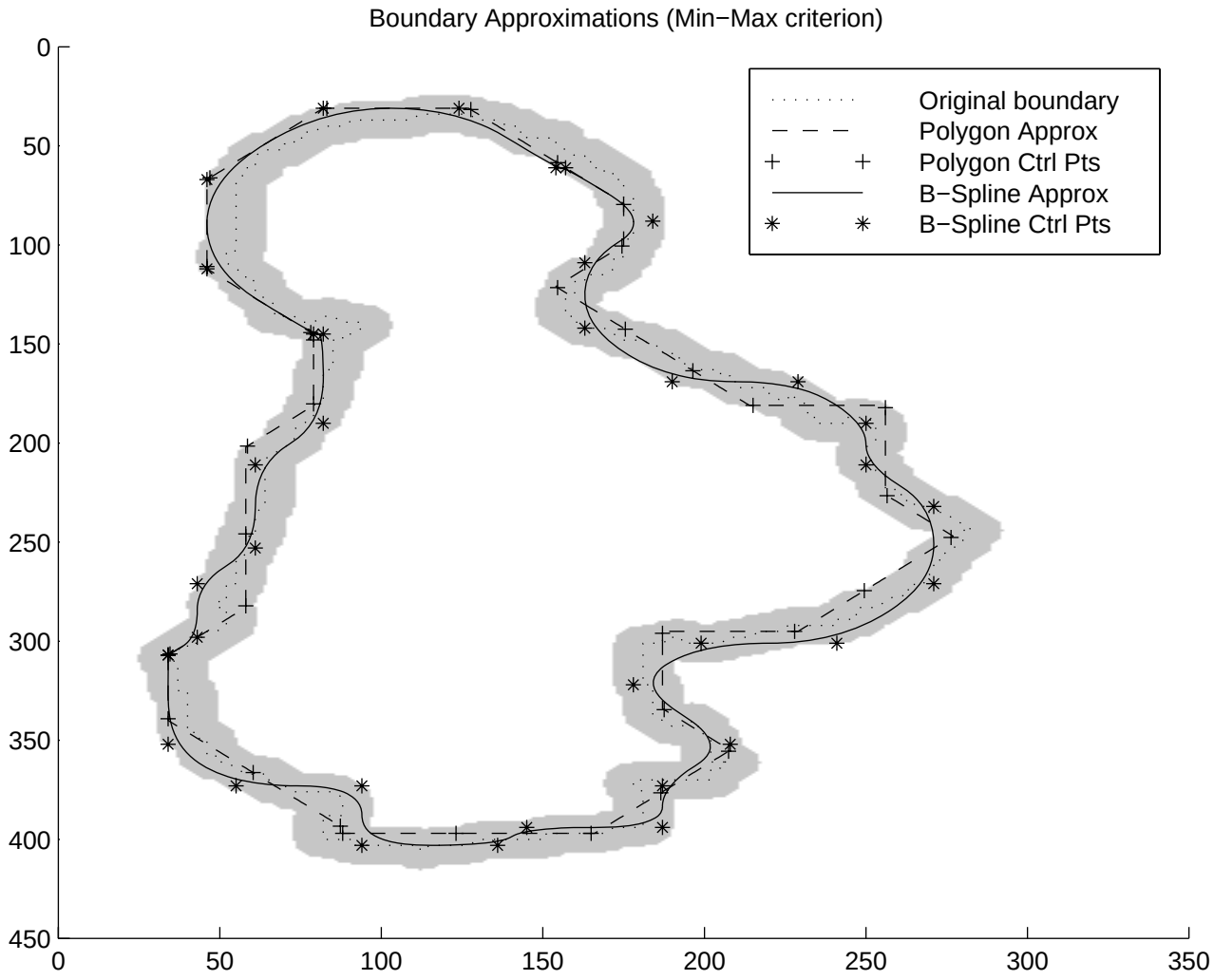


Figure 17: Kid 2 in frame 17. Encoding using the maximum distortion measure. The distortion band has $D_{max} = 3$.

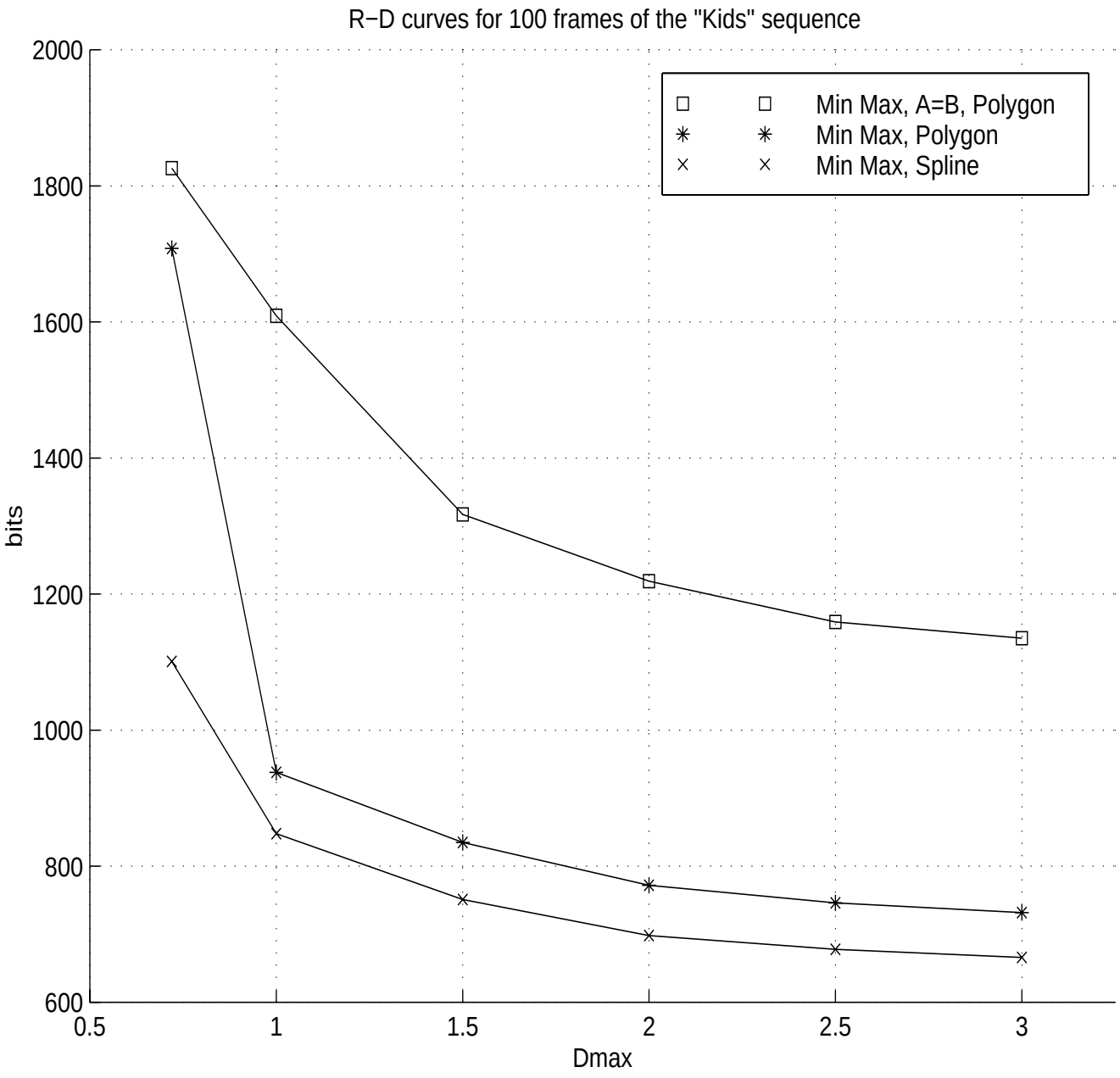


Figure 18: Average operational rate distortion curves for the first 100 frames of the "Kids" sequence.

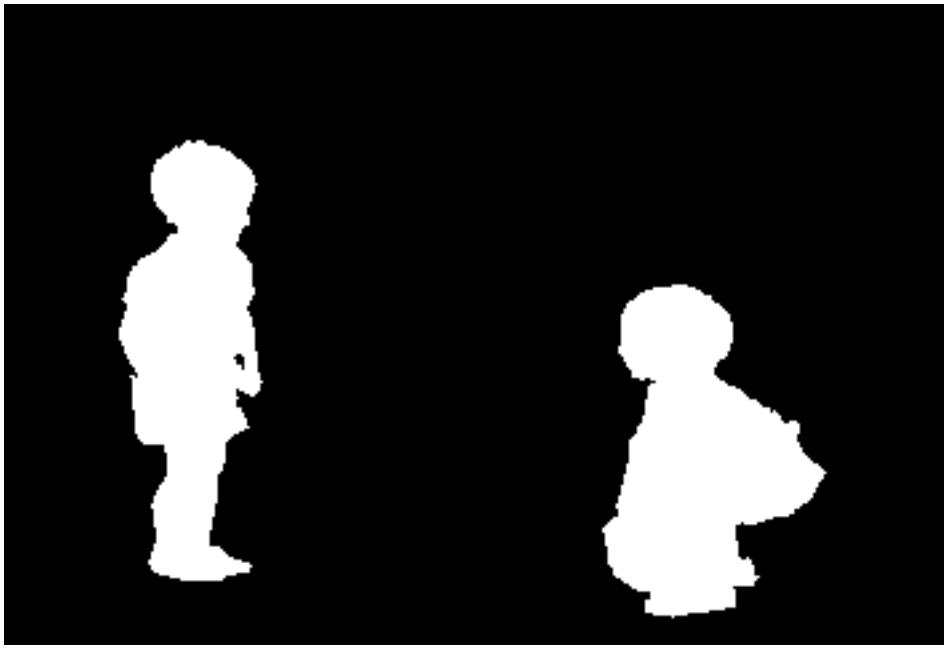

Figure 19: The original $\alpha$ plane for the 17-th frame in the sequence.

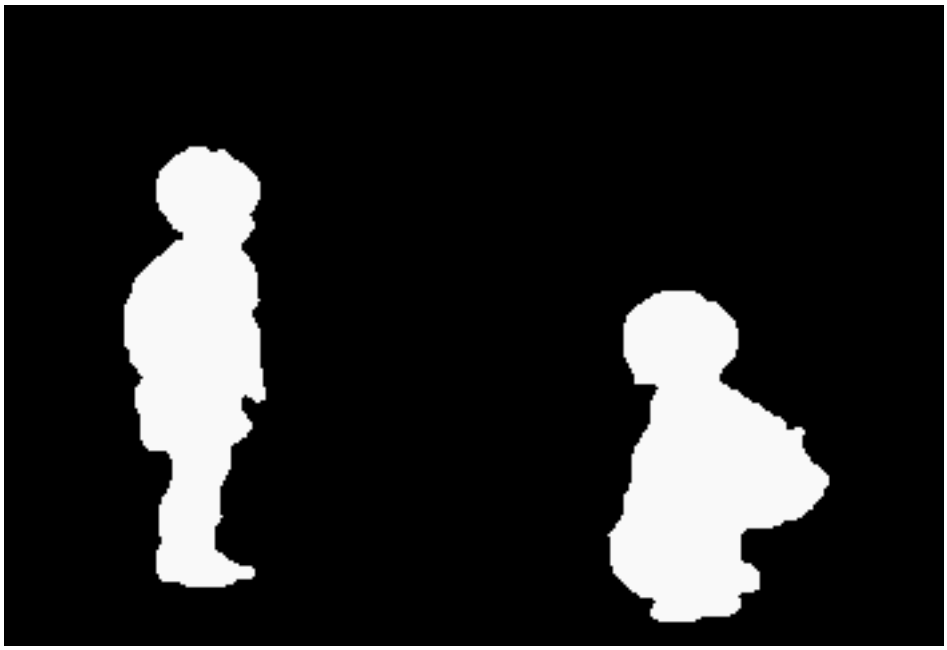

Figure 20: The encoded $\alpha$ plane for the 17-th frame in the sequence using $D_{max} = 0.8$. The algorithm required 858 bits resulting in a $d_n$ of 0.0257.

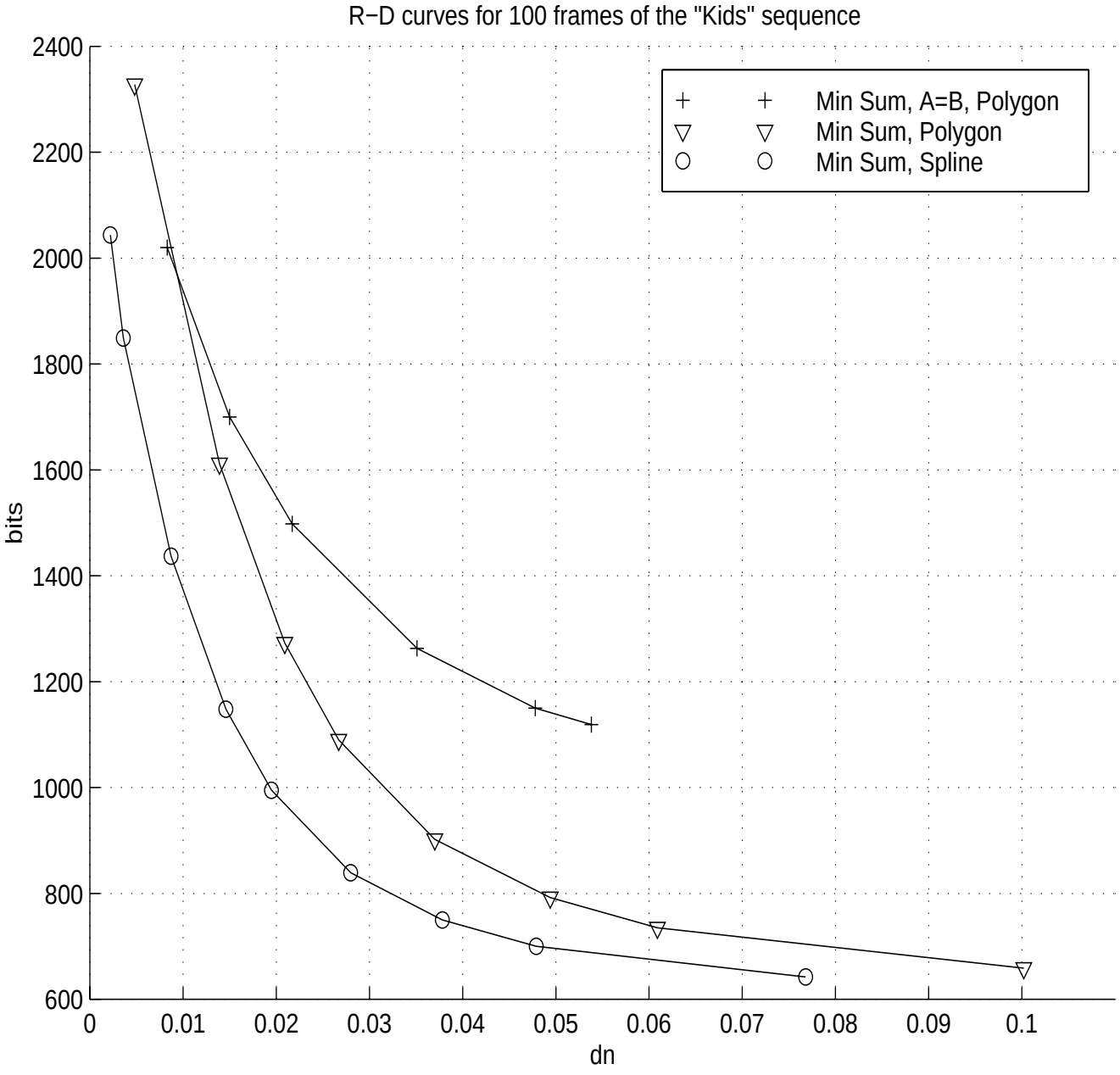


Figure 21: Average operational rate distortion curves for the first 100 frames of the "Kids" sequence.

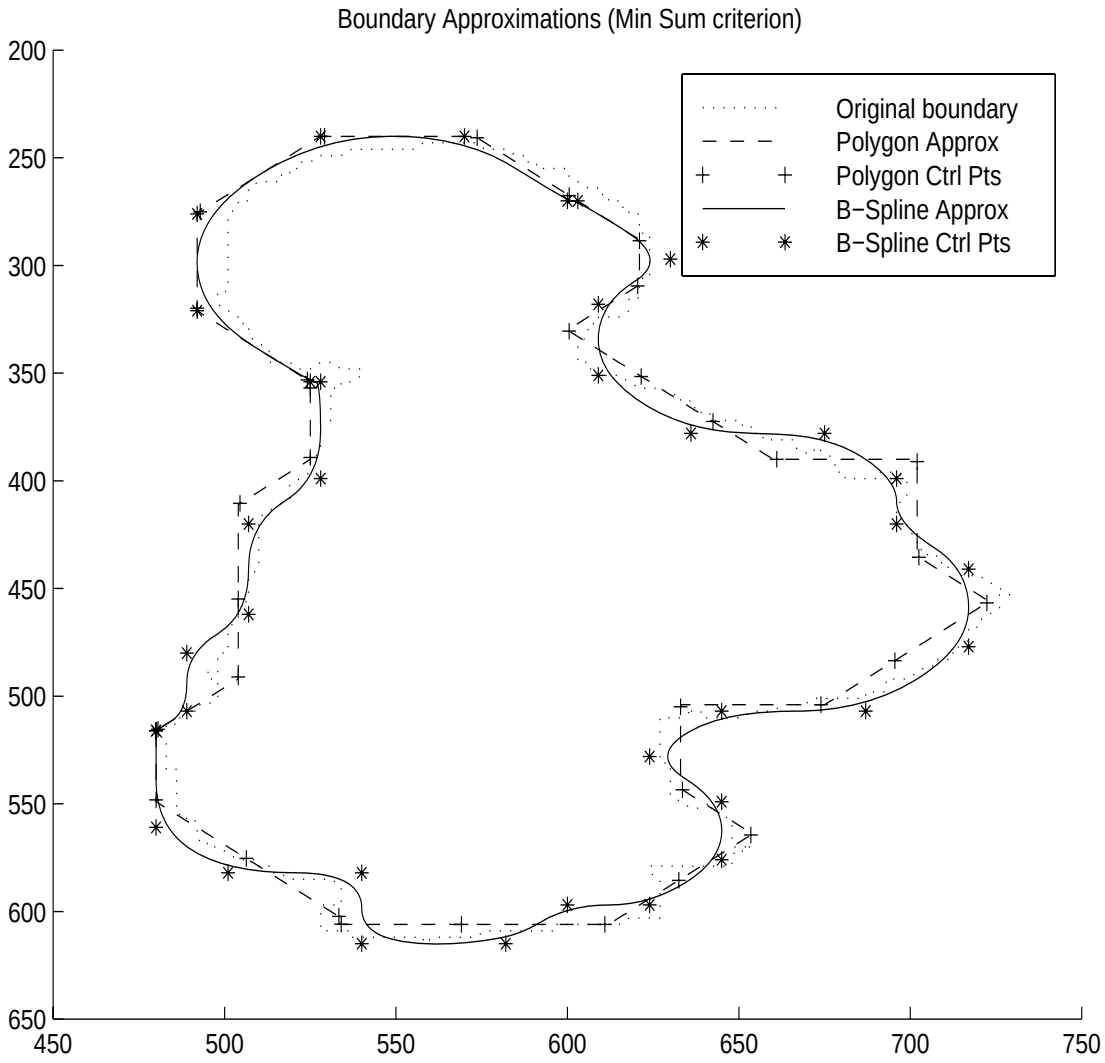


Figure 22: Kid 2 in frame 17. Encoding using the additive distortion measure.

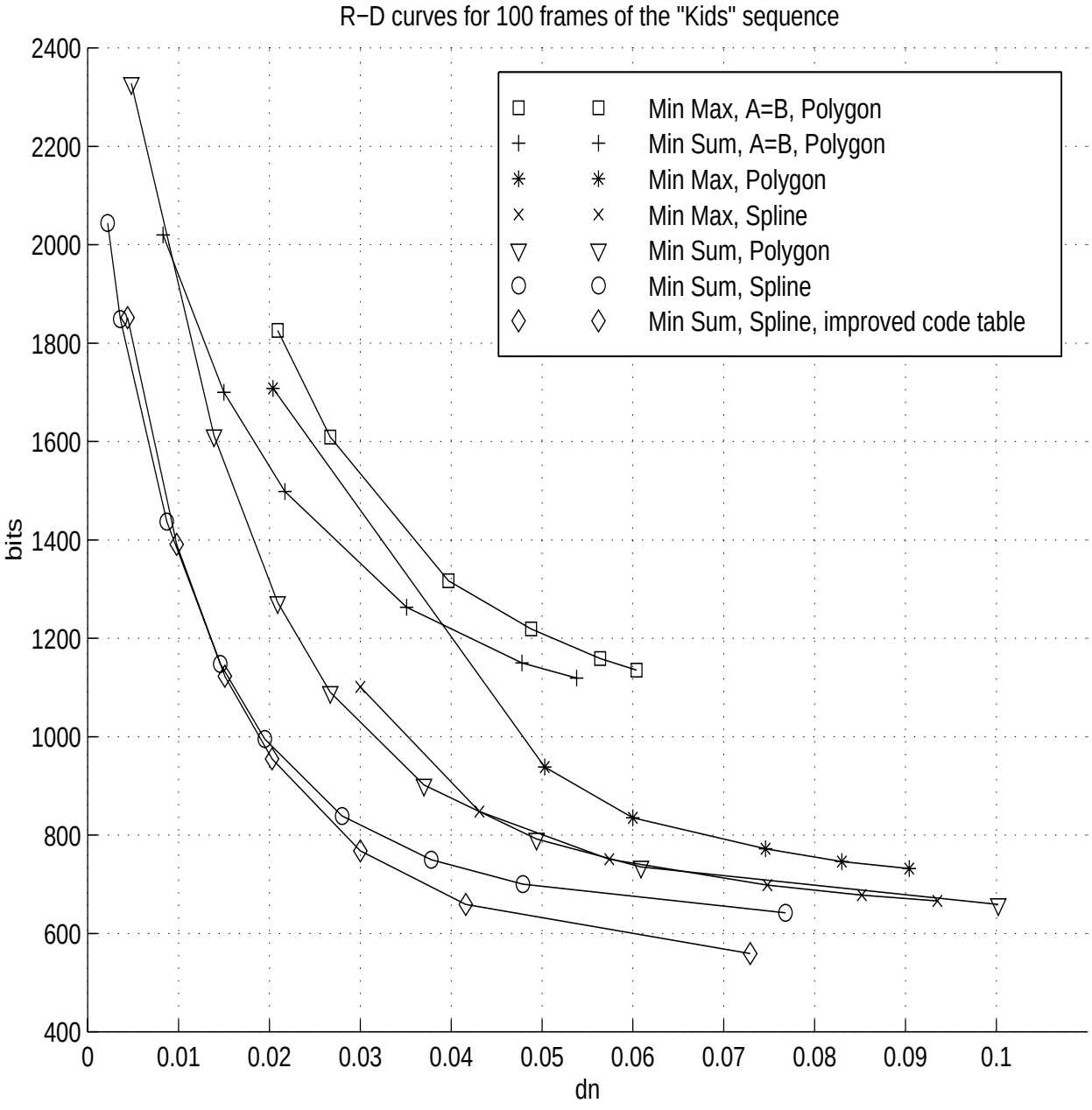


Figure 23: Average operational rate distortion curves for the first 100 frames of the "Kids" sequence.

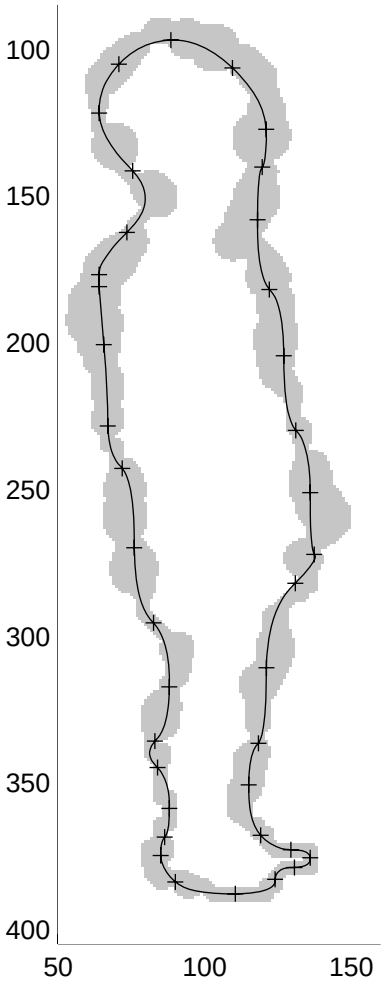


Figure 24: The width of the control point band is inversely related to the confidence in the location of the boundary pixels [27].