INTERNATIONAL ORGANIZATION FOR STANDARDIZATION
ORGANISATION INTERNATIONALE DE NORMALISATION
ISO/IEC JTC1/SC29/WG11
CODING OF MOVING PICTURES AND ASSOCIATED AUDIO INFORMATION

ISO/IEC JTC1/SC29/WG11
MPEG 95/360
November 1995

Source: **RACE/MAVT, University of Hannover, Robert Bosch GmbH, and Deutsche Telekom AG**

Authors: **Peter Gerken, Michael Wollborn, Stefan Schultz (University of Hannover)**

Title: **Polygon/spline approximation of arbitrary image region shapes as proposal for MPEG–4 tool evaluation – Technical description**

Abstract

Polygon/spline approximation of arbitrary image region shapes is submitted for tool evaluation. First, polygon approximation is performed. Starting with an initial 4–vertices polygon which represents the farest elongations into different directions, the polygon approximation is iteratively refined taking into account the local requirements until a given tolerance criterion is satisfied everywhere. Then, spline approximation is calculated using the polygon vertices. Whenever spline approximation also satisfies the tolerance criterion for a certain part of the shape, spline approximation substitutes polygon approximation for that part thus giving the approximated shape a more natural look. By this combined approach, the advantages of both techniques are utilized and their disadvantages avoided. Thus, the combined approach is efficient in terms of reduction of the number of shape points for which coordinates must be transmitted and with this also in terms of bit rate reduction. It has no interdependency during the iterative refinement of the approximation in different parts of the shape, and it has a natural look. Predictive coding is proposed for further bit rate reduction.

1. Introduction

This paper gives a technical description of the proposal submitted by University of Hannover for tool evaluation in MPEG–4. Proponents are RACE/MAVT, University of Hannover, Robert Bosch GmbH, and Deutsche Telekom. Goal of the tool is the efficient coding of arbitrary shapes of any image content. Approximation shall represent a shape with only a small portion of the original bit rate but with only a little deviation so that reconstruction errors introduced by the approximation are almost not visible. The proposal is based on [3].

2. Initialization

Given a mask with 2 regions as input, the boundary between these 2 regions shall be approximated. First, an initial polygon is generated as shown in Fig. 1. For this, the maximum elongation of the shape is looked for. These points, called A and B, are connected by a straight line. To both sides of this line, the contour point with the longest distance is looked for, called C and D. These 4 points define a quadrangle which is used as initial polygon for the approximation.

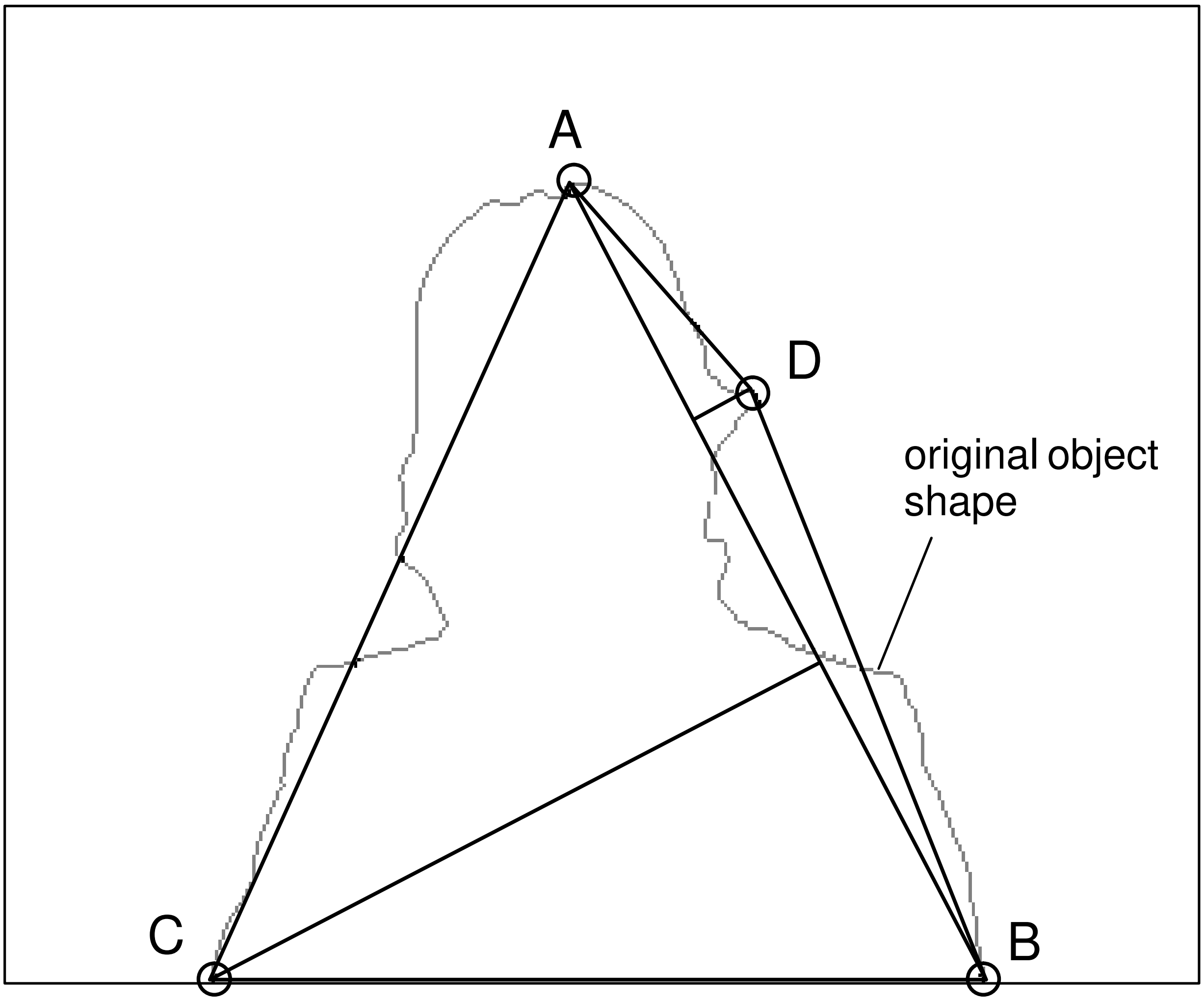

Fig. 1: Generation of the initial polygon.

3. Refinement

The initial polygon is iteratively refined until a given allowed maximum distance d_{max} is fulfilled everywhere. Fig. 2 shows the principle. For each polygon piece between two neighbouring vertices, it is tested whether the tolerance criterion is satisfied for each contour point which lies between the two vertices. Assume as an example that for the intervals (C,A), (A,D), and (B,C) this is true, but nor for the interval (D,B). Then, no refinement is necessary in (C,A), (A,D), and (B,C), but in (D,B). Here, the contour point with the maximum distance is looked for, and a new vertex, called E, is inserted at this position. For the two new polygon pieces (D,E) and (E,B), the test is repeated and, if necessary, again new vertices are inserted.

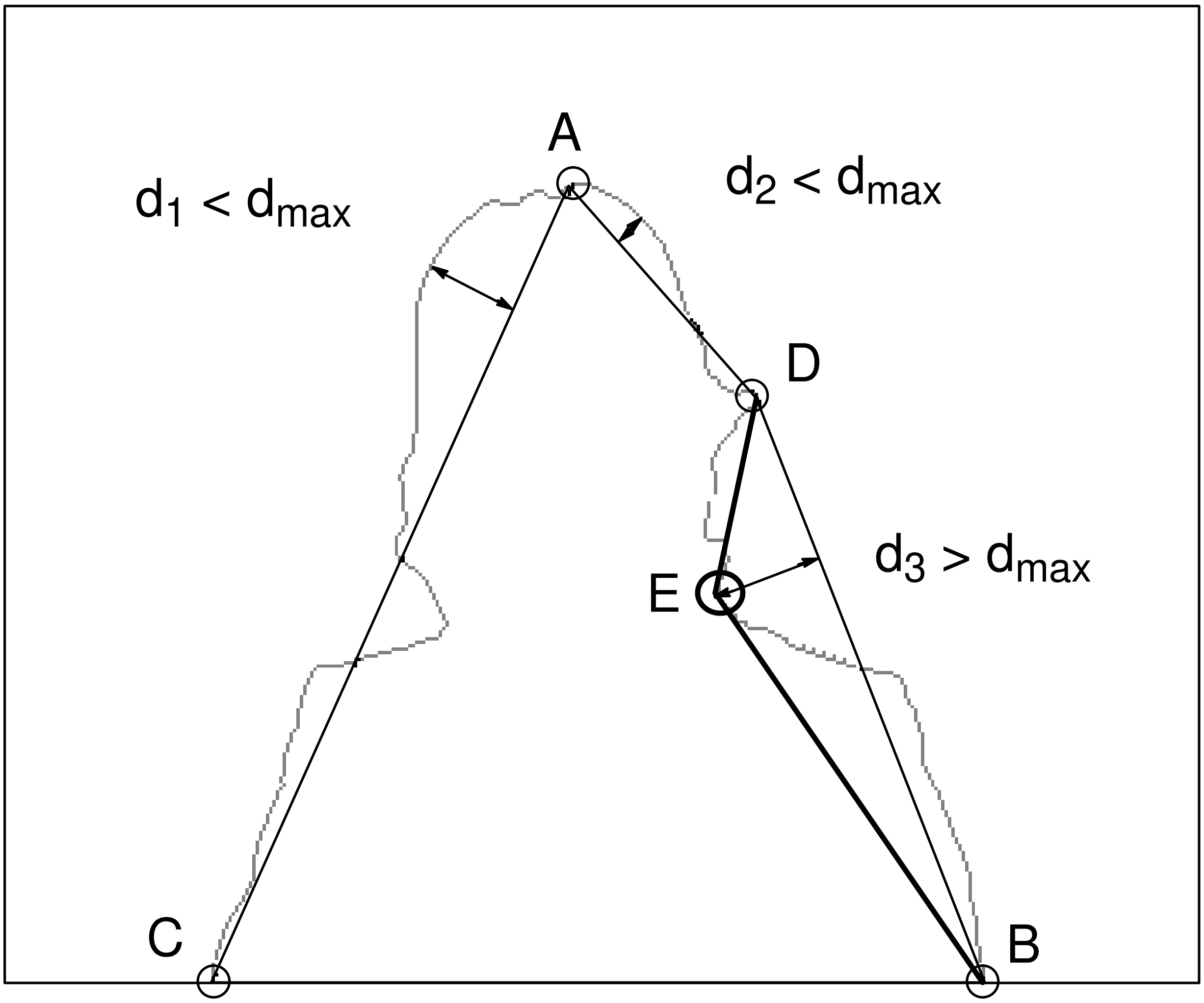

Fig. 2: Refinement of the initial polygon.

4. Combined polygon/spline approximation

Once polygon approximation is finished, the vertices are used to calculate a cubic B–spline approximation of the shape. Again, in each interval between two neighbouring vertices it is tested whether the tolerance criterion is satisfied for each contour point if spline approximation is used. If so, spline approximation substitutes polygon approximation in this interval. This will be mainly done where the shape describes a convex or concave form. Where the shape describes more or less a straight line polygon approximation will be mainly selected. Thus, the approximation is in general a combination of polygon pieces and spline pieces. An example of a final approximation result is given in Fig. 3. The allowed maximum distance d_{max} in Fig. 3 is lower than that used in Fig. 1 and Fig. 2.

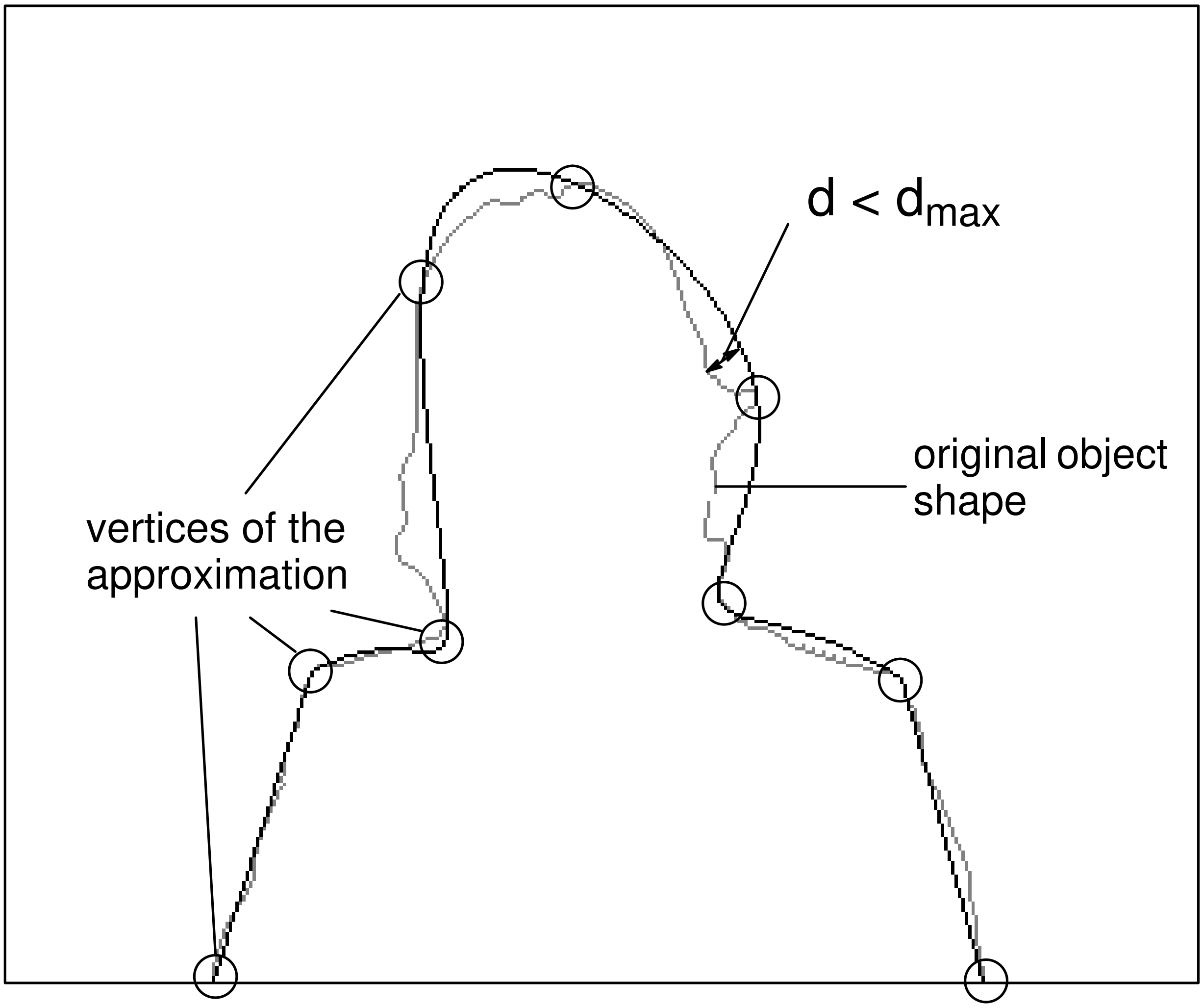

Fig. 3: Example of a final approximation result with 9 vertices.

5. Predictive coding

If an approximation of an old shape at a similar spatial position is available, that approximation can be used as prediction for the new shape. If additionally a displacement vector field is available, these displacements can be applied to shift the old vertices. It is then tested whether all shifted old vertices are within the tolerance range around the new shape. All vertices for which this does not hold are rejected from approximation. The maintained vertices form a polygon which is used in the refinement process instead of the 4–vertices initial polygon described in Section 2. If less than 3 vertices are maintained, prediction is not used.

If prediction is possible, not for all vertices the coordinates need to be transmitted. Instead, the following events are coded:

- runlengths of maintained and rejected vertices,
- runlengths of maintained vertices and vertices inserted during the refinement process,
- coordinates of inserted vertices relative to the respectively last maintained or inserted vertex,
- runlengths of the approximation functions (numbers of consecutive polygon pieces and consecutive spline pieces) or, in case also prediction of the approximation functions is applied [2], runlengths of the binary prediction errors.

If prediction is not possible, the following events are coded:

- coordinates of all vertices relative to the respectively last vertex,
- runlengths of the approximation functions (numbers of consecutive polygon pieces and consecutive spline pieces).

All events form 2–dimensional ensembles. Thus, the same entropy coding technique can be applied to all ensembles.

6. Introduction

This paper gives a technical description of the proposal submitted by University of Hannover for tool evaluation in MPEG–4. Proponents are RACE/MAVT, University of Hannover, Robert Bosch GmbH, and Deutsche Telekom. Goal of the tool is the efficient coding of arbitrary shapes of any image content. Approximation shall represent a shape with only a small portion of the original bit rate but with only a little deviation so that reconstruction errors introduced by the approximation are almost not visible. The proposal is based on [3].

7. Initialization

Given a mask with 2 regions as input, the boundary between these 2 regions shall be approximated. First, an initial polygon is generated as shown in Fig. 4. For this, the maximum elongation of the shape is looked for. These points, called A and B, are connected by a straight line. To both sides of this line, the contour point with the longest distance is looked for, called C and D. These 4 points define a quadrangle which is used as initial polygon for the approximation.

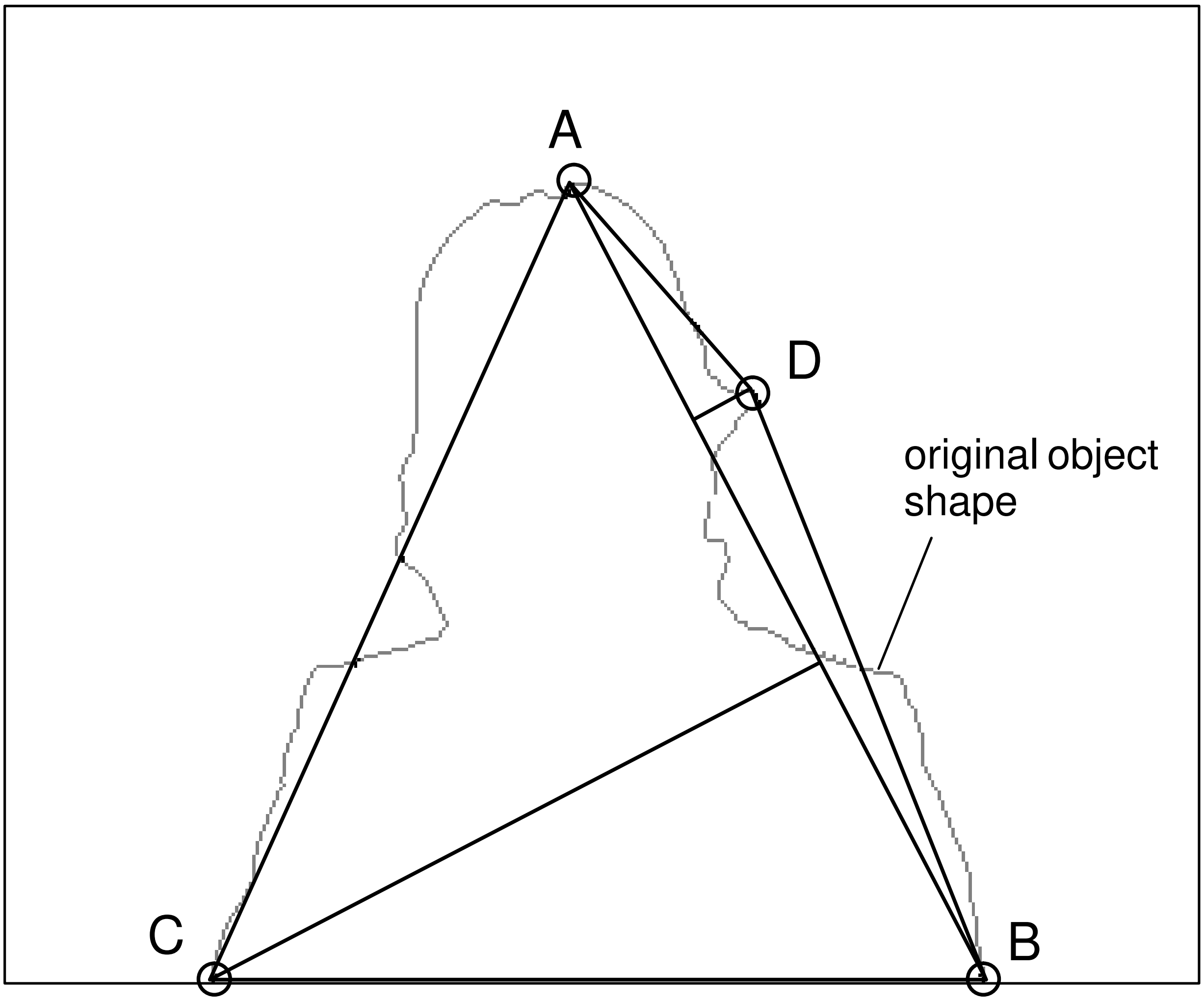

Fig. 1: Generation of the initial polygon.

8. Refinement

The initial polygon is iteratively refined until a given allowed maximum distance d_{max} is fulfilled everywhere. Fig. 5 shows the principle. For each polygon piece between two neighbouring vertices, it is tested whether the tolerance criterion is satisfied for each contour point which lies between the two vertices. Assume as an example that for the intervals (C,A), (A,D), and (B,C) this is true, but nor for the interval (D,B). Then, no refinement is necessary in (C,A), (A,D), and (B,C), but in (D,B). Here, the contour point with the maximum distance is looked for, and a new vertex, called E, is inserted at this position. For the two new polygon pieces (D,E) and (E,B), the test is repeated and, if necessary, again new vertices are inserted.

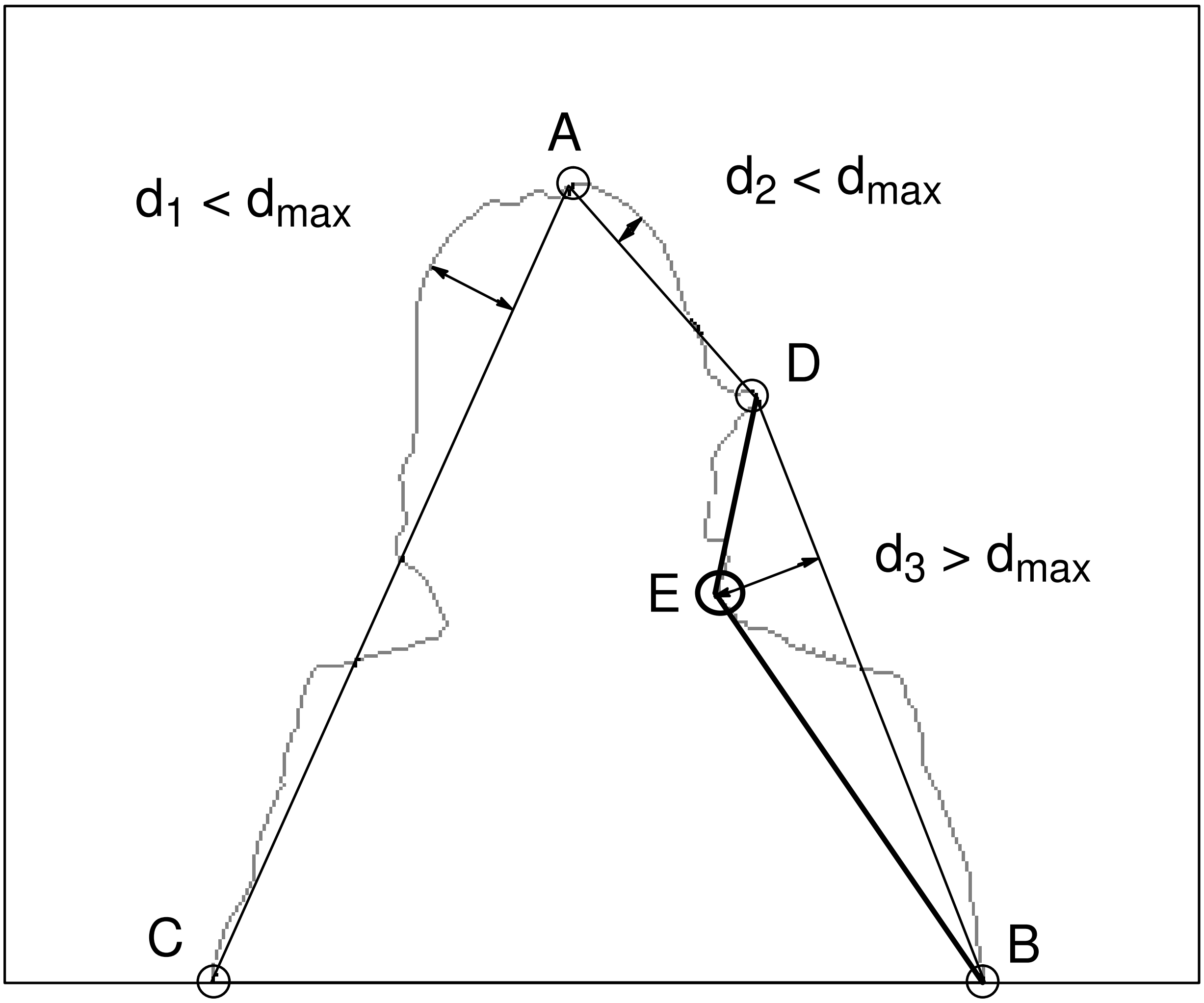

Fig. 2: Refinement of the initial polygon.

9. Combined polygon/spline approximation

Once polygon approximation is finished, the vertices are used to calculate a cubic B–spline approximation of the shape. Again, in each interval between two neighbouring vertices it is tested whether the tolerance criterion is satisfied for each contour point if spline approximation is used. If so, spline approximation substitutes polygon approximation in this interval. This will be mainly done where the shape describes a convex or concave form. Where the shape describes more or less a straight line polygon approximation will be mainly selected. Thus, the approximation is in general a combination of polygon pieces and spline pieces. An example of a final approximation result is given in Fig. 6. The allowed maximum distance d_{max} in Fig. 3 is lower than that used in Fig. 1 and Fig. 2.

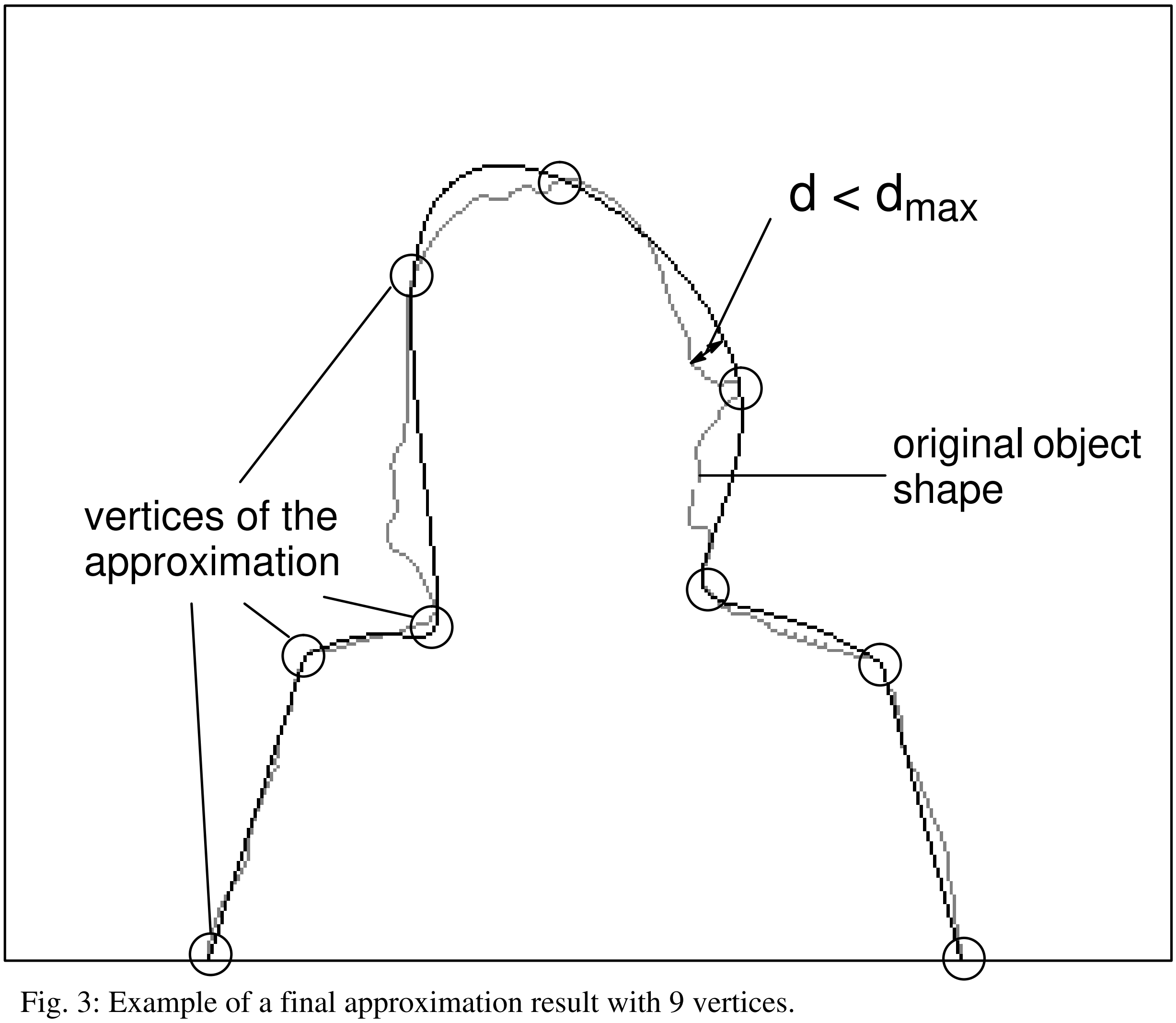

Fig. 3: Example of a final approximation result with 9 vertices.

10. Predictive coding

If an approximation of an old shape at a similar spatial position is available, that approximation can be used as prediction for the new shape. If additionally a displacement vector field is available, these displacements can be applied to shift the old vertices. It is then tested whether all shifted old vertices are within the tolerance range around the new shape. All vertices for which this does not hold are rejected from approximation. The maintained vertices form a polygon which is used in the refinement process instead of the 4–vertices initial polygon described in Section 2. If less than 3 vertices are maintained, prediction is not used.

If prediction is possible, not for all vertices the coordinates need to be transmitted. Instead, the following events are coded:

- runlengths of maintained and rejected vertices,
- runlengths of maintained vertices and vertices inserted during the refinement process,
- coordinates of inserted vertices relative to the respectively last maintained or inserted vertex,
- runlengths of the approximation functions (numbers of consecutive polygon pieces and consecutive spline pieces) or, in case also prediction of the approximation functions is applied [2], runlengths of the binary prediction errors.

If prediction is not possible, the following events are coded:

- coordinates of all vertices relative to the respectively last vertex,
- runlengths of the approximation functions (numbers of consecutive polygon pieces and consecutive spline pieces).

All events form 2–dimensional ensembles. Thus, the same entropy coding technique can be applied to all ensembles.

11. Properties of the polygon/spline approximation

- The approximation is very efficient (due to the polygon part of approximation the number of shape points for which coordinates must be transmitted is reduced to a small portion, order of magnitude: 1/20 to 1/30 even for very low values of the allowed maximum distance d_{max}).
- The approximation is free of mutual dependencies among different intervals during the refinement process (due to the polygon part of approximation and the use of the polygon vertices for spline approximation).
- The approximation looks natural (due to the use of spline pieces whenever they satisfy the tolerance criterion).

12. Statistical results

Average bit rates measured as total amount of bits for coding a shape related to the number of contour points are given for different image sequences in Table 1. For comparison, bit rates achieved by lossless chain codes are given in literature to be around 1.3 bit per contour point [1][4].

Test sequence	Bit rate
Akiyo	0.35
Sean	0.54
Mother & Daughter	0.68
Hall monitor	1.2

Table 1: Bit rates in bit per contour point for different test sequences.

13. Performance

13.1 Efficacy

Arbitrary shapes of any image contents can be approximated with the submitted tool. Image region shapes are efficiently coded by the proposed approach. Thus, it addresses both improved coding efficiency as well as content–based functionalities.

13.2 Adaptability

13.2.1 to different input

The proposed approach is useful for different classes of applications, where image regions are differently characterized esp. by different size, e.g. in videophone applications as well as in surveillance applications.

13.2.2 to (number of) objects

There is no restriction by the number and/or the kind of objects.

13.2.3 to different/variable bit rates

The required bitrate depends on the chosen approximation tolerance. Thus, an adaptation to different/variable bitrates is possible by varying the tolerance parameter.

13.2.4 to different error patterns

Error protection is possible as demonstrated by the proposed tool for error robustness submitted by Siemens AG [5].

13.2.5 to different temporal/spatial resolution

The temporal resolution influences only the efficiency of the prediction, but not the applicability of the approach. There is no restriction by different spatial resolutions.

13.3 Complexity

13.3.1 Delay

The delay depends on the size and the position of the image region.

13.3.2 Memory

The required memory is very small, since the algorithm only needs an input and an output mask. Additionally memory for storing the vertex positions is also very small, leading to an estimated maximum required memory of 100 or 500 KB for QCIF or CIF images, respectively.

13.3.3 Computational power

The computational power required for the tool is approximately 3 MOPS/frame for QCIF and 5 MOPS/frame for CIF images.

13.3.4 Complexity scalability (quality vs. h/w costs)

Since the required computational power is mainly due to the image size, a possible complexity scalability could be achieved by processing only a subsampled version of the image region mask which is to be approximated. This would lead to additional slight approximation errors, while the required computational power is reduced.

14. References

[1] M. Eden, M. Kocher, ”On the performance of a contour coding algorithm in the context of image coding. Part I: Contour segment coding”, *Signal Processing,* Vol. 8, No. 4, July 1985, pp. 381–386.

[2] P. Gerken, ”Object–based analysis–synthesis coding of image sequences at very low bit rates”, *IEEE Transactions on Circuits and Systems for Video Technology, Special Issue on Very Low Bit Rate Video Coding,* Vol. 4, No. 3, June 1994, pp. 228–235.

[3] M. Hötter, ”Predictive contour coding for an object–oriented analysis–synthesis coder”, *IEEE International Symposium on Information Theory*, San Diego, California, U.S.A., January 1990, p. 75.

[4] E. Persoon, K.S. Fu ”Shape discrimination using Fourier descriptors”, *IEEE Transaction on Systems, Man, and Cybernetics,* Vol. 7, 1977, pp. 170–179.

[5] F. Seytter, ”Hybrid ARQ type–II error protection tool”, *Doc. ISO/IEC JTC1/SC29/WG11 MPEG95/xxx,* November 1995.