

# PROGRESSIVE POLYGON ENCODING OF SEGMENTATION MAPS

Marc Servais and Theo Vlachos

CVSSP, University of Surrey  
Guildford, GU2 7XH, United Kingdom  
m.servais@surrey.ac.uk

Thomas Davies

BBC Research and Development  
Kingswood Warren, Tadworth,  
Surrey, KT20 6NP, United Kingdom

## ABSTRACT

A new method for the lossy compression of segmentation maps is presented. In the first stage of the proposed algorithm, region boundary segments are coded, providing a coarse polygonal approximation of regions, as well as a complete representation of their adjacency. The second stage consists of the progressive refinement of these segments, until either the desired degree of accuracy is reached, or the target bit-rate is achieved. Rate-distortion curves demonstrate good lossy performance in the range of 0.3 to 0.6 bits per contour point.

## 1. INTRODUCTION

Segmentation maps are employed in a variety of applications, including object-based video processing and coding, where they are used to define a region-based description of an image. Every region has a unique label and consists of a connected neighbourhood of pixels. It is often desirable to compress segmentation maps as they contain a high degree of redundancy. An important consideration is how to achieve the right balance between the degree of compression on the one hand and the fidelity of the decoded map on the other.

Shape compression (for individual foreground objects) has been researched extensively in recent years and two main approaches have emerged: direct bitmap coding of the object mask, and coding of the shape boundary. The latter method has been shown to allow a greater degree of compression [1]. Region boundary coding is typically achieved through the use of (lossless) chain-coding or (lossy) polygon/spline approximation [2].

In the case of lossy region boundary coding (with polygons or splines), the goal is to minimise both the coding cost and the distortion caused by approximating the boundary [2, 3, 4]. Some approaches impose additional criteria, such as minimising the number of polygon vertices for a given permissible error [5]. Progressive polygon encoding [6] has

been proposed as a way of enabling progressive transmission of a region's shape. This can proceed until either the desired bit-rate is reached, or the error is sufficiently small.

The methods outlined above are for *individual* regions (e.g. video objects in MPEG-4). However, a segmentation map can be viewed as a collection of regions. This suggests that many of the approaches used in shape coding would be useful in the coding of segmentation maps. Furthermore, the coding cost can be reduced because common boundaries between neighbouring regions only need to be coded once.

Some research into the coding of segmentation maps has been reported within the context of region-based video coding [1, 7, 8]. However, in these cases the aim was to minimise the *global* coding cost, thus not necessarily achieving an optimally coded segmentation map.

This paper describes a novel algorithm for the *progressive* encoding of segmentation maps. First, common boundary segments between neighbouring regions are identified. Using these segments an initial polygon structure is created, which only coarsely approximates the actual map, but which fully specifies the connectivity between neighbouring regions (Section 2). Following this, the existing segments are progressively refined until either the desired bit-rate is reached, or an acceptable error is achieved (Section 3).

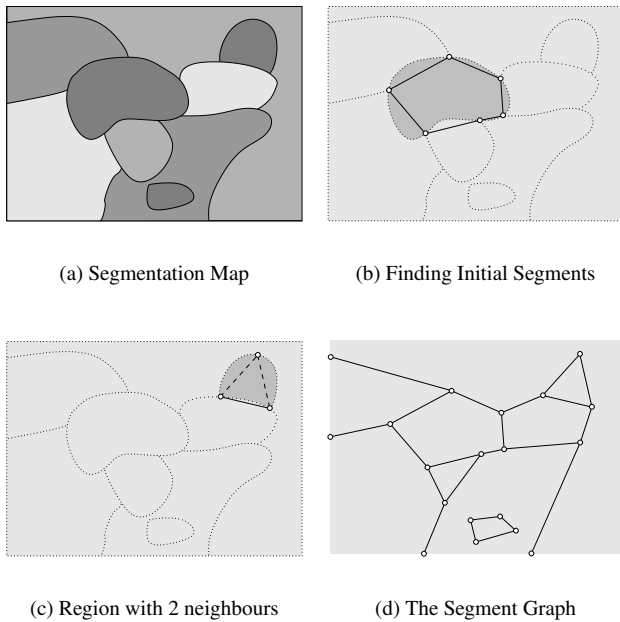
Finally, experimental results are presented for two segmentation maps encoded at a variety of bit-rates (Section 4). The performance of the proposed method and the advantages it offers to region-based video coding schemes are discussed.

## 2. ENCODING INITIAL POLYGON STRUCTURE

Given a segmentation map comprising a number of regions, the goal is to approximate each region with a polygon. However, a simple polygon approximation of each region in turn is likely to result in a new segmentation map in which the the polygon-shaped regions are not correctly aligned. This is because the polygons corresponding to neighbouring regions could either overlap or leave gaps along their common boundary. This can be avoided by the method outlined below and illustrated in Figure 1:

---

This work has been funded by BBC Research and Development and the Centre for Vision Speech and Signal Processing, University of Surrey.



**Fig. 1.** Generating the initial segments

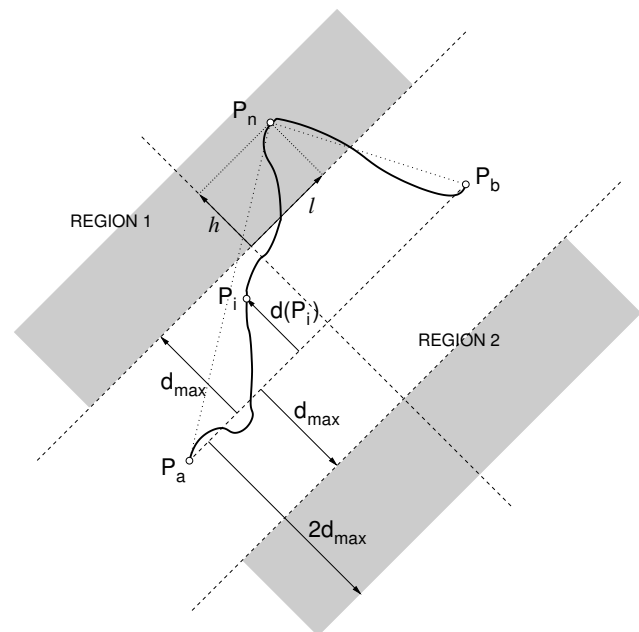
- For each pair of neighbouring regions, the common region boundaries are identified.
- Each common boundary is approximated by a straight-line segment connecting the two common boundary endpoints.
- A region can then be approximated by a polygon comprising the segments along that region's boundary, as depicted in Figure 1(b).
- If a region only has two neighbours, it will just have one segment associated with it. In this scenario, two extra segments are created. This is done by joining the two existing endpoints with the boundary point furthest from the segment, as illustrated in Figure 1(c).
- Those regions totally surrounded by another region are also treated differently, since their boundaries are circular. Such regions are approximated by four segments according to the initial polygon approximation algorithm in [9]. This involves selecting the two boundary points that are furthest apart, plus another two points furthest from the line joining the first pair.
- It can occasionally happen that some of the initial segments intersect one another. Should this occur, the crossing segments are refined (using the process outlined in Section 3) until they no longer cross.

Taken collectively, the initial segments provide a rough approximation of region shapes, as well as a complete representation of the adjacency between regions. Having determined the initial segments, the encoder's task is to compress them efficiently. This is achieved by first representing

the segments as a graph of connected nodes, as shown in Figure 1(d). Graph-based coding (based on the approach described in [8]) with relative addressing of node positions and entropy coding are then used to achieve compression.

### 3. PROGRESSIVE REFINEMENT

Once the initial segments have been specified, a more accurate representation of the original segmentation map can be obtained, by refining each of the segments in turn. As illustrated in Figure 2, the progressive refinement algorithm proceeds as follows:



**Fig. 2.** Refining a segment (between the points  $P_a$  and  $P_b$ )

- Consider the contour from  $P_a$  to  $P_b$  and the straight-line segment between the same two points.
- For each point  $P_i$  along the contour, calculate the perpendicular distance,  $d(P_i)$ , between  $P_i$  and the line segment from  $P_a$  to  $P_b$ .
- For some given threshold,  $d_{max}$ : if there exists a point  $P_i$  for which  $d(P_i) > d_{max}$ , then the segment needs to be split. If there is no such point, then the segment is considered sufficient.
- If splitting is required then:
  - Encode  $SPLIT = 1$ .
  - Find the point along the contour that has the greatest perpendicular distance from the straight line segment  $P_aP_b$ . Call this point  $P_n$ .
  - Encode the position of point  $P_n$ .
  - Discard segment  $P_aP_b$  and replace it with two new straight-line segments:  $P_aP_n$  and  $P_nP_b$ .

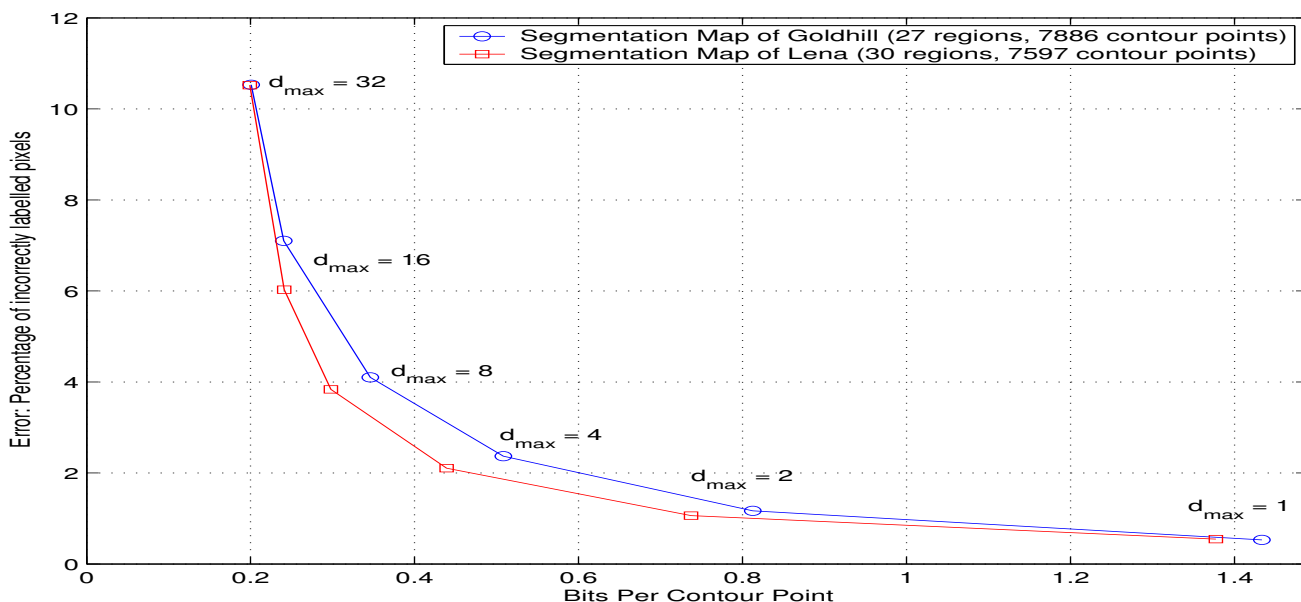


Fig. 3. Rate-distortion curves for progressive polygon encoding of two segmentation maps

- Otherwise (if no splitting required): Encode  $SPLIT = 0$ .
- Proceed to the next segment.

The above process is applied to all segments for a large initial value of  $d_{max}$  (e.g. 64). Once all segments have been checked,  $d_{max}$  is halved and the process is repeated. This continues until either a small enough value of  $d_{max}$  is reached, or the target number of bits is attained.

When performing entropy coding of the  $SPLIT$  flag, the following rules of thumb were found to provide a reasonable guide to actual statistics:

- For large  $d_{max}$ :  $Probability(SPLIT == 1) \approx \frac{1}{d_{max}}$ .
- When  $d_{max}$  is halved, the number of segments likely to split will double.

Furthermore, when coding the position of the new node,  $P_n$ , the following properties were found to be useful:

- Clearly  $d(P_n) > d_{max}$ . Note that if this were not the case, the segment would not need to be split.
- In the vast majority of cases  $d(P_n) < 2d_{max}$ . This is because for  $d(P_n) > 2d_{max}$  the segment would (almost always) have been split in the previous iteration.
- In general,  $P_n$  is more likely to occur close to the midpoint of the line between  $P_a$  and  $P_b$ , than far beyond either of the two endpoints.

Thus in roughly 95 % of cases,  $P_n$  falls into one of the two regions shaded grey in Figure 2. The above properties allow for the position of  $P_n$  to be encoded efficiently, using entropy coding of the following three values:

- Side: It is necessary to specify whether  $P_n$  lies on the left or the right of the straight line from  $P_a$  to  $P_b$ . (In Figure 2 it lies on the left.)
- Perpendicular distance:  $h$  is equivalent to  $d(P_n) - d_{max}$ .
- Parallel distance:  $l$  is the component parallel to  $P_aP_b$  and is measured from the midpoint of the line  $P_aP_b$ .

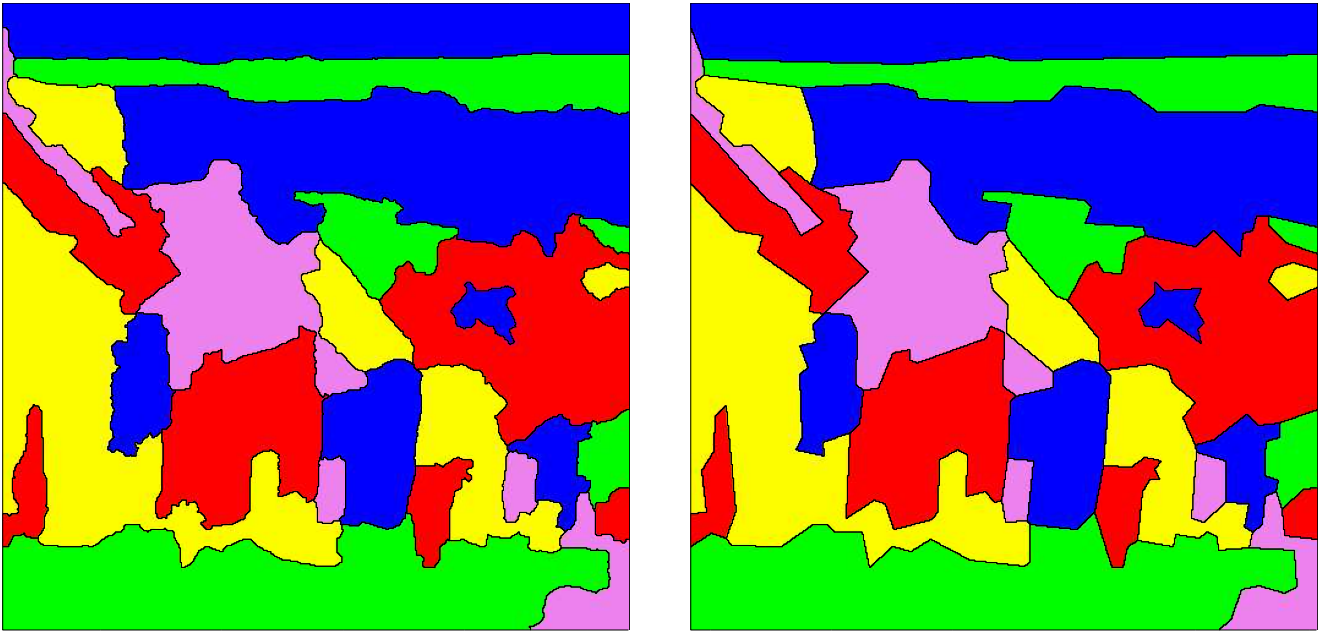
The algorithm outlined above allows for progressive refinement of the segmentation map in such a way that an *embedded* bit-stream is produced. Thus a decoder can decode the bit-stream until the desired level of refinement is achieved (i.e. a small enough value of  $d_{max}$ ).

## 4. RESULTS

The proposed algorithm was tested on two segmentation maps derived from the *Lena* and *Goldhill* images. The rate-distortion curves are plotted in Figure 3. It can be seen that the cost of encoding the coarse segmentation maps is approximately 0.2 bits per contour point (bpcp). After this, the distortion falls off rapidly as the bit-rate is increased. (The total number of contour points does not include those along the image boundary, since these are trivial to encode.)

No standard segmentation maps exist, so a direct comparison between different methods is difficult. However, the observed results compare favourably with the values of 0.65 bpcp (for  $d_{max} = 4$ ) and 0.5 bpcp (for  $d_{max} = 8$ ) obtained in [3]. Lossy compression rates of 0.4 to 0.6 bpcp are reported in [7], though the corresponding error is not quoted.

Figure 4(a) shows the original  $512 \times 512$  *Goldhill* segmentation map. This was encoded (using the proposed me-



**Fig. 4.** (a) Original segmentation map; (b) Polygon approximation for  $d_{max} = 4$  (with 249 nodes and requiring 499 bytes)

thod) to produce a bit-stream of 499 bytes, for  $d_{max} = 4$ . The resulting segmentation map is shown in Figure 4(b), with 97.6 % of the pixels correctly labelled.

A MATLAB demonstration can be downloaded from [www.ee.surrey.ac.uk/CVSSP/VMRG/hdtv/code.htm](http://www.ee.surrey.ac.uk/CVSSP/VMRG/hdtv/code.htm). This includes the two segmentation maps used in testing, as well as source code for the decoding of segmentation maps.

## 5. CONCLUSION

The results reported in this paper demonstrate that efficient lossy coding of segmentation maps is possible at rates in the order of 0.5 bits per contour point. The proposed method first encodes a set of initial segments in order to represent each region coarsely as a polygon. This is followed by the progressive refinement of these segments, until the desired accuracy or bit-rate is reached. It is anticipated that further gains may be obtained by a more accurate modelling of statistics, which should improve the entropy coding stage.

The progressive coding of segmentation maps offers advantages for object-based video coding, since it allows segmentation to be drawn within a rate-distortion framework in a video codec. (Ideally, the segmentation process itself needs to be similarly controllable to allow for a variable number of regions.)

Another option worth further investigation is allowing  $d_{max}$  to be weighted according to the similarity between neighbouring regions [4]. Thus if two regions have widely differing average intensities (and/or average motions),  $d_{max}$

should be reduced along their common boundary in order to allow for a more precise approximation of that boundary.

## 6. REFERENCES

- [1] J.Konrad, A-R Mansouri, E. Dubois, V-N Dang, and J-B Chartier, "On Motion Modelling and Estimation for Very Low Bit Rate Video Coding," in *Proc. VCIP*, 1995, pp. 262–273.
- [2] G. Schuster, G. Melnikov, and A. Katsaggelos, "Operationally Optimal Vertex-Based Shape Coding," *IEEE Signal Processing Magazine*, vol. 15, no. 6, pp. 91–108, November 1998.
- [3] K.J. O'Connell, "Object-Adaptive Vertex-Based Shape Coding Method," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, no. 1, pp. 251–255, February 1997.
- [4] L.P. Kondi, G. Melnikov, and A.K. Katsaggelos, "Joint Optimal Object Shape Estimation and Encoding," *IEEE Trans. Circuits Syst. Video Tech.*, vol. 14, no. 4, April 2004.
- [5] P.I. Hosur and Kai-Kuang Ma, "Optimal Algorithm for Progressive Polygon Approximation of Discrete Planar Curves," in *Proc. ICIP*, 1999, pp. 16–20.
- [6] C. Le Buhan Jordan and T. Ebrahimi, "Progressive Polygon Encoding of Shape Contours," in *Proc. IPA*, 1997, pp. 17–21.
- [7] S. Pateux and C. Labit, "Codage efficace de carte de segmentation pour la compression oriente rgions de squences d'images," Tech. Rep., IRISA publication 1073, Jan. 1997.
- [8] H. Nicolas, S. Pateux, and D. Le Guen, "Minimum Description Length Criterion and Segmentation Map Coding for Region-Based Video Compression," *IEEE Trans. Circuits Syst. Video Tech.*, vol. 11, no. 2, pp. 184–198, February 2001.
- [9] Y. Altunbasak and A.M. Tekalp, "Occlusion-Adaptive, Content-Based Mesh Design and Forward Tracking," *IEEE Trans. Image Proc.*, vol. 6, no. 9, September 1997.