PROGRESSIVE POLYGON ENCODING OF SHAPE CONTOURS

C Le Buhan Jordan and T Ebrahimi

Swiss Federal Institute of Technology EPFL, Switzerland

ABSTRACT

This paper presents a new progressive contour coding method based on digital polygonal approximation. By applying digital geometry concepts such as digital contours, segments and distances at the approximation stage, it is possible to design an inherently progressive encoding scheme which directly exploits the resulting geometrical knowledge. Experimental results show a performance similar to that of state of the art polygon encoding methods that do not achieve progressive transmission. Applications include indexing and retrieval of arbitrarily shaped objects in image databases, as well as composition and manipulation of such image objects in the compressed domain as targeted by the future video compression standard MPEG-4.

INTRODUCTION

Emerging multimedia applications, as targeted by the future MPEG-4 standard (9), will offer independent manipulation and transmission of different arbitrarily shaped objects within a given audio-visual scene, where natural video object planes (VOP) may be combined with computer generated visual objects. The shape of each image object (binary mask) will be encoded together with the texture and motion information, without any reference to other objects or the image background, to facilitate scene composition. Various shape coding methods are currently investigated in the framework of MPEG-4 (8).

As a semantically meaningful information, shape is also a key parameter for image database indexing and retrieval. Such an application preferably requires a progressive representation scheme, enabling rough browsing as well as perfect final rendering, and a semantic shape description, which is invariant to rotation and scaling, to facilitate shape indexing, matching and retrieval.

Among various existing shape representation schemes (5), geometrical representations based on spline or polygonal approximation of shape contours have been applied in fields as various as pattern recognition, computer graphics and image coding. Contours are represented by their most salient points, called vertices, which are transmitted to the decoder. Geometrical shape manipulation can then be directly performed on those vertices rather than on the whole shape, prior to shape decoding and filling. While being intrinsically geometrical and semantic, the resulting representations achieve compression of the shape information over a wide range of bitrates, controlled by the desired approximation accuracy.

Existing polygon encoding methods compress the vertex information by means of efficient relative addressing methods, possibly combined with syntax adaptive arithmetic coding (2). Vertices are encoded in the order they are met along the contour. In accurate (possibly lossless) representations, visually meaningful vertices (at sharp angles for instance) cannot be distinguished from noise-level vertices that accurately describe the original contour: vertex saliency is not considered.

In shape indexing, retrieval and matching applications, a simple yet salient representation of the shape information is required, while accurate shape data also needs to be stored for final image rendering. These requirements may be fulfilled by a progressive representation. In this paper, a new scaleable polygonal approximation and encoding method is proposed. The contour extraction and approximation algorithms are described first. It is then explained how digital geometry constraints may be used at the encoding stage to reduce the information range and achieve efficient compression of the vertex information in a progressive manner. Experimental results are presented which show a good performance of the proposed scheme when compared to corresponding state of the art methods.

CONTOUR REPRESENTATION

The first step in any contour coding method is to describe the shape in terms of its contours. Contour definition in a digital image is a tough problem when an exact reconstruction is desired for any existing shape, possibly including thin details of one pixel width.

A classical contour representation approach maps contour points onto digital image pixels. The shape inner contour is considered in 8-adjacency, while the outer contour is described in 4-adjacency. This dual representation allows consistent contour extraction, but faces some difficulties when thin details of width one pixel need to be represented in a reversible manner, as the contour crosses and overlaps itself. Existing algorithms are quite complicated for both contour extraction and region filling.

Another approach consists in considering contours as they are intuitively and mathematically defined: the shape contour is the boundary between the shape interior and the shape exterior. The digital topology approach recently developed by Kovalevsky (3) describes the digital image in terms of cell-lists, consisting of a 2D image pixel, associated 1D boundary segments (6-adjacent crack edges, vertical and horizontal edges which separate the shape interior and exterior pixels), which bound the 2D pixel, and 0D elements (corner points), which bound the 1D elements. Contours in the 2D digital space are then described as a set of connected 1D/0D elements (Fig.1). Corresponding contour extraction and shape filling algorithms are simpler and topologically more consistent with such an "inter-pixel" representation. Moreover, some corresponding properties may be directly exploited at the encoding stage, such as the fact that contour points are unique.

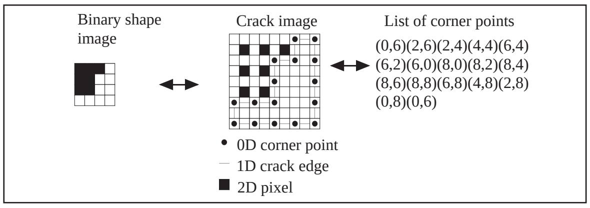

Figure 1. Inter-pixel contour representation.

DIGITAL POLYGONAL APPROXIMATION

Shape contours are first extracted. As points are more suitable than crack edges for geometrical processing (square grid), each contour is described in a compact form by an ordered list of 0D corner points with integral coordinates. Since the 1D information is implicit in the list order, this representation is complete.

Each contour may be approximated by a polygon at any desired accuracy, possibly lossless (1). Polygonal approximation is often obtained by a recursive refinement algorithm. The main axis of the contour is first extracted, providing the first two vertices. Each polygon edge is then recursively split by introducing a new vertex at the most distant associated contour point, until the desired accuracy is reached (Fig.2).

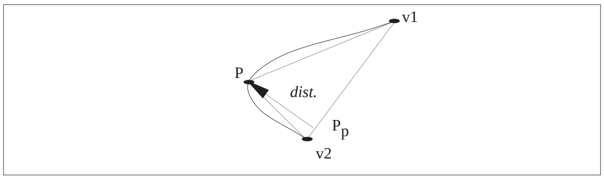

Figure 2. Edge [v1v2] is split by adding a new vertex in P if the corresponding distance is larger than the tolerated error.

This accuracy is often defined as the peak Euclidean distance from the approximation to the original curve, as it can be mathematically computed. However, this distance measure does not take into account the intrinsic image grid quantization, and is therefore too severe, especially in the lossless case. Indeed, the computed Euclidean error may be non-zero even when a lossless reconstruction is obtained, due to final edge tracing in the discrete image grid. By considering a digital distance such as the chessboard distance rather than the Euclidean distance, it is possible to save up to 40% in the total number of vertices for lossless reconstruction (4):

$$d_8(P_1,P_2)=\max(|x_2-x_1|,|y_2-y_1|) \qquad \text{(Eq.1)}$$

PROGRESSIVE APPROXIMATION

The use of a discrete distance also enables a progressive contour approximation which can be further exploited at the encoding stage. A progressive representation is defined by a coarse approximation and its successive refinements, possibly down to the maximal accuracy. Each refinement is defined *relative* to its parent approximation. Progressive polygonal approximation consists in extracting first a coarse polygon, which constitutes the upper approximation level defined by a tolerated error $d_8=n$. Starting from this coarse polygon, the recursive refinement procedure is applied with a decreased target accuracy $d_8=n$-1. This yields the level n-1 polygon refinement. The procedure is repeated until the final desired accuracy is reached, usually $d_8=0$ for a lossless representation. This progressive polygon refinement results in n+1 representation levels, from the coarsest (most salient vertices) to the most accurate one.

PROGRESSIVE ENCODING SCHEME

Overview

Progressive vertex encoding consists in transmitting first the raw polygonal approximation obtained at the upper level, then its successive refinements. A series of vertex positions defining the first polygon must be sent first. For refinement polygons, information about the number of child vertices along the coarse polygon edges must be transmitted, so that the decoder can correctly produce the ordered list of vertices. The positions of these refinement vertices can then be encoded relative to their parent edge. The order of transmission is illustrated in Fig.3, for an image containing P approximated contours which are encoded with n+1-k scalability levels.

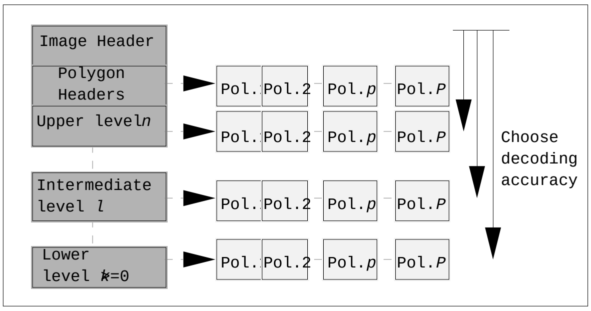

Figure 3. Progressive transmission of polygon approximation.

Efficiently encoding a series of vertex coordinates is difficult due to the lack of any underlying model. Raw approximations consist in a reduced set of vertices poorly correlated: it is not easy to guess where a vertex is likely to be, even when the previous transmitted vertices along the contour are known. Existing polygonal encoding schemes use differential encoding of the coordinates combined with sophisticated relative addressing, such as decomposition into octants (6).

Adaptive arithmetic coding that dynamically fits the data may also be used, at the price of an increased complexity (2).

In order to improve the encoding performance, the progressive polygonal approximation can be exploited at the encoding stage. Indeed, while encoding as well as decoding, information from already transmitted upper levels is available, except for the coarsest approximation level. Corresponding upper level vertices can be directly encoded (vertex position index in the contour image), in order to reduce the associated decoding complexity in applications where the salient polygon is often browsed (shape matching and retrieval). More expensive lower levels can then be encoded relatively to their parent approximation, by exploiting the associated knowledge, as explained in the following.

Geometrically Constrained Progressive Vertex Coding

While encoding as well as decoding polygon refinements, geometrical knowledge may be derived from available coarser approximations. In particular:

- A level *l* vertex must be within a chessboard distance *l*+1 from the previously encoded level *l*+1 polygon, otherwise it would have been introduced as a necessary vertex at level *l*+1 (Fig.4). (1)

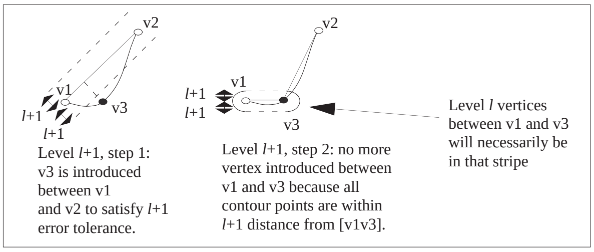

Figure 4. Stripe geometrical rule (1).

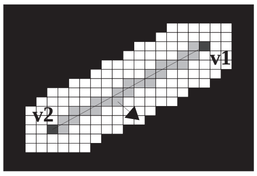

Figure 5. Reduced set of possible positions for child vertices along a level 2 parent edge, geometrical rule (1).

This yields a drastic reduction in the number of possible positions for the corresponding vertices (Fig.5). By reducing the data range for vertex indices, fewer bits are required to encode the corresponding values, resulting in a higher compression. It is possible to define a set of additional geometrical rules that reduce even more the set of possibilities for a vertex position, based on an exhaustive analysis of the local geometrical context:

- Vertices necessary belong to their contour bounding box. (2)
- For a single child vertex, the possible positions are at a chessboard distance *l*+1 from the parent edge. (3)
- A child vertex cannot overlap its parent vertices. (4)

For each level *l*+1 edge, these geometrical constraints initially reduce the discrete set of possible positions S for the associated level *l* child vertices. Vertex positions are then indexed left to right, top to bottom within S and encoded with $log_2(N_S)$ bits, where N_S is the number of positions in S: the smallest N_S, the highest the compression.

As soon as one vertex is transmitted, it is possible exploit the newly available knowledge: a complementary set of geometrical rules are applied to dynamically update the set of possible positions. These rules exploit the fact that vertices are chosen along the original contour, and therefore verify the corresponding knowledge (no overlaps, no crossings, unique contour points,...). Moreover, lossless edges also cover the original contour, and therefore satisfy the associated geometrical constraints. For these geometrical rules to be always valid, it may be necessary to constrain the polygonal approximation algorithm. For instance, the recursive refinement method used is suboptimal and may sometimes result in aligned vertices, as it is a split-only procedure without edge merging capability. This requires detection and removal of redundant vertices after polygonal approximation has been performed, and before geometrical coding. Details about polygonal approximation constraining can be found in (4).

Each time a new vertex has been transmitted, new positions are invalidated for all the following vertices while locally impossible positions, only invalid for the next vertex, are also filtered out. These dynamic rules are the following:

- There cannot be two child vertices at the same position. (5)
- A new vertex cannot overlap an existing edge. (6)
- A new vertex cannot be redundant relatively to the previous transmitted pair of vertices, that is the positions yielding an edge digitally aligned with the previous one are *locally* invalid (Fig.6). (7)

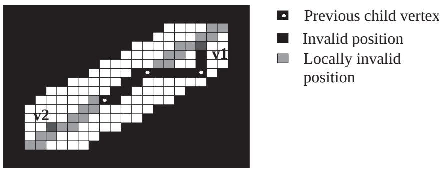

Figure 6. Dynamically reduced set of positions for child vertices along a level 2 parent edge, 3 previously transmitted child vertices, geometrical rules (5,6,7).

- A new edge cannot cross existing edges, that is the positions on the other side of any previously transmitted edge relatively to the previous vertex are *locally* invalid (Fig.7). (8)

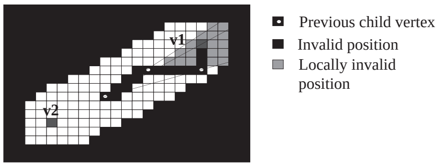

Figure 7. Dynamically reduced set of positions for child vertices along a level 2 parent edge, 3 previously transmitted child vertices, geometrical rule (8).

- A child vertex must be connected with the second parent vertex by a valid path. Consequently, dead-ends are not allowed: any position with less than two 4-connected valid or locally invalid neighbours is definitely invalid. (9)
- The new vertex must generate a valid edge: in the lossless case, any position yielding an edge overlapping at least one invalid position is *locally* invalid. In the lossy case, any position yielding an edge containing one parent vertex or the previous transmitted child vertex is *locally* invalid. (10)

Rules (7) and (10) can be applied to the last child vertex relatively to the second parent vertex (last child edge), thus providing two additional geometrical rules. At any refinement level, for each vertex, both the encoder and the decoder dynamically determine the discrete set of possible positions by successively applying the four static and eight dynamic rules. The vertex position is then indexed within this set of possible positions, resulting in efficient progressive vertex encoding.

EXPERIMENTAL RESULTS

The progressive transmission achieved with the proposed method is illustrated in Fig.8 for the first frame of the video sequence "News". The decoder may choose to decode only the beginning of the bitstream, corresponding to a coarse approximation of the shape.

The proposed method has been implemented and compared with various existing shape coding methods experimented in the framework of MPEG-4 (8), in intra coding mode. Detailed results can be found in (4). In particular, performance has been compared with existing polygon encoding schemes, both in scaleable and non-scaleable modes. These methods are:

- Relative addressing combined with Syntax Adaptive Arithmetic Coding (SAAC), as proposed by Gerken in (2).
- Octant-based decomposition combined with dynamic range relative addressing and generalised chain coding, as proposed by O'Connell in (6).
- Adaptation of the former method to achieve representation of salient vertices in a hierarchical manner, as proposed by Qian and Sezan in (7).

Corresponding results are plotted in Fig.9: total shape coding bits versus shape mismatch error (number of erroneous pixels in the reconstructed binary shape image normalised by the number of original shape pixels). The progressive polygon encoding method uses three scalability levels.

On one hand, the proposed scheme performs twice better than the hierarchical vertex encoding method. Indeed the latter technique encodes the vertex saliency information as an additional data field for each vertex without exploiting any progressive representation scheme, thus resulting in an important extra cost for scalability. On the other hand, the proposed method yields results very similar to that of the state of the art SAAC technique, while additionally offering progressive transmission. This state of the art technique has been much improved in the framework of MPEG-4 by octant-based relative addressing and generalised chain coding (6). This method achieves classical (Freeman) chain coding in the lossless case, resulting in a much higher compression but at the price of the loss of geometrical information: every contour point is a vertex. The proposed technique is outperformed by this method; in order to improve its performance, it is possible to introduce a chain coding mode at lossless level. Lossless refinement edges are encoded rather than vertices. As opposed to the non-scaleable generalised chain coding, geometrical information remains available for coarser levels. About 20% of the lossless bitrate can be saved. In a preliminary experiment, first order Huffman coding has been used. A subset of the geometrical rules can still be applied, resulting in the local removal of some possible moves, and consequently in the reduction of the associated chain code words length.

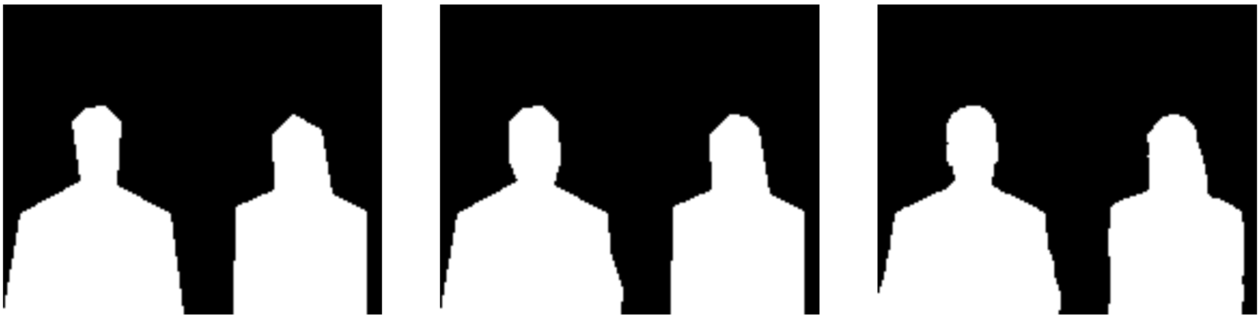

Figure 8. Progressive decoding of sequence 'News', frame 0. From left to right: coarse approximation (error 2, 341 bits), intermediate approximation (error 1, 416 bits), lossless representation (1087 bits).

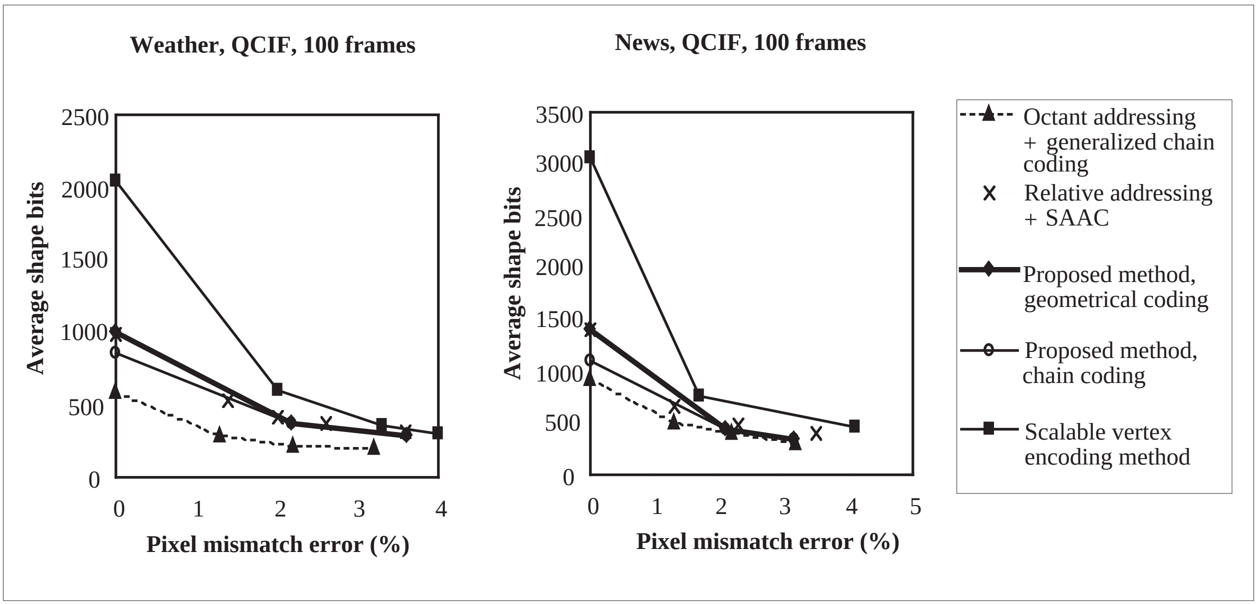

Figure 9. Comparison of various polygon encoding methods for two video sequences in intra mode, 100 frames, QCIF format.

These preliminary results demonstrate that digital geometrical knowledge can be exploited to perform efficient progressive polygon encoding. The compression performance may be further improved by means of statistical entropy coding. The proposed method can also be adapted to apply the efficient coding syntax of the octant-based generalised chain coding method, in order to reach the corresponding performance, particularly for coarse level encoding (at the price of additional complexity). This would additionally provide an efficient non-progressive mode for the proposed method. Future work also includes the adaptation of the scheme to perform temporal prediction and coding so that it is suitable for video coding applications.

CONCLUSIONS

In this paper a new progressive polygon encoding method for binary shapes is proposed. As a semantic and geometrical representation, vertex-based coding is particularly suitable for visual object coding, manipulation in compressed domain, indexing and retrieval, as targeted by the future video compression standard MPEG-4. While most state of the art polygon encoding schemes are non progressive, the proposed method achieves scaleable polygon transmission. To this aim, various concepts of digital geometry are exploited, such as discrete straight lines and crack edges to define shape contours. Progressive transmission is then achieved by encoding each polygon refinement relatively to the previously transmitted coarse polygonal approximation. A set of static and dynamic geometrical rules are applied to constrain the set of possible positions for refinement vertices, resulting in a reduced transmission cost for the corresponding indices. While outperforming the scaleable hierarchical polygonal encoding scheme which has been recently investigated in the framework of MPEG-4 Core Experiments, the proposed progressive method yields an intra coding performance similar to that of state of the art shape coding techniques which are non progressive.

REFERENCES

1. Dunham G., 1986, IEEE Trans. PAMI,8-1, 67-75

2. Gerken P., 1994, IEEE Trans. CSVT,4-3, 228-235.

3. Kovalevsky V.A., 1993, "Shape in Picture - Mathematical Description of Shape in Grey-level Images", NATO ASI Series F: Computer and Systems Sciences, 126, 21-36.

4. Le Buhan Jordan C., 1997, "A progressive contour coding scheme based on digital polygonal approximation", Technical Report EPFL-LTS no.97-02, Lausanne, Switzerland, avail. from http://ltswww.epfl.ch/~lebuhan/shape.html.

5. Marshall S., 1989, Image and Vision Computing, 7-4, 281-294.

6. O'Connell K.J., 1997, IEEE Trans. CSVT,7-1, 251-255.

7. Qian R., Sezan I., 1996, "Vertex-based hierarchical shape representation and coding", ISO/IEC JTC1/SC29/WG11, MPEG-4 Contribution M1187.

8. ISO/IEC JTC1/SC29/WG11, 1997, "Core experiments on MPEG-4 video shape coding", MPEG-4 Document N1471.

9. ISO/IEC JTC1/SC29/WG11, "MPEG Web Page", avail. from http://drogo.cselt.stet.it/mpeg.