Scalable Shape Representation for Content Based Visual Data Compression

C. Le Buhan Jordan F. Bossen T. Ebrahimi

Signal Processing Laboratory
Swiss Federal Institute of Technology
1015 Lausanne, Switzerland

Abstract

Two major classes of shape coding methods are reviewed, namely bitmap coding and contour coding, and in particular their scalable extensions. In addition to their absolute compression efficiency, we analyze their performance in the framework of a complete image/video coding scheme, and show that they bring complementary functionalities depending on the targeted application.

1 Introduction

Recent developments in multimedia applications (editing, video games and interactive video, computer generated graphics, etc.) have raised a need for new functionalities (such as object manipulation, selective object rendering, different temporal resolution of objects, etc.) in addition to improved compression efficiency. Object oriented coding seems a more natural approach to practical systems, as in these applications, the original source material is frequently composed of different objects put together in a mosaic form or in a layered fashion. Future compression standards such as MPEG-4, MPEG-7 and JPEG2000 will define a coding syntax based on arbitrary shaped visual objects (e.g. Video Objects -VOs- in MPEG-4 terminology) [14]. This implies that, in addition to texture (and motion in video), information about the shape of every object is encoded. Hence, various shape coding tools have been recently investigated in the framework of MPEG-4 standardization activities.

In this paper, emphasis is given to binary shape coding as opposed to gray scale shape or segmentation coding. Most binary shape coding techniques may be classified as either bitmap-based (alpha plane) or feature-based (contour, skeleton, vertex). We focus on two techniques representing these approaches, namely, bitmap compression based on higher order arithmetic coding, and polygonal approximation encoding. In multimedia networks, devices with different bandwidths and available decoding powers are inter-connected. Bitstream scalability is desirable, so that simple decoding of the first bits results in a coarse shape approximation that may be further refined. Such a coarse representation may also be used for indexing and retrieval of the shape information. Therefore, we focus on the adaptation of the above mentioned techniques to achieve progressive compression. We also discuss their advantages and drawbacks in a complete coding environment, with special attention to the interaction between shape coding and other coding tools (texture and motion coding).

2 Context-based Arithmetic Encoding (CAE)

The bitmap based approach considered here draws its source from text compression techniques. Langdon and Rissanen [8] proposed an efficient method based on finite state machines and arithmetic coding. The idea is quite simple: the image is coded pixel by pixel in a scanline order. For each pixel, the state of the finite state machine is defined by the values of pixels within a template. This template typically includes pixels in the close vicinity of the pixel to be coded. A probability distribution is associated with each state which is used to drive the arithmetic coder.

This technique may be extended to achieve scalable transmission [1, 9]. First the binary mask is decomposed into several layers of different resolutions. The base layer, that is the layer with the lowest resolution, is coded using the classical non-scalable technique. The enhancement layers, that is all the remaining layers, are then encoded in a similar fashion but using a different template. Whereas the template for coding the base layer only includes pixels from the current layer, the template for coding the enhancement layers also includes pixels from previously coded layers.

The main feature of this approach is its superior co-

ding efficiency, while bearing a relatively low complexity. It is also well adapted for low delay applications. For video coding applications, it has been extended to achieve block-based coding and temporal prediction, as described in [2]. The resulting method has been adopted as shape coding tool in the MPEG-4 Verification Model [14].

3 Progressive Polygon Encoding (PPE)

Shape may also be represented by their contours, like in the human visual system. This is of interest in applications where a high-level, semantic representation is needed. Hence, shape retrieval methods often process contour features, such as high curvature points on object boundaries [10]. If an adequate shape representation is used prior to entropy coding, semantic features can be accessed at the cost of entropy decoding only, while bitmap compressed data would require the decoder to perform contour analysis. From the image coding point of view however, additional processing stages are needed to extract the contours at the encoder side, and to fill the shapes for final rendering at the decoder side, which increases the complexity.

Geometrical approximation by polygons or higher order curves may be used as a high level feature representation, while offering quality control. In [11], each contour is recursively split into polygon edges until the approximation reaches a predefined error threshold. Resulting edge vertices are differentially encoded. When lossless representation is needed, this method is degenerated to achieve efficient chain coding, yet at the cost of losing the high-level geometrical features since every contour point becomes a vertex. The desire to combine efficient lossless representation for final rendering with high-level representation of most significant contour points in a single bitstream has led to the design of a Progressive Polygon Encoding (PPE) method [7]. First a coarse polygonal approximation is built. The resulting vertices correspond to salient points along the contour and may be encoded by any existing vertex coding method or even directly, so that the decoder can rapidly access them for fast browsing/retrieval. Complementary lossless representation for final rendering is achieved by successively transmitting the polygonal approximation refinements. The insertion order of the refinement vertices as well as their positions are encoded relatively to the coarser polygon edges. Efficient entropy coding is achieved by exploiting both the intrinsic image grid quantization and the geometrical knowledge available at the encoder and decoder, such as the fact that refinement vertices must be close to their parent edge. This method performs similar to classical non progressive polygonal encoding schemes, while enabling quality scalable transmission and/or decoding.

In video coding applications, contour based coding can also be adapted to exploit temporal redundancy [5]. Extension of the proposed quality scalable PPE scheme to temporal coding remains to be investigated. Actually, a feature-based shape representation for indexing and retrieval may be necessary in intra-coded frames only (random-access points in the video). Any non progressive temporal coding method, not necessarily semantic, can then complete the scheme between successive I-frames, possibly the efficient map-based temporal CAE method [2].

4 Discussion

In video coding applications, scalability may come in different flavors, namely spatial, quality (SNR) and temporal. As in this paper we do not investigate temporal prediction modes for shape, temporal scalability will not be dealt with. Spatial scalability consists in sending a sub-sampled image first, then its successive refinements up to the original image size. Quality scalability consists in transmitting a low quality image first, then progressively refining it possibly up to a lossless representation.

The scalability achieved by the CAE method can be viewed in both ways, either spatial or quality-wise. When decoding only a few layers the obtained mask is a sub-sampled version of the original one (spatial scalability). This mask may also be upsampled to the size of the original mask, leading to an approximated reconstruction (quality scalability). The PPE method directly achieves quality scalability: coarser approximations only contain the most salient features of the shape, typically high curvature points on the shape contour. In addition, a vertex description is a vectorial representation, which enables straightforward up/downsampling by any ratio (possibly non-integer) to achieve spatial scalability.

When defining scalability, an important parameter is the associated granularity, i.e. the number of refinement layers that may be defined. CAE refines the shape by doubling the image dimensions, while PPE refines a contour by decreasing the maximal discrete distance between the original and reconstructed curves by a unit step; in either case, it does not seem useful to encode more than 4 layers.

Coding efficiency

Scalable shape coding results obtained for both the contour-based PPE and map-based CAE methods without temporal prediction are presented in Fig. 1. Three scalability levels are used, which result in three

rate/distortion points: lossless, quasi-lossless, lossy. In practice, PPE enables lower distortions than CAE which is more efficient by up to 30% in the lossless case (full bitstream decoding).

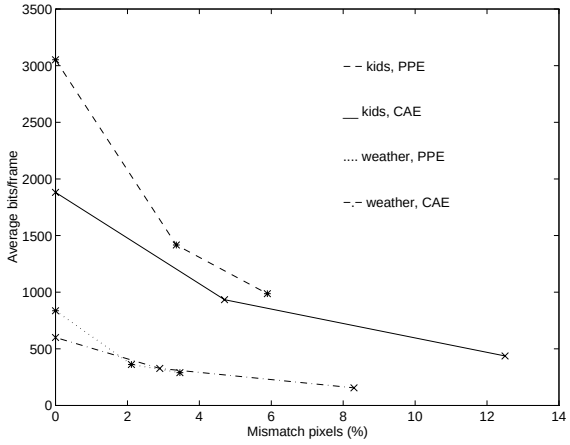

Figure 1: Rate/distortion curves, CAE vs PPE, Sequences 'Kids' (SIF) and 'Weather' (QCIF), 100 frames, 3 scalability levels.

Lossless shape coding may be simply evaluated in terms of number of transmitted bits. For both techniques, lossless coding requires about twice as many bits when compared to quasi-lossless coding. This cost, sometimes due to segmentation noise, may be prohibitive in some applications such as wireless communications. When evaluating the lossy coding performance, two major difficulties arise, namely, the choice of a shape distortion measure, and the estimation of the influence of lossy shape coding on the overall motion/texture coding scheme.

First, shape distortion should be measured to characterize visual artifacts on edges that are subjectively annoying. A boundary oriented distortion measure is often used, but it cannot take into account the removal/insertion of small isolated regions. Therefore we measure shape distortion as the percent ratio between the number of mismatched pixels and the original shape size. For subjective evaluation, decoded images corresponding to quasi-lossless quality are presented in Fig. 2. Polygonal approximation results in a low-pass filtering of the contours, while downsampling introduces blockiness.

As lossy shape coding results in either removing (erosive approximation) or adding (dilative approximation) some pixels from/to the texture data, it also affects the overall coding performance [6]. In MPEG-4, block-based texture and motion coding schemes are applied [14]. Where necessary, reference blocks are padded: exterior pixels are filled with a texture color predicted from the interior pixels along the shape boundary. Clearly, shape accuracy (as well as segmentation accuracy and shape downsampling) influences motion and texture representations, especially when background pixels are introduced along the shape boundaries (dilative case). Because of this interdependency, the design of an efficient rate-distortion control scheme for arbitrarily shaped objects is difficult, and lossy shape coding should be limited to small distortions.

Other functionalities

In order to embed a shape coding method in a complete coding scheme, application requirements and constraints must be carefully reviewed. When a macroblock based texture/motion coding scheme is applied, embedding a macro-block based shape coding scheme is straightforward. Many structures can be shared, such as the motion estimation scheme, which decreases the implementation complexity, decoding delay and memory requirements. In particular, it becomes possible to transmit macroblocks on the fly. The CAE method, which can be easily adapted to process macroblocks, is therefore well suited for classical DCT based coding methods [2]. In recently developed alternate methods to block-based coding, such as region-based or mesh-based compression, constraints are different. In their frame-based motion/texture coding syntax, these methods already implement the concepts of contours or vertices, either for the associated semantic or geometrical features. In this context, embedding a chain-based or vertex-based boundary representation is a consistent choice [4, 13].

Although highly desirable in some applications, joint shape/texture/motion scalability remains a difficult problem. While most existing progressive transmission schemes take advantage of frequency properties for motion/texture (e.g. transmit low frequency components first), shape scalability changes the size and shape of the spatial region of support, which complicates the mapping of motion/texture refinements to previously transmitted data. For instance, in a classical block-based scheme, blocks may be spatially shifted to match the refined shape. Therefore it may be necessary to consider shape and motion/texture scalability separately. In addition, some applications require independent access to shape, texture, motion fields for separate retrieval, editing and manipulation. In this case, a shape coding method can be used with features independent from the motion/texture coding schemes, to a certain extent. A VO-based coding method, possibly embedding specific shape fun-

Figure 2: Sequence 'Kids', frame 0. Left: original. Center: quasi-lossless, CAE (avg. 934 bits/frame). Right: quasi-lossless, PPE (avg. 1417 bits/frame).

ctionalities such as the proposed scalable schemes, can be used under the assumption that the whole (preferably lossless) shape will be decoded before final texture/motion rendering in order to limit the associated interdependencies.

Separate transmission of shape and motion/texture information may also help error-resilient object coding, if a dedicated protection scheme can be designed for the crucial shape information. However, a macroblock structure also brings specific adavantages, as macroblock data is encoded in one pass with dependency limited to previously transmitted closest neighbors. Error robustness for shape data is investigated in [3] for the CAE method, while a preliminary proposal for error resilient vertex coding was proposed in [12].

5 Conclusion

In this paper, two major classes of shape coding techniques and their scalable extensions are reviewed, namely, map-based arithmetic encoding and polygonal approximation encoding. It is shown that the evaluation of their respective performance should take into account the complete coding scheme, and that they bring complementary functionalities in accordance with the target applications. As an efficient and simple coding compression algorithm, the Context-based Arithmetic Encoding method (CAE) is well suited for generic block-based coding schemes, while in specific applications where geometrical or semantic shape features are utilized, for instance editing, indexing or retrieval, the Progressive Polygon Encoding method (PPE) provides a more suitable representation.

References

[1] F. Bossen and T. Ebrahimi. A simple and efficient binary shape coding technique based on bitmap representation. In *ICASSP*, 1997.

[2] N. Brady, F. Bossen, and N. Murphy. Context-based arithmetic encoding of 2D shape sequences. In *ICIP*, 1997.

[3] N. Brady and L. Ducla-Soares. Error resilience of arbitrarily shaped VOs (CE E14). Technical Report ISO/IEC JTC1/SC29/WG11/M2370, July 1997.

[4] M. Menezes de Sequeira and D. Cortez. Partitions: a taxonomy of types and representations and an overview of coding techniques. *Signal Processing:Image Communication*, 10:5–19, 1997.

[5] P. Gerken. Object-based analysis-synthesis coding of image sequences at very low bit rates. *IEEE Trans. on Circuits, Systems and Video Tech.*, 4(3):228–235, 1994.

[6] C. Le Buhan Jordan and T. Ebrahimi. Study of the effect of lossy shape coding on motion/texture coding and reconstructed VOP quality evaluation. Technical Report ISO/IEC JTC1/SC29/WG11/M1280, Sept. 1996.

[7] C. Le Buhan Jordan and T. Ebrahimi. Progressive polygon encoding of shape contours. In *IEE Conf. on Image Proc. and its Applic.*, 1997. http://ltswww.epfl.ch/~lebuhan/shape.html.

[8] G. Langdon Jr. and J. Rissanen. Compression of black-white images with arithmetic coding. *IEEE Trans. on Comm.*, 29(6):858–867, 1981.

[9] ISO/IEC JTC1/SC29/WG9. Draft international standard 11544 - information technology - coded representation of picture and audio information - progressive bi-level image compression. Technical report, 1992.

[10] R. Mehrotra and J.E. Gary. Similar shape retrieval in shape data mangement. *Computer*, 28(9):57–62, 1995.

[11] K. O'Connell. Object-adaptive vertex-based shape coding method. *IEEE Trans. on Circuits, Systems and Video Tech.*, 7(1):251–255, 1997.

[12] K. O'Connell, P. Gerken, C. Le Buhan, and J. Kim. Error resilient vertex-based (S4h) shape coding description. Technical Report ISO/IEC JTC1/SC29/WG11/M1963, Apr. 1997.

[13] P. van Beek and M. Tekalp. Object-based video coding using forward tracking 2-D mesh layers. In *SPIE*, volume 3024, pages 699–710, 1997.

[14] MPEG Web. http://drogo.cselt.stet.it/mpeg. 1997.