PCS 1997

Coding of Binary Shape in MPEG-4

Jörn Ostermann, AT&T Labs - Research

email: ostermann@research.att.com

ABSTRACT

MPEG-4 Visual, that part of the upcoming MPEG-4 standard describing the coding of natural and synthetic video signals, allows the encoding of video objects using motion, texture and shape information. In this contribution, the five shape coding algorithms (vertex, baseline, MMR, CAE, chroma-keying) evaluated within MPEG-4 are reviewed. The evaluation leading to the selection of a block-based shape coder based on a context-based arithmetic encoder (CAE) / decoder with motion compensation are presented.

1. INTRODUCTION

MPEG-4 Visual will be the first international standard allowing the transmission of shape information for video objects (VO) in a video sequence. Following an object-based approach, MPEG-4 Visual transmits texture, motion, and shape information of one VO within one bit stream. The bit streams of several video objects and accompanying composition information can be multiplexed such that the decoder receives all the information to decode a video sequence containing several objects. This object-based representation allows the decoder to manipulate objects and compose them into new sequences.

Three types of VOs are distinguished: For opaque objects, binary shape information is transmitted. Transparent objects are described by gray-scale alpha maps (8 bits) defining the outline as well as the transparency of an object. For transparent objects, the object outline is encoded using the binary shape coding technique. If the object has constant transparency this transparency value gets transmitted along with optional information for a filter to be applied to the transparency information within 4 pels of the object boundary. For non-constant transparency information, this information is encoded using a hybrid coder with DCT and motion compensation. This encoder is similar to the coder of the luminance signal.

In November 1995, MPEG received 12 proposals for shape coding [1]. Within 6 months, several other methods were proposed - some based on previous proposals and some new. All these proposals were constantly refined by the proposers. In the competitive environment, several algorithms improved their coding efficiency by more than 50% on April 1997. In this contribution, those five binary shape coding techniques (vertex, baseline, MMR, CAE, chroma-keying) are reviewed that still performed competitively in April 1997 [2]. The reasons for selecting a block-based shape coder based on a context-based arithmetic encoder (CAE) / decoder with motion compensation are presented.

2. BINARY SHAPE CODING TECHNIQUES

The goal of shape coding is to encode the shape information of a moving video object. It is assumed that texture and motion information is also transmitted for the video object. Therefore, a shape coder would be allowed to share motion information with the texture coder. The binary shape coder can be grouped into three categories [8]: Bitmap-based shape coders encode whether a pel belongs to the object or not (Sections 2.3, 2.4). Contour-based shape coders describe the outline of an object. In order to retrieve the bitmap of the object shape, these contours are filled with the object label (Sections 2.1, 2.2). Implicit shape coders use a chroma-key to encode the shape information implicitly in the texture information of the VO (Section 2.5).

2.1 Vertex-Based Shape Coding

Vertex-based shape coding codes the outline of the shape. This shape is approximated using a polygon approximation for lossy shape coding. For lossless shape coding, the polygon approximation "degenerates" to a chain code [1][3][9][10].

One interesting feature of a contour based shape coder is the ability to control the shape approximation error by a maximum allowable distance between the original and the approximated shape. Furthermore, the placement of vertices allows an easy control of local variations of the shape approximation error (Figure 1).

The efficiency of this shape coding method depends to the large extent on the encoder. The art of lossy vertex-based shape coding lies in selecting the appropriate vertices for the polygons. The approach chosen by the experimenters starts with finding the longest axis of the shape and using the two endpoints as the initial polygon. For each polygon side, it is now checked whether the approximation lies within a given tolerance. If not, a new vertex is inserted at the point of largest prediction error. Then, for polygon side, it is decided whether it still lies within the allowable shape approximation error (Figure 1).

For temporal prediction, a motion vector can be assigned to each vertex. Furthermore, vertices can be removed and added in order to accommodate temporal shape changes.

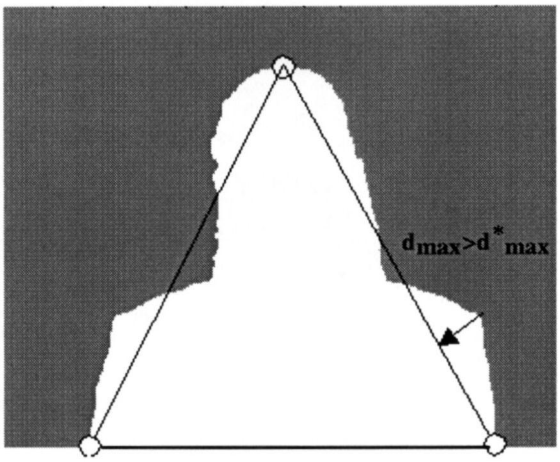

Figure 1: Iterative shape approximation using a polygon. Wherever the distance between the original shape exceed a given threshold d^*_{max}, a new vertex is inserted (from [4]).

2.2 Baseline-based Shape Coding

A baseline-based shape coder also codes the contour of an object. It places the shape into a 2D coordinate system such that the projection of the shape onto the x-axis is longest. The x-axis is called baseline, from which the distance (y-coordinate) between the baseline and a point on the shape outline is coded. The shape contour is sampled counter clockwise. When sampling the shape contour, neighboring contour points usually have increasing or decreasing x-coordinates. Those contour points where the direction changes are called turning points.

In a first step, the addresses of turning points are transmitted. Between turning points, the shape is periodically sampled along the x-axis. The related y-coordinate (distance of the contour point to the baseline) is differentially encoded using a fixed arithmetic encoder.

For temporal prediction, global motion compensation is used.

2.3 Modified Modified Reed Shape Coding

The Modified Modified Reed (MMR) shape coder is a macroblock-based shape coder. In contrast to the previously mentioned shape coders, it directly encodes the bitmap of the shape using a MMR code that is also used in the fax G3 standard.

For rate control, the shape coder allows the shape information of the current macroblock (16x16) to be subsampled by a factor of 2 (8x8 pels) and 4 (4x4 pels). For each macroblock, the subsampling factor (also known as Conversion Ratio CR) is transmitted to the decoder. Generally, the encoder tries to encode the shape with the highest CR that gives a shape representation within the limits of a given approximation error. The decoder decodes the shape and then upsamples the shape of the macroblock according to CR to the desired resolution.

For temporal prediction, the shape of the current macroblock is predicted from the previous frame using motion compensation. Motion vectors are estimated using full search on the binary shape signal and coded as a delta to the motion vector of the texture information.

2.4 Context-based Arithmetic Encoder (CAE) for Shape Coding

This approach differs from the MMR shape coder only in the encoding of the shape information. Temporal prediction, rate control as well as up-/downsampling is performed using the same technique.

In intra mode, a template (context) of 10 pels is used to define the causal context for predicting the shape value of the current pixel. This approach treats the shape as a Markov source with 10 states. The context for encoding the shape is defined across macroblock boundaries. The template extends up to 2 pels to the left and to the top of the macroblock to be coded (Figure 2). For encoding the state transition, a context-based arithmetic encoder is used.

For coding the temporal prediction error, a template of 9 pels is used. 5 pels of the context are defined by the motion compensated shape of the previous frame, 4 pels by the context of the motion compensated prediction error of the current frame.

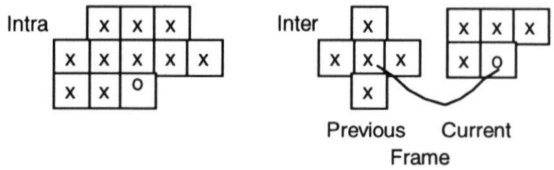

Figure 2: Template for defining the state of the arithmetic encoder/decoder for coding the current pel (o) in intra mode and inter mode (from [5]).

2.5 Shape Coding Using a Chroma-Key

This shape coding technique was inspired from the blue screen technique used in film and TV studios. The object to be coded is placed on a static onecolored background. The color of the background has to be outside of the color space occupied by the texture of the object. Usually, highly saturated colors fulfill this requirement. This image or sequence of images is then encoded using a conventional coder (here MPEG-4 video in full frame mode). To the decoder, the chroma-key (background color) is transmitted. The decoder decodes the images. Pels with a color similar to the chroma-key are considered to be transparent. Otherwise, the pels belong to the object [6].

This implicit shape coding does not allow the extraction of the object shape without decoding the entire frame. Since the shape information is typically carried by the subsampled chroma signal, this technique is not suited for lossless shape coding. Because the shape information is embedded in the texture, the shape coding is lossy as long as there is quantization of the texture. An important advantage of this method is its low computational and algorithmic complexity. The low computational complexity becomes especially apparent if the location of shape boundary macro-blocks is signaled to the decoder such that the shape extraction has to be performed only on this small number of macroblocks.

3. COMPARISON OF BINARY SHAPE CODING TECHNIQUES

The above described shape coding algorithms were thoroughly investigated with respect to their coding efficiency, subjective quality for lossy shape coding, hardware and software complexity and their performance in scaleable shape coders. In areas like error resilience and hardware complexity, we got the advice of experts but no experiments were conducted.

The main decision to be taken was to select a block-based shape coder or a contour-based shape coder. In order to ease this decision, we first selected the best contour-based shape coder and then the best block-based shape coder. Chroma-keying was not considered, because for complex shapes the topology of the shape was not stable enough. For simple objects like head and shoulders chroma keying provides good subjective quality with only minimal color bleeding at object boundaries. However, chroma-keying is not as efficient in terms of coding efficiency as the other shape coder.

As far as contour-based shape coding is concerned, we chose the vertex-based coder. It outperformed the baseline-based coder in coding efficiency for inter mode as well as in terms of computational complexity. Furthermore, we had only one implementation of the baseline-based coder, whereas all the other shape coders had provided two independent implementations. Nevertheless, the baseline-based shape coder provided the best subjective quality in intra mode.

For the block-based shape coding, the CAE coder outperformed the MMR coder in terms of intra and inter shape coding for lossless and lossy coding. Only for large shape approximation errors, we observed a slight advantage of the MMR over CAE [11].

As far as shape coding requirements are concerned, all explicit shape coders are able to provide lossless, subjectively lossless, and lossy coding. The algorithms can be extended to scaleable shape coders, allow bitstream editing, shape only decoding, low-delay applications (with a slight advantage for the block-based shape coders) as well as applications using noisy transmission channels.

Some applications require VOs to be transformed or even deformed. Usually, these application store the contour of an object as a list and do the manipulation on the list. This can be more easily achieved with a vertex-based shape representation since the computation of these lists from a polygon representation is straight forward. When using a bitmap-based shape representation, these lists have to be generated from the bitmap at the decoder. Hence bitmap-based shape representations require more processing for shape manipulations and editing.

Subjective comparisons between the vertex-based coder and the CAE coder showed a slight advantage for the vertex-based method. Generally, outline-based shape coders can provide smooth shape boundaries especially for straight lines. On the other side, a very lossy polygon representation reveals the selected shape coding method on highly curved contours by showing the linear approximation of the polygon. This has to be compared to the blocky shape representation that the bitmap-based shape coders generate when used at low bit rates.

Table 1 compares the CAE and the vertex-based shape coders as done at the Bristol MPEG meeting in April 97. After that meeting, the competitive phase of shape coding ended and we focused on improving the selected CAE shape coder.

4. IMPROVEMENTS ON CAE

Between April 1997 and July 1997 several improvements and functionalities were build on top of the basic CAE shape coder [7].

As mentioned before, the shape within a macroblock can get subsampled and upsampled as a matter of rate control resulting in a blocky shape representation. A new shape adaptive upsampling filter was developed that provides a smooth shape thus offsetting the disadvantage of bitmap-based shape coder compared to outline based shape coder. This filtering is mainly done when coding shapes in intra mode. Therefore this filter improves the subjective quality of shapes coded lossily in intra mode. Because the texture coding is dependent on the coded shape, the same upsampling filter has to be applied in the decoder and the encoder.

5. CONCLUSIONS

Within MPEG-4, several shape coding algorithms were evaluated. In terms of coding efficiency, a block-based context-based arithmetic encoder (CAE) coding the state of a random Markov source and a vertex-based shape coder performed similarly with the CAE coder providing better coding efficiency for lossless shape coding. However, in terms of hardware implementation complexity, the block-based method allows for a regular access to the shape information and therefore was selected as the base technology of MPEG-4 shape coding.

The selection of the shape coding algorithm was based on information available in April 1997. Considering that shape coding is a relatively new area within image processing, more research is required in order to gain a more complete understanding of all the different aspects of shape coding. MPEG-4 will serve as the baseline against which new shape coding techniques will have to compete.

MPEG-4 is the first international standard covering shape coding. It will be finalized in November 1998. The shape coder within MPEG-4 will enable many new functionalities and applications like object-based database access, content-based image and video representation, video editing, efficient storage of movies prior to composition and more.

Table 1: Comparison between the block-based CAE shape coder and the contour-based vertex-based shape coder. Relative statements are always with respect to the other shape coder. Please note that this is a snapshot as of April 1997. Especially results for scalability are not optimized

	CAE	Vertex-based
Coding efficiency: Intra lossless		7.8% lower data rate
Coding efficiency: Inter lossless	20.5% lower data rate	
Coding efficiency: Inter lossy	better at small distortions	better at large distortions
Scalability overhead for 3 layers (layer 3 lossless)	30-50% of lossless one layer rate for predictive coding	no optimized results for predictive coding.
Delay	slightly shorter	
Hardware Implementation Complexity	decoding on chip without random access to external memory	Huffman decoder smaller than arithmetic decoder
Software Implementation Complexity	no optimized coder available, but similar performance for non-optimized code.	

6. REFERENCES

[1] Jörn Ostermann, "Methodologies used for evaluation of video tools and algorithms in MPEG-4", Image Communications, Special Issue on MPEG-4, 1997.

[2] "Core experiments on MPEG-4 video shape coding", Editor: J. Ostermann, ISO/IEC JTC1/SC29/WG11 MPEG97/N1584, March 1998.

[3] 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, S. 381-386.

[4] Jörn Ostermann, "Object-oriented analysis-synthesis coding based on the source model of moving rigid 3D objects", Signal Processing: Image Communication, No. 6. S. 143-161, 1994.

[5] "MPEG-4 video verification model version 8.0", Editor: T. Ebrahimi, ISO/IEC JTC1/SC29/WG11 MPEG97/N1796, July 1997.

[6] T. Chen, C. T. Swain, B. G. Haskell, "Coding of subregions for content-based scalable video", IEEE Trans. On Circuits and Systems for video technology, Vol. 7, pp. 256-260, February 1997.

[7] "Core experiments on MPEG-4 video shape coding", Editor: Jörn Ostermann, ISO/IEC JTC1/SC29/WG11 MPEG97/N1644, July 1997.

[8] "Special Session on 2D Shape Coding", Organizer J. Ostermann, ICIP 97, Santa Barbara, 1997.

[9] K. J. O'Connell, "Object-adaptive vertex-based shape coding method", IEEE Trans. On Circuits and Systems for Video Technology, Vol. 7, No. 1, pp. 251-255, February 1997.

[10] P. Gerken, "Object-based analysis-synthesis coding of image sequences at very low bit rates," IEEE Trans. Circuits and Systems for Video Technology, vol. 4, pp. 228-235, June 1994.

[11] N. Brady, F. Bossen, N. Murphy, "Context-based arithmetic encoding of 2D shape sequences", Special session on shape coding, ICIP 97, Santa Barbara, 1997.