GENERIC, SCALABLE AND EFFICIENT SHAPE CODING FOR VISUAL TEXTURE OBJECTS IN MPEG-4

Shipeng Li[*]
spli@microsoft.com
Microsoft Research, China

Iraj Sodagar
isodagar@sarnoff.com
Sarnoff Corporation

ABSTRACT

This paper presents a generic, scalable, efficient shape coding scheme for scalable object-oriented visual texture. The base-layer coding scheme used is similar to the binary CAE coding scheme adopted in MPEG-4 video. The proposed scheme introduces a new set of generic contexts to efficiently encode (predict) enhanced shape layer based on the lower spatial layer using a context-based arithmetic coder. It is not dependent on any specific sub-sampling filters, thus it can generate the exact shape matching the texture decomposed using any wavelet filters. The proposed shape coding scheme was adopted in MPEG-4 Version 2 Standard to enable the visual texture wavelet coding to be a true spatially-scalable object-based texture coding technique. In addition, when operated in macro-block mode, the proposed scheme provides full backward compatibility with the MPEG-4 scalable shape coding for video objects. A simple solution to solve the chroma shape mismatch is also presented and was adopted in MPEG-4 Version 1 visual texture coding part. The results show that the proposed scalable shape coding scheme also achieves significant better coding efficiency than the non-scalable shape coding and the other competing shape coding scheme.

1. INTRODUCTION

Binary shape coding is an important component of object-based image and video coding in MPEG-4 standard. The shape information or object mask is used to indicate the arbitrary shape of an image or video object and the region in which the texture of the corresponding object is coded. The binary shape information provides object mask with only two values: transparent (pixel outside object) or opaque (pixel inside object).

MPEG-4 Version 1 [2] adopted a non-scalable binary shape coding scheme that efficiently encodes the binary object mask. This scheme was also considered in MPEG-4 Version 1 for object-based texture coding (still texture coding mode). The visual texture coding of MPEG-4 is highly scalable in both resolution and/or quality. If the above non-scalable shape coder is used with texture coding, the decoder must receive, decode and decompose the full resolution shape information before decoding any resolution of the texture. Obviously, this would require more than necessary transmission bandwidth, decoding memory and decoding delay for decoding only lower resolution of the texture.

During the standardization process, few proposals were considered for spatially scalable binary shape coding of MPEG-4 video objects. (CE-S14 [3], scan-interleave [4] and vertex based [5]). However, these proposals were for scalable video coding, where the mask decomposition is a linear sub-sampling process. They did not consider the special characteristics of the mask decomposition schemes used in the arbitrary shape wavelet transform [2][6][7]. Therefore, they cannot build the exact correspondence between spatial layers of mask and spatial layers of texture needed for a true spatial scalability. Besides, [8][9] reported that the current scalable shape coding scheme considered in MPEG-4 Version 2 for video objects generally is less efficient than the non-scalable coder.

For the arbitrary shape wavelet coding scheme adopted in MPEG-4 Version 1 [2], there are two non-linear mask decomposition schemes associated with SA-wavelet [10], which correspond to two different filter types -- odd symmetric filters and even symmetric filters. For the odd symmetric filters case, the shape decomposition is a process similar to linear sub-sampling but with the isolated pixels at the odd position always put in the lowpass subband. For the even symmetric filters case, if there is one pixel within the shape in a 2x2 block in a higher resolution layer, the corresponding pixel must also be within the shape in the lower resolution layer. (i.e. an "OR" decomposition). Apparently, a generic scalable shape coding scheme should be able to handle all the cases mentioned above.

There is a chroma shape mismatch issue associated with scalable shape coding for visual texture. Normally in the non-scalable shape coding case, the chroma shape can be derived from the luma shape by OR sub-sampling. However, in scalable shape coding case, if the wavelet filters used are not even-symmetric and the full-resolution chroma shape is obtained by OR sub-sampling the full-resolution luma shape layer, the chroma shape at each spatial layer obtained by OR sub-sampling the luma shape at the same spatial layer will not match the corresponding chroma shape decomposed from the full-resolution chroma shape. Therefore, we still cannot obtain the chroma shapes at different layers without decoding the whole luma shape.

In [11][12], the authors realized the above problems and proposed patches to the scalable shape coding for video objects in MPEG-4 Version 2 Working Draft (WD) [1] in order to fix the above problems. However, their proposals are not generic enough to handle all the different sub-sampling cases yet to provide backward compatible with video scalable shape coding tool in the Version 2 WD. [12] also addressed the chroma mismatch problem by spending extra bits to encode the differences of the two chroma masks obtained at different spatial layers using the two methods described above.

[*] This work was done while the author was with Sarnoff Corporation.

In this paper, we summarize our contributions to MPEG-4 standard [10][15][16]. First, we present a generic scalable shape coding framework to solve the problem of chroma mismatch without coding the chroma shape differences for arbitrary wavelet filters. Secondly, we propose a new, generic, true spatially-scalable binary shape coding scheme that is not dependent of the mask decomposition schemes. The base layer coding can be the same as the non-scalable shape coding in MPEG-4 version 1 [2]. The enhancement layer coding is based on the context from previous lower resolution layer and from that in the current layer. The proposed scalable shape coding scheme can be operated in either frame mode or block mode. When operated in block mode and with linear sub-sampling mask decomposition, this scalable shape coder is fully backward compatible with the existing scalable shape coder in MPEG-4 Version 2 for video objects. Extensive experiments show that the proposed scalable shape coding scheme offers better overall coding efficiency than the non-scalable coding scheme [2] and the other competing scalable shape coder [12].

The rest of the paper is organized as follows. Section 2 presents a scalable shape coding framework to solve the chroma mismatch problem. Section 3 gives the detail description of the proposed scalable shape coding algorithm. Section 4 provides simulation results and comparisons to other state-of-art shape coders. Section 5 concludes this paper.

2. FRAMEWORK FOR SCALABLE SHAPE CODING IN STILL VISUAL TEXTURE

The usual concept of sub-sampling from luma shape mask to chroma shape mask is OR sub-sampling, since in 4:2:0 color format we'd like to make sure that if a 2x2 luma block has at least one pixel within the luma shape, there will be a chroma component in corresponding position also within the chroma shape. However, this is not necessary true. Virtually any sub-sampling filtering methods (including different sub-sampling positions) can be used to obtain the 4:2:0 chroma from a 4:4:4 images. The possible missing chroma components in 4:2:0 images can be recovered using either interpolation or extrapolation. Therefore it is reasonable to use the shape-adaptive wavelet sub-sampling method as an alternative to obtain the 4:2:0 chroma mask. When a luma component within a luma shape can not find a chroma component in the corresponding position, its chroma component can always be extrapolated.

By using the same wavelet filter as in luma decomposition in the highest resolution layer, we can obtain a 4:2:0 chroma mask. The chroma decomposition in highest layer can then use the same wavelet filter as in luma decomposition in the second highest resolution layer, and so on. Fig. 1 gives the framework of the mask decomposition scheme of the chroma components and their relationship to the luma decomposition. It is clearly shown that no matter what wavelet filters used in luma part, as long as we follow the framework of Fig. 1, we are assured that the chroma mask can always be decomposed from the luma mask at the same layer. Thus the chroma mismatch problem can be solved in such a framework to achieve true shape spatial scalability.

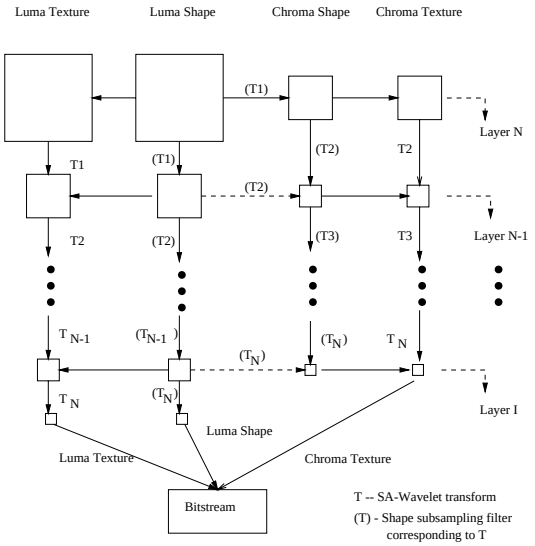

Fig. 1: Decomposition of shape and texture in arbitrarily shaped visual object coding.

3. THE OTF SCALABLE SHAPE CODING SCHEME

A generic, spatially scalable, binary shape coding scheme based on [10][15] is described as follows. Only frame-based coding mode is described since it is best suited for frame-based visual texture coding. For block-based case, interested reader may refer to [1].

Assume the input object mask is a binary rectangular mask (which may be already quantized), i.e., each pixel takes either value 0 (transparent) or 1 (opaque).

Step 1: Encode the block mode.

Assume L-level shape decomposition (which corresponds to the texture decomposition) is to be performed on the input object mask. Divide the mask of the full resolution object into blocks of 2^N x2^N pixels, where $N = L$ if $L >= B$, or $N = B$ if $L < B$. For each 2^N x2^N block, if all its pixels have mask values of 1's, mark this block with symbol AO (all-opaque, mode=1); if all its pixels have mask values of 0's, mark this block with symbol AT (all-transparent, mode=0); otherwise, mark this block with symbol BO (border, mode=2).

Encode the mode for each block in the same way as coding bab_type (intra case) in non-scalable shape coding in the MPEG-4 Version 1 [2].

Step 2: Encode the lowest spatial layer (base layer).

Use the desired SA-DWT filters to decompose the mask into the L+1 levels of spatial layer (1 base layer, L enhancement layers). Denote each layer as Λ_n , where $0<=n<=L$, 0 denotes the full resolution mask layer and Λ_L denotes the base layer (lowest spatial layer). For each layer, all pixels in the AT or AO blocks are not to be coded since their values can be readily filled according to the block mode.

For base layer (lowest spatial layer), according to the raster scan order, encode the mask value (m^L_{ij}) of each pixel in BO blocks

using the same binary context based arithmetic coder as in the intra shape coding scheme in MPEG-4 Version 1 [2].

Step 3: For n from L-1 to 0, encode the mask of each spatial layer Λ_n based on that of the lower spatial layer Λ_{n+1}. (Spatial scalability)

There are two passes in this enhanced layer shape coding. The first pass is to encode half-higher spatial layer $\Lambda_{n+1/2}$ based on previous lower spatial layer Λ_{n+1}. The second pass is to encode the current spatial layer Λ_n based on the half-higher spatial layer $\Lambda_{n+1/2}$. The half-higher spatial layer $\Lambda_{n+1/2}$ is defined as the lower (left) half of the mask after the vertical synthesis from the lower layer Λ_{n+1}. Note that for linear sub-sampling case, this is exactly the same as the scan interleaving (SI) approach in MPEG-4 Version 2 WD. Therefore, the proposed approach provides a unified and more generic scalable shape coding framework for all the cases needed in MPEG-4. We call this generalized scheme OTF (One-Two-Four) prediction-based scalable shape coding scheme to distinguish it from SI scheme. A by-product of this scheme is that it can now support different horizontal and vertical decomposition filters. Fig. 2 illustrates the OTF prediction-based coding process. From our experiments this two-pass prediction coding process provides better coding efficiency.

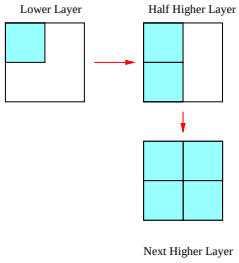

Fig. 2: OTF -- Prediction based coding process in the enhancement layer shape coding.

Pass 1: Coding half-higher spatial layer $\Lambda_{n+1/2}$ from lower spatial layer Λ_{n+1}

For each 1x2 non-overlapped subblock in the raster-scan order in the half-higher spatial layer $\Lambda_{n+1/2}$ in a BO block, the two pixels $T_1=m^{n+1/2}_{(2i)(j)}$ and $T_0=m^{n+1/2}_{(2i+1)(j)}$, where i and j are row and column indices to which this 1x2 subblock corresponds in the lower spatial layer Λ_{n+1}, are coded separately using the context based arithmetic coding schemes described below.

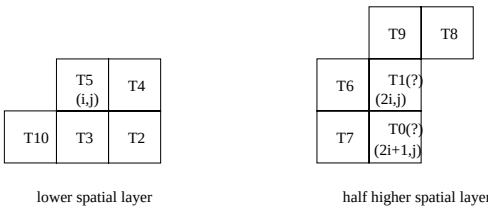

Fig. 3: The related pixel positions and their labels used in the context based arithmetic coding from lower spatial layer to half-higher spatial layer.

Fig. 3 gives the related pixel positions and their labels used in the arithmetic coding contexts. For T_1, the context number is given as follows,

$$context_1=(T_9<<7)|(T_8<<6)|(T_7<<5)|(T_6<<4)|(T_5<<3)|(T_4<<2)|(T_3<<1)|T_2.$$

For T_0, the context number calculation is given by,

$$context_0=(T_1<<7)|(T_{10}<<6)|(T_7<<5)|(T_6<<4)|(T_5<<3)|(T_4<<2)|(T_3<<1)|T_2.$$

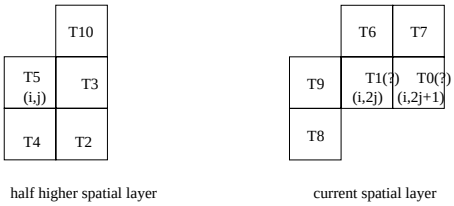

Fig. 4: The related pixel positions and their labels used in the context based arithmetic coding from half-higher spatial layer to current spatial layer.

A context based binary arithmetic coder will be used to encode T_0 and T_1 based on the above contexts.

Pass 2: Coding current spatial Λ_n based on half-higher layer $\Lambda_{n+1/2}$

The second pass can be easily achieved by transposing both the half-higher layer and the current layer mask, and by applying the same procedure of the first pass to encode the current layer mask. Fig. 4 gives the corresponding pixel positions for the contexts used.

This two pass encoding process will be repeated in Step 3 until the full resolution shape is encoded.

Note that in the above contexts, mask values from lower spatial layer Λ_{n+1} or half-higher spatial layer $\Lambda_{n+1/2}$ are used in construction of the context. This is where the implicit prediction occurs.

From the above description, we note that no matter what kind of shape decomposition schemes used, we can always use the same set of contexts and algorithm. The only difference between different shape decomposition schemes is that their context-based probability tables are different. The proposed shape coder can handle arbitrary combinations of different wavelet filters at different spatial layer, even different horizontal and vertical wavelet filters at the same layer.

The above frame-based shape coding scheme can be easily extended to block-based coding case to provide the compatibility with the scalable shape coding for video objects in MPEG-4 Version 2 WD. However, one should pay attention to the method used for the block boundary extension. Detail description of block-based mode operation is beyond the scope of this paper. Fortunately, the block-based mode has been adopted in MPEG-4 Version 2. Interested readers may refer to the standard document [1].

4. EXPERIMENTAL RESULTS

Extensive simulations have been conducted to test the performance of the proposed generic scalable shape coder. Table 1 gives the comparison of the coding efficiency of the proposed (OTF) scalable shape coder versus the scan interleave (SI) method proposed in [12] and the non-scalable shape coder in MPEG-4 version 1 [2]. The test sequences except "random", are from MPEG-4 shape coding test set. The "random" sequence contains 1000 computer generated random shapes. Detail test conditions can be found in [16].

Table 1: Efficiency Test Results

Sequence	Method	Total bits/frame	Percentage over non-scalable
children-logo	OTF	3806	-11.3%
	SI	3516	-2.8%
	non-scal	3418	
children-kids	OTF	1823	19.5%
	SI	1942	14.3%
	non-scal	2266	
clcyaman	OTF	3971	2.7%
	SI	4320	-5.9%
	non-scal	4080	
robot	OTF	2957	2.5%
	SI	3040	-0.02%
	non-scal	3032	
news	OTF	1220	30.0%
	SI	1455	16.5%
	non-scal	1742	
stefan	OTF	663	37.9%
	SI	751	29.6%
	non-scal	1067	
weather	OTF	538	11.2%
	SI	568	6.2%
	non-scal	606	
rain	OTF	128	68.0%
	SI	212	47.0%
	non-scal	400	
random	OTF	1035	36.5%
	SI	1253	23.1%
	non-scal	1629	

Table 1 clearly demonstrates that the proposed OTF scalable shape coder provides significant better coding efficiency than the MPEG-4 non-scalable shape coder. The OTF coder also consistently outperforms the SI coder proposed in [12]. The only exception is the children-logo sequence. This is because the shapes in this sequence are some text fonts, which is not the proposed scalable shape coder targeted for.

5. SUMMARY

In this paper, we present a scalable shape coding scheme that solves the chroma mismatch problem and provides true spatially scalable shape coding. The proposed scheme doesn't require coding mode switching in the enhanced layer. It provides backward compatibility with MPEG-4 Version 1. Moreover, the frame-based coding scheme fits MPEG-4 visual texture coding infrastructure. It provides flexible scalability and significant better coding efficiency than the non-scalable shape coding scheme of MPEG-4 Version 1 and consistently better coding efficiency than scan-interleave (SI) coding scheme. The proposed scheme can also be operated in block-based mode to provide backward compatibility with MPEG-4 Version 2 video shape coding tool. The proposed coding scheme has been adopted in MPEG-4 version 2 WD [1] to form part of the scalable shape coding for visual texture coding.

6. REFERENCES

[1] ISO/IEC 14496-2, "Information Technology – Generic Coding of Audio-Visual Objects: Part II--Video", Working Draft Version 2 Rev. 6.1, ISO/IEC JTC1/SC29/WG11 M4583, Seoul, Mar. 1999.

[2] ISO/IEC 14496-2, "Information Technology – Generic Coding of Audio-Visual Objects: Part II--Video", Final Draft of International Standard, ISO/IEC JTC1/SC29/WG11 N2502a, Atlantic City, Oct. 1998.

[3] J. Ostermann (editor), "Core Experiment on MPEG-4 Video Shape Coding", ISO/IEC JTC1/SC29/WG11 N1644, Bristol, July, 1997.

[4] D.-S. Cho, et. al., "Description of arbitrary shaped spatial scalability", ISO/IEC JTC1/SC29/WG11 M3091, San Jose, Feb. 1998.

[5] J. Chung, D. Shin and J. Moon, "Technical Description of Vertex-based Binary Shape Coding", ISO/IEC JTC1/SC29/WG11, M3409, March, 1998.

[6] S. Li, et. al., "Shape Adaptie Wavelet Coding," Proceedings of the IEEE International Symposium on Circuits and Systems, vol. 5, pp. 281-284, ISCAS'98, Monterey, California, May 1998.

[7] S. Li and W. Li, "Shape Adaptive Discrete Wavelet Transforms for Arbitrarily Shaped Visual Object Coding", submitted to IEEE Transaction on Circuits and Systems for Video Technology (July, 1997).

[8] T. Suzuki, T. Nagumo and Y. Yagasaki, "The Results of Arbitrary shaped scalability", ISO/IEC JTC1/SC29/WG11, M3136, San Jose, Feb. 1998.

[9] D.-S. Cho, et.al., "Results of Arbitrarily Shaped Spatial Scalability", ISO/IEC JTC1/SC29/WG11, M3002, San Jose, Feb. 1998.

[10] S. Li, I. Sodagar and H.-J. Lee, "Proposal for a Generic Scalable Shape Coding Scheme", ISO/IEC JTC1/SC29/ WG11, M3787, Dublin, July 1998.

[11] Y. Ueda and Z. Wu, "Scalable Shape Coding for Still Texture", ISO/IEC JTC1/SC29/WG11, M3621, Dublin, July 1998.

[12] S. H. Son, et. al., "Description of Mini CE on Scalable Shape Coding for Visual Texture Coding using version 2 WD tool," ISO/IEC JTC1/SC29/WG11, M4041, Atlantic City, Oct., 1998.

[13] J.-D. Kim, et. al., "Results of Mini CE on Scalable Shape Coding for Still Texture," ISO/IEC JTC1/SC29/WG11, M4001, Atlantic City, Oct. 1998.

[14] S.-H. Son, D.-S. Cho and J.-S. Shin, "Experimental Results on Scalable Shape Coding for Visual Texture Coding using version 2 WD tool", ISO/IEC JTC1/SC29/WG11, M4041, Atlantic City, Oct., 1998.

[15] S. Li, I. Sodagar, H.-J. Lee and Y.-Q. Zhang, "Description of Mini Core Experiment on Scalable Shape Coding for Visual Texture Coding", ISO/IEC JTC1/SC29/WG11 M3891, Atlantic City, Oct. 1998.

[16] S. Li, I. Sodagar, H.-J. Lee and Y.-Q. Zhang, "Report on core-experiment of scalable shape coding for VTC", ISO/IEC JTC1/SC29/WG11, M4152, Atlantic City, Oct. 1998.