FULLY SCALABLE TEXTURE CODING OF ARBITRARILY SHAPED VIDEO OBJECTS

Habibollah Danyali and Alfred Mertins

School of Electrical, Computer and Telecommunications Engineering
University of Wollongong, Wollongong, NSW 2522, Australia
Email: {hd04, mertins}@uow.edu.au

ABSTRACT

This paper presents a fully scalable texture coding algorithm for arbitrarily shaped video object based on the 3D-SPIHT algorithm. The proposed algorithm, called Fully Scalable Object-based 3D-SPIHT (FSOB-3DSPIHT), modifies the 3D-SPIHT algorithm to code video objects with arbitrary shape and adds spatial and temporal scalability features to it. It keeps important features of the original 3D-SPIHT coder such as compression efficiency, full embeddedness, and rate scalability. The full scalability feature of the modified algorithm is achieved through the introduction of multiple resolution dependent lists for the sorting stage of the algorithm. The idea of bitstream transcoding without decoding to obtain different bitstreams for various spatial and temporal resolutions and bit rates is completely supported by the algorithm.

1. INTRODUCTION

Object-based video coding deals with a video scene as a composition of arbitrarily shaped video objects and encodes each object individually rather than considering the whole scene as a rectangular frame. It enables new types of applications and functionalities and has become one of the most important features of the new generation of video coding standards such as MPEG-4. In the context of an explosive growth of multimedia applications, and consequently multimedia transmission especially over a heterogenous networks such as the Internet, and the distinct role of video in multimedia, nowadays there is a great demand for a fully scalable object-based video coding system. Such a coding system provides a bitstream that consists of embedded parts to offer increasingly better signal-to-noise ratio (SNR) or/and greater spatial resolution or/and higher frame rate for each object in the scene. Different parts of such a bitstream can be selected and decoded by a scalable decoder to meet certain requirements. Moreover, different types of decoders with different complexity and access bandwidth can coexist.

Due to the multiresolution signal representation offered by the wavelet transform, wavelet based coding schemes have a great potential to support SNR, spatial and temporal scalability. This is especially the case for 3D wavelet decompositions like the ones in [1–3], where a 1D discrete wavelet transform (DWT) with or without motion compensation is applied along the temporal axis, followed by a 2D DWT in the spatial domain (or vice versa). Modifications of the DWT, called shape adaptive DWTs (SA-DWTs), like the one in [4], enable wavelet based coding algorithms to be extended for coding of arbitrarily shaped video objects. However, the SA-DWT itself is not the only issue. It is particularly important how the various spatio-temporal wavelet bands are encoded. The Set Partitioning in Hierarchical Trees (SPIHT) algorithm [5] is one of the best wavelet-based coding algorithms for still images. It very efficiently exploits the self similarity between different levels in the wavelet pyramid and provides an embedded bitstream which supports full SNR scalability. The excellent rate-distortion performance and scalable nature of SPIHT for still images make it an attractive coding strategy also for video coding. A 3D extension of SPIHT for video coding has been proposed by Kim and Pearlman in [1, 2]. They applied a 3D wavelet transform to a group of video frames (GOF) and coded the wavelet coefficients by the 3D-SPIHT algorithm. As reported in [1], even without motion estimation and compensation this method performs (in test with SIF (352×240) monochrome 30HZ sequences) measurably and visually better than MPEG-2, which employs complicated means of motion estimation and compensation. Moreover by employing a SA-DWT scheme, SPIHT-based image and video coding algorithms are easily extendable for coding of arbitrarily shaped still and video objects coding [6–8]. Although the 3D-SPIHT bitstream is tailored for full SNR scalability and provides a progressive (by quality) video information, it does not explicitly support spatial and temporal scalability and does not provide a bitstream that can be reordered according to desired spatio-temporal resolutions and fidelity.

In this paper, a fully scalable texture coding method for arbitrarily shaped video objects is presented. The basis for our new algorithm is the 3D-SPIHT algorithm of [2], which we extend with multiple resolution-dependent lists that al-

low us to obtain all spatial, temporal and SNR scalability features together. This new algorithm is related to our 2D algorithm for still images in [9], however, it is extended for coding of arbitrarily shaped video objects and uses different lists and results in a differently structured bitstream. The encoded bitstream can be easily reordered to achieve different levels of spatial resolution, frame rate and quality requested by the decoder. A distinct feature of the presented coder is that the reordered bitstreams for different spatio-temporal resolutions, which are obtained after parsing the main bitstream, are completely fine granular and can be truncated at any point to obtain the best reconstructed sequence at the desired spatio-temporal resolution and bit rate. In other words, our modified SPIHT algorithm provides spatial and temporal scalability without sacrificing compression efficiency and SNR scalability in any way.

The rest of this paper is organized as follow. Section 2 describes our modified algorithm, called FSOB-3DSPIHT. Section 3 gives an overview of the bitstream formation and parsing process. In Section 4, simulation details are explained and some experimental results are shown, and finally, Section 5 concludes the paper.

2. THE FULLY SCALABLE OBJECT-BASED 3D-SPIHT (FSOB-3DSPIHT) ALGORITHM

The 3D-SPIHT algorithm of [2] considers sets of coefficients that are related through a parent-offspring dependency like the one depicted in Fig. 1. In its bitplane coding process, the algorithm deals with the wavelet coefficients as either a root of an insignificant set, an individual insignificant pixel, or a significant pixel. It sorts these coefficients in three ordered lists: the list of insignificant sets (LIS), the list of insignificant pixels (LIP), and the list of significant pixels (LSP). The main concept of the algorithm is managing these lists in order to efficiently extract insignificant sets in a hierarchical structure and identify significant coefficients, which is the core of its high compression performance.

The 3D wavelet decomposition of a GOF as shown in Fig. 1 provides a multiresolution structure that consists of different spatio-temporal subbands that can be coded separately by a scalable encoder to provide various spatial and temporal scalabilities. In general, by applying N_t levels of 1D temporal decomposition and N_s levels of 2D spatial decomposition, at most $N_s + 1$ levels of spatial resolution and $N_t + 1$ levels of temporal scalability are achievable. The total number of possible spatio-temporal resolutions in this case is $(N_s + 1) \times (N_t + 1)$. To distinguish between different resolution levels, we denote the lowest spatial resolution level as level $N_s + 1$ and the lowest temporal resolution level as level $N_t + 1$. The full spatial and temporal resolution (the original sequence) then becomes resolution level 1 in both

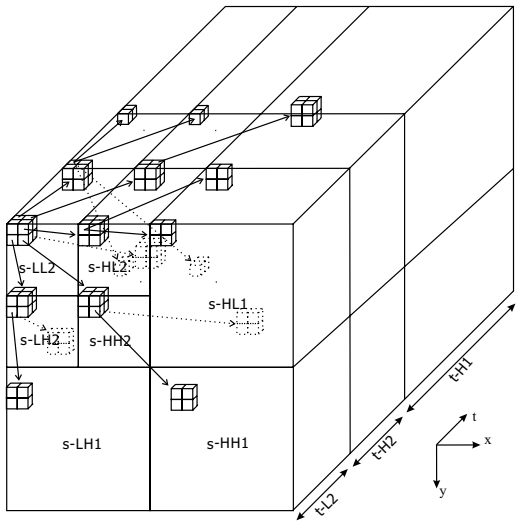

Fig. 1. A 3D wavelet decomposition of a GOF with 2 levels of both temporal and spatial decomposition (21 spatio-temporal bands).

spatial and temporal directions. An algorithm that provides full spatial and temporal scalability would encode the different resolution subbands separately, allowing a transcoder or a decoder to directly access the data needed to reconstruct a desired spatial and temporal resolution. The original 3D-SPIHT algorithm of [2], however, sorts the wavelet coefficients in such a way that the output bitstream contains mixed information of all subbands in no particular order, making it impossible to transcode the bitstream without decoding it.

Changes from 3D-SPIHT to FSOB-3DSPIHT:

The FSOB-3DSPIHT algorithm proposed in this paper solves the spatial and temporal scalability problem through the introduction of multiple resolution-dependent lists and a resolution-dependent sorting pass. For each spatio-temporal resolution level we define a set of LIP, LSP and LIS lists,[1] therefore we have $\text{LIP}_{s,t}$, $\text{LSP}_{s,t}$, and $\text{LIS}_{s,t}$ for $s = s_{max}, s_{max} - 1, \ldots, 1$ and $t = t_{max}, t_{max} - 1, \ldots, 1$ where s_{max} and t_{max} are the maximum number of spatial and temporal resolution levels respectively, supported by the encoder. The parent-offspring relation in our algorithm is the same as with 3D-SPIHT, but we only consider and process those coefficients and sets which belong to the decomposed object, similar to the SA-SPIHT algorithms in [7, 8]. Like 3D-SPIHT, our encoder transmits bitplane by bitplane, but it uses multiple lists for handling different resolution levels. In each bitplane, the FS-3DSPIHT coder starts encoding from the maximum resolution level (s_{max}) and proceeds to the lowest level (level 1). During the resolution-dependent sorting pass for the lists that belong to levels s and t, the

[1] For the original definitions of these lists see [5]

algorithm first does the sorting for the coefficients in the $\text{LIP}_{s,t}$, in the same way as 3D-SPIHT, to find and output significance bits for all list entries and then processes the $\text{LIS}_{s,t}$. During processing the $\text{LIS}_{s,t}$, sets that lie outside the resolution level (s,t) are moved to their appropriate LIS. After the algorithm has finished the sorting and refinement passes for resolution level (s,t) it will do the same procedure for all other remaining lists related to other spatial and temporal levels. For FSOB-3DSPIHT, the total number of bits belonging to a particular bitplane is the same as for OB-3DSPIHT, but FSOB-3DSPIHT arranges them in the bitstream according to their spatial and temporal resolution dependency.

Note that the total storage requirement for the $\text{LIP}_{s,t}$, $\text{LSP}_{s,t}$, and $\text{LIS}_{s,t}$ for all resolutions is the same as for the LIS, LIP, and LSP used by the 3D-SPIHT algorithm.

3. BITSTREAM FORMATION AND PARSING

Fig. 2 shows the bitstream structure generated by the encoder. The bitstream is divided into different parts according to the different bitplanes. Inside each part, the bits that belong to the different spatial resolution levels are separable, and similarly, inside each spatial resolution level the bits that belong to different temporal resolutions come in order. To support bitstream parsing by an image server/transcoder, some markers are required to be put into the bitstream to identify the parts of the bitstream that belong to the different spatial and temporal resolution levels and bitplanes.

The encoder needs to encode the video object texture only once at a high bit-rate. Different bitstreams for different spatial and temporal resolutions and fidelity can be easily generated from the encoded bitstream by selecting the related resolution parts. The transcoding process is a simple reordering of the original bitstream parts and can be carried out by a video server or a transcoder, without the need to decode any parts of the bitstream. For example, to provide a bitstream for resolution level (s,t), only the spatial parts that belong to the spatial resolution levels greater or equal to s are kept, and in each selected spatial part only the temporal parts that fall inside the requested temporal resolution level are kept, and all other parts are removed. Note that all marker information for identifying the individual bitplanes and resolution levels is only used by the parser and does not need to be sent to the decoder.

The decoder required for decoding the reordered bitstream exactly follows the encoder, similar to the original 3D-SPIHT algorithm. It needs to keep track of the various lists only for spatial and temporal resolution levels greater or equal to the required one. Thus, the proposed algorithm naturally provides computational scalability as well.

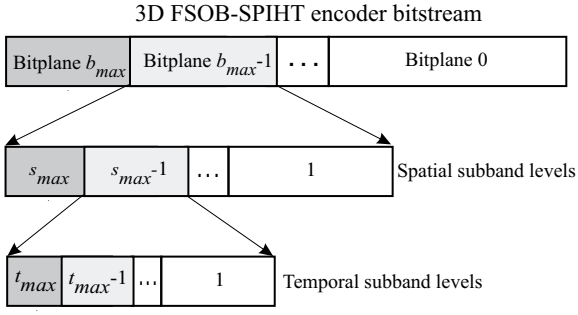

Fig. 2. Structure of the 3D FSOB-SPIHT encoder bitstream which is made up of different quality,spatial and temporal resolution parts for texture of each video object.

4. SIMULATION DETAILS AND EXPERIMENTAL RESULTS

In this section we first give some information about our simulation setup and the tests performed to show the FSOB-3DSPIHT full scalability features, and then we present some objective PSNR results for different spatial and temporal resolutions and bit rates. Reference frames for lower resolutions were defined by taking the lowest frequency subband frames after applying appropriate levels of temporal and spatial wavelet decomposition to the original sequence.

A MPEG-4 test sequence, Akiyo CIF, was selected for the test. The original frame rate of the sequence is 30 frames per second and only the foreground (Akiyo object) is considered for coding. On the encoder side, the input video object sequence is divided into separate groups of object frames (GOF) of size 16. Three levels of 1D object-based temporal filtering by Haar filter is first applied to each GOF, followed by four levels of 2D spatial SA-DWT by 9/7-tap filters [10] with symmetric extension at the boundary of the objects. The FSOB-3DSPIHT encoder then progressively encodes the decomposed GOF from the lowest (coarsest) spatio-temporal band to the highest band.

After encoding 240 frames (15 GOFs) of the texture of the video object sequence, the FSOB-3DSPIHT bitstream was fed into a transcoder to produce progressive (by quality) bitstreams for different spatial and temporal resolutions. The user has the choice to decode any combined resolution from 5 different spatial and 4 temporal resolutions at any desirable bit rate. On the decoder side, the decoder receives a reordered bitstream for each GOF. The FSOB-3DSPIHT decoder uses this bitstream to decode only the required spatio-temporal bands. The inverse spatial and temporal wavelet decomposition is then applied to the decoded spatio-temporal subbands to create a reconstructed version of the video object sequence in the requested resolutions.

Table 1 shows the mean PSNR results obtained for the

Table 1. Y-PSNR results for 240 frames of 30HZ CIF Akiyo foreground at different spatial and temporal resolutions and bit rates obtained by a FSOB-3DSPIHT decoder.

Spatial resolution	Temporal resolution	Rate (kbps)	Akiyo Object Mean Y-PSNR (dB)
CIF	30	128	34.14
CIF	30	80	31.72
CIF	15	64	31.23
CIF	15	48	29.57
QCIF	15	64	35.15
QCIF	15	48	33.36
QCIF	7.5	48	34.27
QCIF	7.5	32	31.31
QCIF	7.5	20	28.43
$\frac{1}{4}$ QCIF	7.5	32	39.60
$\frac{1}{4}$ QCIF	7.5	20	34.47
$\frac{1}{4}$ QCIF	3.75	20	37.65
$\frac{1}{4}$ QCIF	3.75	16	35.22

luminance components of the decoded object sequence at some of the spatio-temporal resolutions and bit rates. As the PSNR values in Table 1 show, high-SNR bitstreams can be easily generated at low rates with reduced spatio temporal resolution. Note that the main objective of this paper is providing full scalability for video object texture coding, so we have not dealt with motion compensation during temporal filtering, however, by employing a motion compensated temporal filtering scheme, without any modification to the FSOB-3DSPIHT algorithm, better PSNR results at the same rates can be expected.

5. CONCLUSIONS

We have presented a fully scalable object-based 3D-SPIHT (FSOB-3DSPIHT) algorithm for video object texture coding. All types of combined spatial, temporal and SNR scalability are supported by FSOB-3DSPIHT. The interesting features of the original 3D-SPIHT algorithm such as high compression efficiency, embeddedness, and very fine granularity of the bitstream are kept. The encoded bitstream can be simply reordered (transcoded) without the need of decoding to obtain different bitstreams tailored for the spatial and temporal resolutions and bit rate requested by the decoder. The proposed fully scalable video object codec is a good candidate for multimedia applications such as video information storage and retrieval systems, and video transmission especially over heterogenous networks where a wide variety of users needs to be differently serviced according to their network access and data processing capabilities.

6. REFERENCES

[1] B.-J.Kim and W. A. Pearlman, "An embedded video coder using three-dimensional set partitioning in hierarchical trees (SPIHT)," in *proc. IEEE Data Compression Conf.*, Mar. 1997, pp. 251–260.

[2] B.-J. Kim, Z. Xiong, and W. A. Pearlman, "Low bit-rate scalable video coding with 3-d set partitioning in hierarchical trees (3-D SPIHT)," *IEEE Trans. Circ. and Syst. for Video Technology*, vol. 10, no. 8, pp. 1374–1387, Dec. 2000.

[3] S.-T. Hsiang and J. W. Woods, "Embedded video coding using invertible motion compensated 3-D subband/wavelet filter bank," *Signal Processing: Image Communication*, vol. 16, no. 8, pp. 705–724, May 2001.

[4] Shipeng Li and Weiping Li, "Shape-adaptive discrete wavelet transforms for arbitrarily shaped visual object coding," *IEEE Trans. Circ. and Syst. for Video Technology*, vol. 10, no. 5, pp. 725–743, Aug. 2000.

[5] A. Said and W. A. Pearlman, "A new, fast and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. Circ. and Syst. for Video Technology*, vol. 6, pp. 243–250, June 1996.

[6] G. Xing, J. Li, S. Li, and Y.-Q. Zhang, "Arbitrarily shaped video-object coding by wavelet," *IEEE Trans. Circ. and Syst. for Video Technology*, vol. 11, no. 10, pp. 1135–1139, Oct. 2001.

[7] G. Minami, Z. Xiong, A. Wang, and S. Mehrotra, "3-d wavelet coding of video with arbitrary regions of support," *IEEE Trans. Circ. and Syst. for Video Technology*, vol. 11, no. 9, pp. 1063–1068, Sept. 2001.

[8] Y. Yuan and C. W. Chan, "Coding of arbitrarily shaped video objects based on SPIHT," *IEE Electronics Letters*, vol. 36, no. 13, pp. 1105–1106, Jun. 2000.

[9] H. Danyali and A. Mertins, "Highly scalable image compression based on SPIHT for network applications," in *Proc. IEEE Int. Conf. Image Processing*, Rochester, NY, USA, Sept. 2002.

[10] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, "Image coding using wavelet transform," *IEEE Trans. Image Processing*, vol. 1, pp. 205–220, Apr. 1992.