OBJECT-BASED IMAGE COMPRESSION WITH SIMULTANEOUS SPATIAL AND SNR SCALABILITY SUPPORT FOR MULTICASTING OVER HETEROGENEOUS NETWORKS

Habibollah Danyali[1] *and Alfred Mertins*[2]

[1]School of Electrical, Computer and Telecommunications Engineering
University of Wollongong, Wollongong, NSW 2522, Australia
habib@titr.uow.edu.au
[2]Faculty of Mathematics and Science, Institute of Physics
University of Oldenburg, 26111 Oldenburg, Germany
alfred.mertins@uni-oldenburg.de

ABSTRACT

This paper proposes an image compression algorithm for texture coding of arbitrarily shaped image objects which supports spatial and SNR scalability features simultaneously. The proposed algorithm, is based on the Set Partitioning in Hierarchical Trees (SPIHT) algorithm and called Fully Scalable Object-Based SPIHT (FSOB-SPIHT). It adds spatial scalability features to the SPIHT bitstream through the introduction of multiple resolution dependent lists for the sorting stage of the algorithm. It keeps important features of the original SPIHT coder such as compression efficiency, and full SNR (rate) scalability. The idea of bitstream transcoding without decoding, to obtain different bitstreams for various spatial resolutions and bit rates all from a single bitstream, is completely supported by the algorithm. Therefore the proposed algorithm efficiently facilitates multicasting visual information over heterogeneous networks.

1. INTRODUCTION

Enabling coding of arbitrarily shaped objects and providing rate (fidelity) and resolution (spatial) scalability are two necessary features for a new generation of image coding systems. Object-based image coding systems consider an image as a composition of different arbitrarily shaped objects and encode each object individually rather than considering the whole picture as a rectangular matrix of pixels. A scalable coding scheme provides a bitstream that consists of embedded parts to offer increasingly better signal-to-noise ratio (SNR) or/and greater spatial resolution for each object in the scene. Such a scalable object-based image coding system could provide a very convenient coding scheme for progressive transmission of visual information over a heterogenous networks such as the Internet.

The multiresolution image representation provided by the two dimensional discrete wavelet transform (DWT) gives wavelet based image coding schemes a great potential to support both SNR, and spatial scalability. Modifications of the DWT, called shape adaptive DWTs (SA-DWTs), like the one in [1], enable wavelet based coding algorithms to be extended for coding of arbitrarily shaped objects. In recent years, a family of very efficient zerotree based wavelet image coders has been developed and emerged as one of the most promising techniques to meet the challenges for image coding. Based on the idea of grouping wavelet coefficients at different scales and predicting zero coefficients across scales, Shapiro [2] introduced the Embedded Zero-tree Wavelet (EZW) coding scheme. The EZW was improved by Said and Pearlman [3] in their work called Set Partitioning in Hierarchical Trees (SPIHT) algorithm. SPIHT provides very efficient compression and supports progressive image transmission and is considered as a benchmark for the state-of-the-art image coding algorithms. By employing a SA-DWT scheme, the SPIHT-based image coding algorithm is easily extendable to coding of arbitrarily shaped still and video objects [4, 5]. Although the SPIHT bitstream is tailored for full SNR scalability, it does not explicitly support spatial scalability and does not provide a bitstream that can be reordered according to desired resolutions and fidelity.

In [6, 7] we extended the SPIHT with multiple resolution-dependent lists and a resolution-dependent sorting pass to be able to combine spatial and SNR scalability features together for rectangular images. In this paper, we extend the method of [7] to texture coding of arbitrarily shaped still objects. We only consider texture coding here and assume that the shape information is transmitted in a multiresolution manner as side

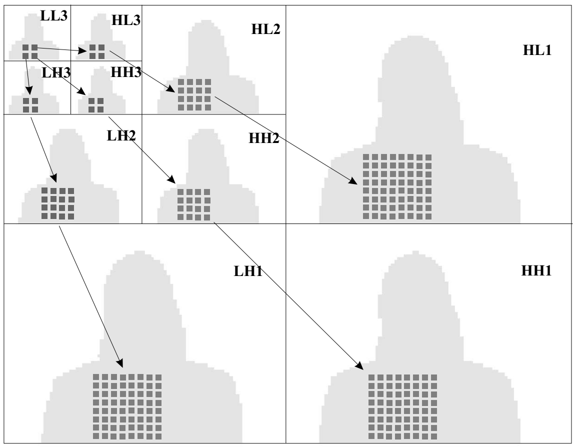

Fig. 1. Orientation of SPIHT sets across wavelet subbands of an object.

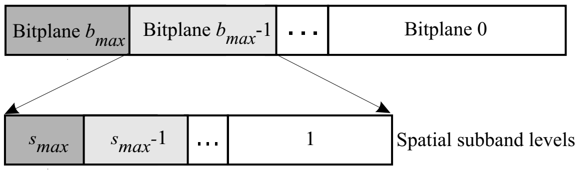

Fig. 2. Structure of the FSOB-SPIHT encoder bitstream which is made up of different quality and spatial resolution parts for texture of each object.

information. The new algorithm, called fully scalable object-based SPIHT (FSOB-SPIHT), adds the spatial scalability feature to the SPIHT bitstream without sacrificing compression efficiency and SNR scalability in any way. The FSOB-SPIHT bitstream can be easily reordered to achieve different levels of spatial resolution and quality requested by the decoder.

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

The SPIHT [3] algorithm, considers groups of coefficients in different scales of the wavelet image pyramid together as sets through a parent-offspring dependency like the one depicted in Fig. 1. At the beginning, the roots of the sets are located at the lowest frequency subband of the wavelet pyramid. It follows a bitplane-manner coding, and in each 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.

In general, by applying N_s levels of 2D DWT to an image, at most $N_s + 1$ levels of spatial resolution are achievable. To distinguish between different resolution levels, we denote the lowest spatial resolution level as level $N_s + 1$. The full spatial resolution (the original image resolution) then becomes resolution level 1. The three subbands (HL$_k$, LH$_k$, HH$_k$) that need to be added to increase the spatial resolution from level $k + 1$ to level k, are referred to as level k resolution subbands. An algorithm that provides full spatial scalability would encode the different levels of resolution subbands separately. The original SPIHT algorithm, however, sorts the wavelet coefficients in such a way that the output bitstream contains mixed information of all subbands in no particular order.

The FSOB-SPIHT algorithm proposed in this paper solves the spatial scalability problem through the introduction of multiple resolution-dependent lists and a resolution-dependent sorting pass. For each spatial resolution level we define a set of LIP, LSP and LIS lists, therefore we have LIP$_s$, LSP$_s$, and LIS$_s$ for $s = s_{max}, s_{max} - 1, \ldots, 1$ where s_{max} is the maximum number of spatial resolution levels, supported by the encoder. The parent-offspring relationship in our algorithm is the same as with SPIHT, but we only consider and process those coefficients and sets which belong to the decomposed object (see Fig. 1), similar to the modified algorithms in [4, 5]. In each bitplane coding process, the FSOB-SPIHT 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 level s, the algorithm first does the sorting for the coefficients in the LIP$_s$, in the same way as SPIHT, to find and output significance bits for all list entries and then processes the LIS$_s$. During processing the LIS$_s$, sets that lie outside the resolution level s are moved to their appropriate LIS$_{s-1}$. After the algorithm has finished the sorting and refinement passes for resolution level s it will do the same procedure for the next finer resolution level until all spatial resolution levels are finished. The total number of bits belonging to a particular bitplane is the same as for an object-based modification of SPIHT like the methods in [4, 5], but FSOB-SPIHT arranges them in the bitstream according to their spatial resolution dependency.

Note that the total storage requirement for the LIP$_s$, LSP$_s$, and LIS$_s$ for all resolutions is the same as for the LIS, LIP, and LSP used by the SPIHT algorithm.

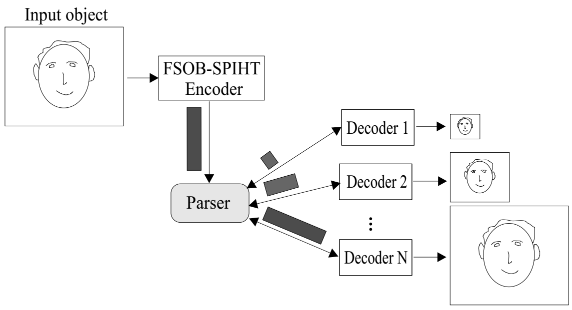

Fig. 3. An illustration of parsing a single FSOB-SPIHT bitstream for decoding at different qualities and resolutions.

3. BITSTREAM FORMATION AND PARSING

The structure of the bitstream generated by the encoder is shown in Fig. 2. It is made up of different parts belonging to the different bitplanes. Inside each bitplane part, the bits that belong to the different spatial resolution levels come in order. To support bitstream parsing by an image server/transcoder, some markers are required to be put into the bitstream to separate the parts of the bitstream that belong to the different spatial resolution levels in each bitplanes.

The encoder needs to encode the object texture only once at a high bit-rate. Different bitstreams for different spatial resolutions and fidelity can be easily generated from the encoded bitstream by selecting the related resolution parts. Fig. 3 illustrate an example of multicasting images for different users with different capabilities. The parsing process is a simple reordering of the original bitstream and can be carried out by the image server that stores the encoded bitstreams or by an individual parser as a simple part of an active network. The parser does not need to decode any parts of the bitstream. For example, to provide a bitstream for resolution level s, in each bitplane part, only the spatial parts that belong to the spatial resolution levels greater or equal to s are kept and all other parts are removed. As a distinct feature, the reordered bitstreams for each spatial resolution are completely rate-embedded (fine granular at bit level) which means it can be truncated at any point to obtain the best reconstruction of the object at that bit rate. 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 bitstreams exactly follows the encoder, similar to the original SPIHT algorithm. It needs to keep track of the various lists only for spatial resolution levels greater or equal to the required one. Thus, the proposed algorithm naturally provides computational scalability as well.

4. SIMULATION DETAILS AND EXPERIMENTAL RESULTS

The FSOB-SPIHT encoder and decoder were fully software implemented. The respective first frames of two MPEG-4 CIF color (in YUV format) test sequences, Akiyo and Foreman, were selected for the test. Only the foreground objects of the test images was considered for coding. The shape/segmentation mask for these test sequences are supplied by MPEG. On the encoder side, four levels of 2D SA-DWT by 9/7-tap filters [8] with symmetric extension at the boundary of the objects were first applied to the objects, then the FSOB-SPIHT encoder was set to progressively encode the decomposed objects from the maximum required bitplane to bitplane zero with five levels of spatial scalability support.

After encoding, the FSOB-SPIHT bitstream was fed into a parser to produce progressive (by quality) bitstreams for different spatial resolutions requested by a decoder. The FSOB-SPIHT decoder uses the reordered bitstream to decode only the required spatial subbands which are necessary for reconstructing the requested spatial resolution of the object. The inverse SA-DWT is then applied to the decoded spatial subbands to create a reconstructed version of the object in the requested resolutions. Reference frames for lower resolutions were defined by taking the lowest frequency subband frames after applying appropriate levels of SA-DWT to the original objects, and the fidelity was measured by the peak signal-to-noise ratio (PSNR). The bit rates for all levels were calculated according to the number of pixels in the foreground of the original full size image.

Table 1 compares PSNR results of FSOB-SPIHT and OB-SPIHT obtained for the luminance (Y) components of the test objects at various spatial resolutions and bit rates. The OB-SPIHT results refer to our implementation of the original SPIHT for object-based coding, similar to [4, 5]. The results for spatial resolution level 1 clearly show that the FSOB-SPIHT does not sacrifice the compression efficiency of the OB-SPIHT. The small deviation between FSOB-SPIHT and OB-SPIHT is due to a different order of coefficients within the bitstreams. For resolution levels 2 and 3, as the results show, the performance of FSOB-SPIHT is much better than OB-SPIHT. For these resolutions, the FSOB-SPIHT decoder was able to decode bitstreams that were properly tailored by the parser for the given resolution level. For the OB-SPIHT case, there is no possibility of reordering the bitstreams for

Table 1. PSNR results for luminance component of foreground objects of the first frame of Akiyo and Foreman MPEG-4 CIF color sequences at different spatial resolutions and bit rates.

Object	Spatial resolution	Rate (bpp)	OB-SPIHT PSNR (dB)	FSOB-SPIHT PSNR (dB)
Akiyo	Full	0.25	29.31	29.27
		0.5	33.43	33.31
		1	39.33	39.18
	Half	0.125	28.10	28.10
		0.25	32.26	33.21
		0.5	37.06	40.15
		1	42.83	51.17
	Quarter	0.0625	27.83	28.76
		0.125	32.06	35.11
		0.25	36.68	46.26
		0.5	41.97	69.83
Foreman	Full	0.25	35.51	35.66
		0.5	40.05	39.87
		1	45.27	45.27
	Half	0.125	32.84	33.23
		0.25	37.88	38.82
		0.5	43.25	46.17
		1	49.73	57.03
	Quarter	0.0625	30.71	31.46
		0.125	36.10	38.91
		0.25	41.92	50.47
		0.5	47.27	69.34

the requested resolution level, and therefore, the bitstreams were first decoded to obtain the coefficients in the wavelet pyramid for all subbands at the given bit rate, and then the requested spatial resolution was reconstructed by applying the inverse SA-DWT only to the required subbands for that resolution level. The reason for the improved performance of FSOB-SPIHT over OB-SPIHT for resolution levels larger than one is clear. In the transcoded FSOB-SPIHT bitstream for a particular resolution, all bits belong only to that resolution, while in the OB-SPIHT bitstream, the bits that belong to the different resolution levels are interwoven. As the resolution level increases, the difference between FSOB-SPIHT and OB-SPIHT becomes more and more significant. Note that the bit rates mentioned in Table 1 are the coding budgets that spent only for coding of the Y component. All the results are obtained without extra arithmetic coding of the encoder output bits. As shown in [3], an improved coding performance (about 0.3-0.6 dB) for SPIHT and consequently for FSOB-SPIHT can be achieved by further compressing the binary bitstreams with an arithmetic coder. Fig. 4 shows a subjective results for Akiyo object at different spatial resolution and rate obtained by

Fig. 4. Foreground object (only Y component) of the first frame of Akiyo CIF sequence at full (left), half (middle) and quarter (right) spatial resolution. Original objects (top) and reconstructed objects (bottom) at 0.25 bpp for full and half resolution and 0.125 bpp for quarter resolution.

the FSOB-SPIHT decoder.

5. CONCLUSIONS

We have presented a fully scalable object-based SPIHT (FSOB-SPIHT) algorithm for texture coding of arbitrarily shaped still object. The proposed algorithm adds the spatial scalability feature to the SPIHT bitstream while keeping important features of the original SPIHT algorithm such as high compression efficiency and rate-embeddedness (very fine granular SNR scalability) of the bitstream. The flexible scalable bitstream of FSOB-SPIHT is easily reorderable (transcodable) without any need of decoding, to obtain different bitstreams tailored for different spatial resolutions and bit rates requested by the decoder. FSOB-SPIHT is a good candidate for multimedia applications such as object-based information storage and retrieval systems, and transmission of visual information especially over heterogenous networks.

6. ACKNOWLEDGMENT

The first author would like to acknowledge the financial support provided for him by the Ministry of Science, Research and Technology (MSRT), Iran and Kurdistan University, Sanandaj, Iran during doing this

research as a part of his PhD study at the University of Wollongong, Australia.

7. REFERENCES

[1] S. Li and W. 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.

[2] J. M. Shapiro, “Embedded image coding using zerotree of wavelet coefficients,” *IEEE Trans. Signal Processing*, vol. 41, pp. 3445–3462, Dec. 1993.

[3] 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.

[4] 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.

[5] 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.

[6] 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, vol. 1, pp. 217–220.

[7] H. Danyali and A. Mertins, “Fully scalable wavelet-based image coding for transmission over heterogeneous networks,” in *Proc. 1st Workshop on the Internet, Telecommunications and Signal Processing, WITSP'02*, Wollongong, NSW, Australia, Dec. 2002, pp. 173–178.

[8] 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.