3-D Wavelet Coding of Video with Arbitrary Regions of Support[†]

Gavin Minami[†]*, Zixiang Xiong*[*]*, Albert Wang*[‡]*, and Sanjeev Mehrotra*[‡]

[†]Dept of Electrical Engineering, University of Hawaii, Honolulu, HI 96822
[*]Dept of Electrical Engineering, Texas A&M University, College Station, TX 77843
[‡]Microsoft Corporation, Redmond, WA 98052

September 28, 2000

Contact information:
Name: Zixiang Xiong. Tel: (979) 862-8683; Fax: (979) 862-4630; Email: zx@lena.tamu.edu

Transaction Letter

Abstract

We examine three-dimensional (3-D) wavelet coding of video with arbitrary regions of support (AROS). A critically sampled wavelet transform is applied to the AROS and a modified 3-D set partitioning in hierarchical trees (SPIHT) algorithm is used to quantize and code the wavelet coefficients in the AROS only. Experiments show that, for typical MPEG-4 pre-segmented sequences, our proposed method can achieve a gain of up to 5.6 dB in average PSNR at the same rate over 3-D SPIHT coding of regular volumes that embed the AROS of the given video sequences.

1 Introduction

With the development of MPEG-4 [1], the standard video coding paradigm has been extended from the usual frame-based approach to an object-based approach. Object-based coding has the advantage of allowing the structure of the video content to survive the process of acquisition, editing, and distribution. This structural information is useful for content-based search and retrieval. Object-based video coding is thus expected to play an important role in the future MPEG-7 standard [2]. In light of this, we address 3-D wavelet coding of video with arbitrary regions of support (AROS) using 3-D SPIHT coding [3, 4]. 3-D SPIHT is an extension of the celebrated SPIHT algorithm originally developed by Said and Pearlman [5] for image compression. We modify the SPIHT algorithm so that only those wavelet coefficients in the AROS are coded. Different AROS wavelet transform approaches are compared in terms of rate-distortion (R-D) performance using the modified SPIHT coding. Experiments show that our proposed method can achieve a sizable gain in average PSNR at the same bit rate over 3-D SPIHT coding of a regular volume that embeds the AROS of a given video sequence. Application examples are object-based video coding and 3-D wavelet coding of video with motion compensated AROS.

2 Wavelet transforms of signals with AROS

Many approaches have been proposed in the literature for transforming signals with AROS to the DCT or wavelet domain [6-18]. Almost all of them focus on separable transforms for images using

[†]Z. Xiong's work was supported in part by the NSF under the CAREER grant 0096070, the ARO under a YIP grant, and a gift grant from the Microsoft Corporation.

two-channel filter banks. Depending on how the boundaries are treated in the transform of finite-length AROS signals, these approaches can be roughly divided into the following four classes:

AROS transforms based on projections onto convex sets (POCS) [19]: By constraining the wavelet coefficients outside the AROS to be a constant (e.g., zero), two convex sets of images can be defined: one in the image domain (given by the pixels inside the AROS); another one in the transform domain (given by the zero coefficients outside the AROS). The iterative POCS-based procedure can then be used to compute a point in the intersection of these two sets, namely a critically sampled wavelet representation of an image with AROS [8]. The main drawback of this approach is that the basis functions restricted to the AROS are highly non-orthonormal near the boundaries, resulting in excessive quantization error in the image domain. Furthermore the boundary coefficients themselves are large and ill-behaved to make up for the zero coefficients outside the AROS, resulting in excessive bit rate. Finally the iterations required for reconstruction are computationally undesirable.

AROS transforms using boundary filters: In this method, boundary filters [9] or optimized boundary filters (with certain moment properties and equal energy) [10] are used at the signal boundaries. The drawbacks of this approach are that the boundary basis functions must be computed on-the-fly, which is a difficult task, or must be stored for every possible local boundary shape, which is infeasible in 2-D or 3-D when the filters have any non-trivial support.

AROS transforms using lifting [11]: The lifting scheme can be used to implement all regular wavelet transforms. It also handles more general cases such as wavelet transforms of signals with AROS (e.g., an interval or a sphere). In our work, however, we use the next approach.

AROS transforms using extension methods over the boundaries: This is the most popular approach. Simple extension methods such as filling up blocks not fully covered by the AROS [12] and one-pixel padding when the signal length is odd in the wavelet transform [13] often result in overcomplete representation in the transform domain. Critically sampled shape adaptive DCT algorithms that involve pixel alignments before row and column transforms are proposed in [14, 15]. Shape adaptive wavelet transforms using boundary extensions are studied in [16, 17, 18]. The most straightforward solution is given by Li and Lei [18] using biorthogonal symmetric filters with symmetric extensions over the boundaries. Since this is the method we use, we briefly describe it, in the case of odd-length filters. It was shown in [10] that odd-length biorthogonal filters in combination with odd-symmetric extensions over the boundaries yield near-optimal coding results.

Consider a 1-D signal with finite length $n > 1$. (No transform is needed when $n = 1$. An isolated pixel is multiplied by $\sqrt{2}$ and moved to the neighboring lowpass position if it is on an odd location for coding purposes.) Obtain a periodic signal with period $N = 2n - 2$ after odd-symmetric extensions over its boundaries, as shown in Fig. 1. Note that, regardless of the signal length n, N is always even so that there are an equal number of lowpass and highpass coefficients in each period after a 1-level wavelet transform. Compute the wavelet coefficients by switching between lowpass and highpass filtering along the extended periodic signal. This is possible because the centers of the the lowpass and highpass filters are offset by one, when odd-length symmetric biorthogonal filters are used. The order of filtering can start with either lowpass or highpass within each period. The filtered signal (interleaved wavelet representation) is also periodic with period N. Moreover, there are only n distinct coefficients in each period, with one extra lowpass or highpass coefficient when n is odd. This means that, by using a biorthogonal symmetric wavelet transform with symmetric extensions

over the boundaries, one can compute critically sampled wavelet transforms for both even- and odd-length signals. Fig. 1 illustrates the boundary extensions and filtering orders for the following four cases: 1) even length signal starting at an even position; 2) even length signal starting at an odd position; 3) odd length signal starting at an even position; and 4) odd length signal starting at an odd position.

Fixing the lowpass filtering at even positions and highpass filtering at odd positions (or vice versa) has the advantage that, after transform along one dimension, both lowpass and highpass coefficients are automatically aligned for subsequent transforms along other dimensions. For example, for the 2-D AROS shown in Fig. 2 after filtering in the horizontal direction, the lowpass coefficients all have even x coordinates, while the highpass coefficients all have odd x coordinates. This aligns the coefficients for subsequent filtering in the vertical direction. From Fig. 2 it is also plain to see how an AROS in the image domain induces an AROS for each subband in the transform domain.

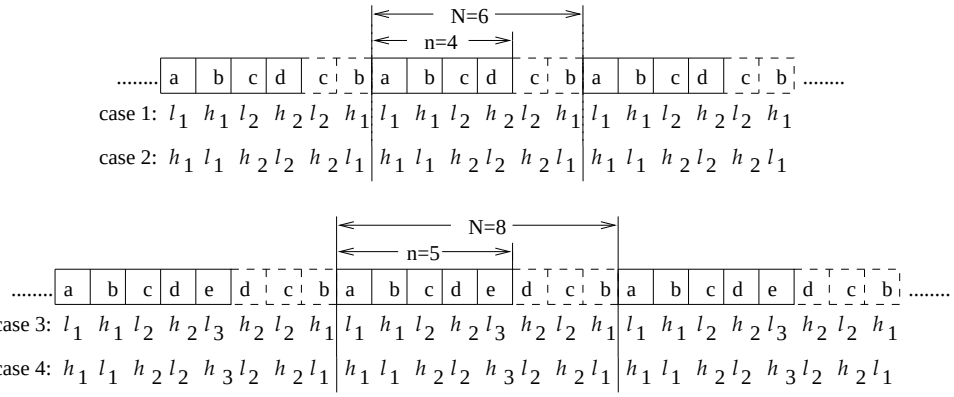


Figure 1: Critically sampled wavelet transform for both even- and odd-length signals using odd-length symmetric biorthogonal filters and odd-symmetric extensions over the boundaries.

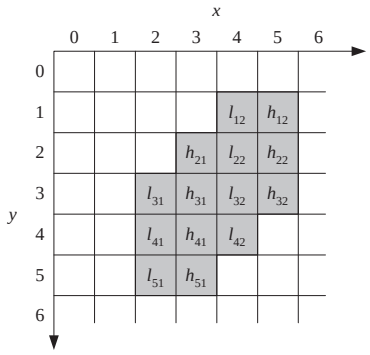
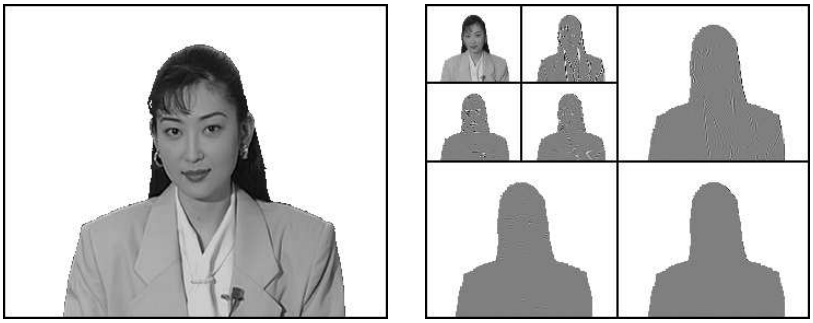


Figure 2: Example 2-D AROS with horizontal lowpass and highpass coefficients aligned vertically.

Fig. 3 shows the foreground object in the first frame of the pre-segmented QCIF Akiyo sequence and its corresponding 2-level critically sampled wavelet representation using the 9-7 biorthogonal filters of [20].

For 3-D AROS within a video scene, we use the same rule in wavelet filtering to perform a 3-D dyadic wavelet decomposition: lowpass filtering at even positions and highpass filtering at odd positions in all three dimensions. This rule allows us to automatically induce an AROS in the lowpass and highpass bands in the wavelet domain from the AROS for the original 3-D video scene. The resulting wavelet-domain AROS is used in the ensuing coding stage.



(a) (b)

Figure 3: Wavelet representation of the foreground object in the first frame of the pre-segmented QCIF Akiyo sequence. (a) The original frame. (b) Critically sampled wavelet representation of the object.

3 3-D wavelet coding of video with AROS

3.1 Previous work: the SPIHT coding algorithm

The 2-D SPIHT algorithm [5], like the embedded zerotree wavelet (EZW) coding algorithm [21], views wavelet coefficients as a collection of spatial orientation trees, with each tree consisting of coefficients from all subbands that correspond to the same spatial location in an image (see Fig. 4 (a)). It uses multipass "zerotree" coding to transmit the largest wavelet coefficients (in magnitude) first. A set of tree coefficients is significant if the largest coefficient magnitude in the set is greater than or equal to a certain threshold (e.g., a power of two); otherwise, it is insignificant. Similarly, a coefficient is significant if its magnitude is greater than or equal to the threshold; otherwise, it is insignificant. In each pass the significance of a larger set in the tree is tested first: if the set is insignificant, a binary "zerotree" bit is used to set all coefficients in the set to zero; otherwise, the set is partitioned into subsets (or child sets) for further significance tests. After all coefficients are tested in one pass, the threshold is halved before the next pass.

The underlying assumption of SPIHT coding is that most images can be modeled as having decaying power spectral densities. That is: if a parent node in the wavelet coefficient tree is insignificant, it is very likely that its descendents are also insignificant. The zerotree symbol is used very efficiently in this case to signify a spatial subtree of zeros.

When the thresholds are powers of two, SPIHT coding can be thought of as a bit-plane coding scheme. It encodes one bit-plane at a time, starting from the most significant bit. With the sign bits and refinement bits (for coefficients that become significant earlier) being coded on the fly, SPIHT achieves *embedded* coding in the wavelet domain using three lists: the list of significant pixels (LSP); the list of insignificant pixels (LIP); and the list of insignificant sets (LIS). The 2-D SPIHT coder performs competitively with most other coders published in the literature [22], while possessing desirable features such as relatively low complexity and rate embeddedness. It represents the current state-of-the-art of wavelet image coding.

3.2 3-D SPIHT

The 2-D SPIHT algorithm [5] was extended to 3-D embedded SPIHT video coding in [3, 4]. Besides motion compensation, the 3-D SPIHT algorithm is in principle the same as 2-D SPIHT, except that 3-D wavelet coefficients are treated as a collection of 3-D spatio-temporal orientation trees (see Fig.

4 (b)) and that context modeling in arithmetic coding is more involved. A block-based motion estimation scheme is implemented in the 3-D SPIHT coder in [3, 4], and an option for not using motion estimation is also allowed to reduce the encoding complexity. Global affine motion compensation was combined with the 3-D SPIHT algorithm in [23]. Every 16 frames form a group of pictures for 3-D wavelet transformation. Extension to color is accomplished without explicit bit allocation, and can be used for any color space representation. Spatio-temporal orientation trees coupled with powerful SPIHT sorting and refinement turns out to be very efficient. Even without motion compensation, the new video coder provides comparable performance to H.263 objectively and subjectively when operating at bit rates of 30 to 60 Kb/s. More importantly, it outperforms MPEG-2 at the same bit rate (1.5 to 4 Mb/s). In addition to being rate scalable, the 3-D SPIHT video coder allows multiresolutional scalability in encoding and decoding in both time and space. This added functionality along with many desirable features, such as full embeddedness for progressive transmission, precise rate control for constant bit rate traffic, and low complexity for possible software only video applications, makes the new video coder an attractive candidate for multimedia applications like Internet video.

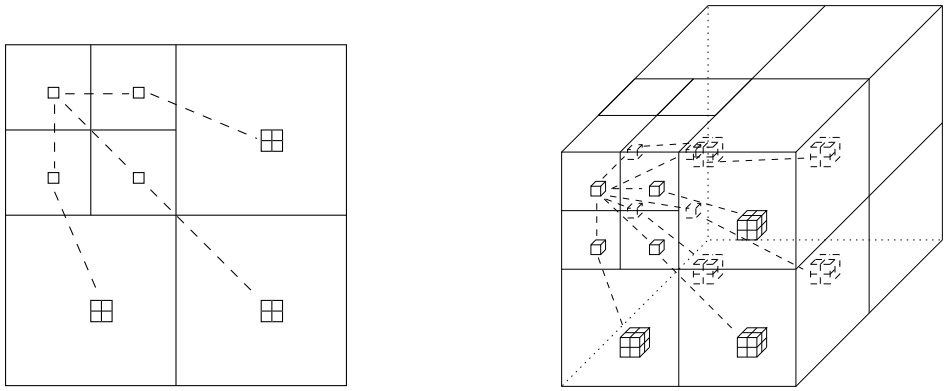


Figure 4: EZW-style 2-D spatial orientation tree and 3-D spatio-temporal orientation tree.

3.3 3-D SPIHT coding of video with AROS

Although the focus of this work is 3-D video coding, it is much easier to describe the coding operation in term of 2-D wavelet coding of images with AROS. The conceptual extension from 2-D to 3-D is straightforward. We henceforth focus on the 2-D case. Recall that the SPIHT algorithm maintains three lists (LSP, LIP, and LIS) in the process of bit-plane coding. It outputs three types of bits: significance bits, sign bits, and refinement bits. The aim of SPIHT coding of images with AROS is to modify the original SPIHT algorithm so that extraneous coefficients not in the AROS are not coded, assuming segmentation information about the AROS is available at both the encoder and decoder.

Starting with the wavelet-domain AROS induced from that for the original 2-D image, the modified SPIHT algorithm skips coding a spatial orientation tree if all coefficients in the tree (drawned in dotted lines in Fig. 5) are not in the AROS. This is simply done by not putting the coordinates of the root node (in the lowest frequency band) of the tree in the LIP and the LIS in the SPIHT initialization step.

For a spatial orientation tree with some coefficients not in the AROS (drawned in solid lines in Fig. 5), the significance test of a coefficient in the tree is skipped if that coefficient is not in the AROS. Likewise, the significance test of a subset in the tree is skipped if all coefficients in the subset

are not in the AROS. As sign bits and refinement bits are only associated with coefficients in the AROS, no modification is needed in related parts of the original SPIHT algorithm for these bits.

Finally, when all coefficients in a spatial orientation tree are in the AROS, the tree is coded in the same way as in the original SPIHT.

Note that if the AROS is the whole image, then the modified SPIHT algorithm described above will give exactly the same R-D performance as the original SPIHT algorithm does (there is no need to send the segmentation information in this case). So the new algorithm can be called a shape-adaptive SPIHT algorithm that includes the original SPIHT algorithm as a special case.

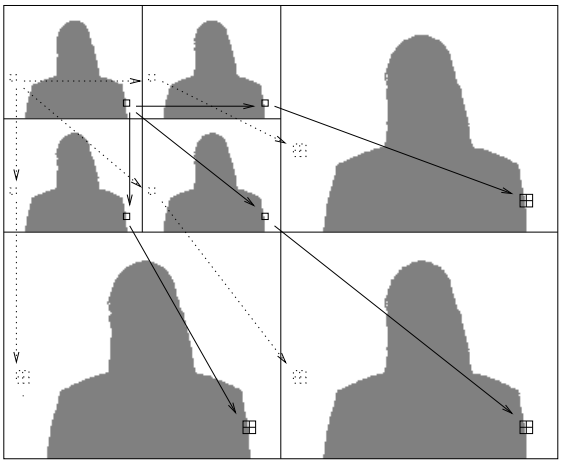


Figure 5: Two spatial orientation trees superimposed on the wavelet-domain AROS. All coefficients in one tree (drawn in dotted lines) are not in the AROS, this tree is not coded in the modified SPIHT. Another tree (drawn in solid lines) has some of its coefficients not in the AROS, significant test of a subset in the tree is skipped if every coefficient in the subset is not in the AROS.

The above procedure can be readily extended to the modification of the original 3-D SPIHT algorithm for coding of 3-D video with AROS.

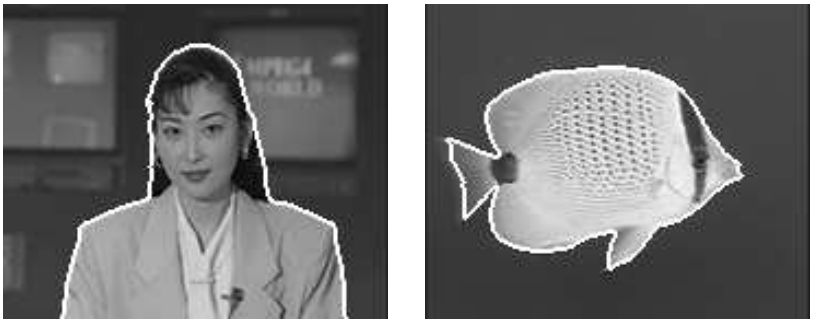


Figure 6: First frame of the pre-segmented QCIF Akiyo (left) and Bream (right) sequences.

4 Experimental Results

Two pre-segmented QCIF black and white sequences (Akiyo and Bream) are used in all experiments, where segmentation information is not accounted in AROS coding and PSNR is calculated with respect to the AROS. The first frame of each sequence is shown in Fig. 6. PSNR results using SPIHT with and without arithmetic coding are tabulated together, with those corresponding to the latter case included in parentheses.

We first compare the two critically sampled wavelet transform methods (POCS-based and boundary-extension-based) highlighted in Section 2 for images with AROS in real coding scenarios. Table 1 gives the PSNR results from modified SPIHT coding of the foreground Akiyo image in Fig. 6 using both transform methods. The boundary-extension-based method results in 1-1.78 dB gain over the POCS-based method at the same bit rate. We therefore use the boundary-extension-based method exclusively in the rest of the experiments.

Rate (b/p)	0.25	0.50	1.00
POCS-based method [8]	29.47 (29.01)	35.65 (35.43)	44.76 (44.47)
Boundary-extension-based method [18]	31.10 (30.79)	37.02 (36.70)	45.79 (45.46)

Table 1: SPIHT coding results in terms of PSNR (dB) for the foreground Akiyo image in Fig. 6 using two different critically sampled wavelet transforms. The modified SPIHT image coder is used to code the AROS.

4.1 Modified SPIHT coding of images with AROS

There are several ways to code images with AROS. One simple way (method A) is to first embed the image with AROS in a regular square or rectangular shaped image, set all pixels outside the AROS to a constant (say zero)[1], and then apply SPIHT coding on the regular shaped image. Another way (method B) is to apply critically sampled wavelet transform on the AROS but use the original SPIHT algorithm to code the regular shaped image in the wavelet domain. A better way (method C) is to use the modified SPIHT algorithm to code the critically sampled wavelet transform coefficients in the AROS only. We test these three coding methods on the foreground Akiyo and Bream images in Fig. 6. Comparisons of coding results are shown in Table 2. We see that method C indeed gives the best results as it is the most economic way of coding images with AROS. When SPIHT is used without arithmetic coding, method C gains 4.94-8.20 dB at the same rate over method A. Method B performs midway between method A and method C in this case. When SPIHT is used with arithmetic coding, method C gains 2.25-5.39 dB at the same rate over method A. Method B is only slightly worse than method C now that arithmetic coding removes most of the redundancy in the extraneous zeros outside the AROS. We note that the coding gain of methods B and C over method A depends on the AROS. For example, when the AROS is the full image, all three methods will perform the same, i.e., there is no coding gain in this case.

	Akiyo (foreground)			Bream (foreground)		
Rate (b/p)	0.25	0.50	1.00	0.25	0.50	1.00
Rectangular **T** with original SPIHT	28.85 (25.85)	34.44 (31.28)	42.64 (39.66)	23.91 (21.61)	28.29 (26.35)	36.42 (33.50)
AROS **T** with original SPIHT	30.85 (27.31)	36.84 (33.42)	45.55 (41.82)	26.99 (23.95)	32.45 (29.07)	41.68 (37.95)
AROS **T** with modified SPIHT	31.10 (30.79)	37.02 (36.70)	45.79 (45.46)	27.23 (26.66)	32.58 (32.19)	41.81 (41.70)

Table 2: Comparison of different AROS image coding methods in terms of PSNR (dB) for coding the foreground Akiyo and Bream images in Fig. 6. **T** in the table stands for transform.

[1]We have also experimented with using row/column padding to extrapolate pixels outside the AROS, as is done in MPEG-4. 3-D SPIHT coding results show that smooth extrapolatation with row/column padding does not always perform better than zero extrapolatation of pixels outside the AROS.

4.2 Modified 3-D SPIHT coding of video with AROS

We now extend the AROS image coding experiments to video coding. The pre-segmented Akiyo and Bream sequences are 300 frames each. We code each video sequence with AROS 30 frames at a time (the frame rate is assumed to be 30 f/s,) and uniformly allocate bit rates among different 30-frame segments. Two extra zero frames are padded at the end of each 30-frame segment for the 3-D wavelet transform. Three-dimensional extensions of the three coding methods described in the previous subsection are tested, and average PSNR numbers (over 300 frames) from coding three different video sequences with AROS at different bit rates are given in Table 3. Method C outperforms method A by 2.47 (2.93) - 5.27 (5.60) dB at the same rate. Again, when 3-D SPIHT is used with arithmetic coding, the difference between methods B and C is very small. In this case, the coding gain of method C over method A mostly comes from using critically sampled wavelet transform.

	Akiyo (foreground)			Akiyo (background)			Bream (foreground)		
Rate (kb/s)	20	40	60	20	40	60	20	40	60
3-D (176x144x32) **T** with original 3-D SPIHT	26.78 (24.60)	31.08 (29.12)	33.84 (31.98)	35.32 (32.89)	40.05 (37.58)	42.71 (40.56)	19.66 (18.41)	21.28 (20.38)	22.33 (21.49)
AROS **T** with Original 3-D SPIHT	28.99 (25.64)	33.24 (30.33)	36.07 (33.25)	40.50 (35.46)	45.30 (41.62)	47.94 (44.83)	22.13 (20.56)	23.73 (22.30)	25.00 (23.36)
AROS **T** with modified 3-D SPIHT	29.04 (28.02)	33.29 (32.19)	36.10 (35.00)	40.52 (36.97)	45.32 (43.04)	47.96 (46.16)	22.17 (21.71)	23.75 (23.31)	25.02 (24.51)

Table 3: Comparison of different AROS video coding methods in terms of average PSNR (dB) at different bit rates for coding the foreground Akiyo, the background Akiyo, and the foreground Bream sequences.

Fig. 7 shows the PSNR vs. frame number plots for the foreground Akiyo sequence using three coders at the same rate: the original 3-D SPIHT coder, the new modified 3-D SPIHT coder, and the latest MPEG-4 coder (MS version). GOP=30 in all cases. We picked the shape-adaptive DCT option in MPEG-4 with the IPPP... structure within each GOP. PSNR values given by 3-D SPIHT coders have large variance. This is because GOPs are coded independently. In addition, the modified 3-D SPIHT coder is 0.50 dB worse than the MPEG-4 coder in terms of average PSNR.

5 Conclusions

We extend the original SPIHT algorithm from coding of images and video with regular regions of support to coding of those with AROS. Different wavelet transforms for images with AROS are compared. Three methods for coding images and video with AROS are presented. When coupled with critically sampled wavelet transform using the boundary-extension-based method, the modified SPIHT algorithm gives excellent coding results for images and video with AORS. It is suitable for object-based coding. Current research focuses on incorporating motion models in video sequences with AROS for better coding performance.

6 Acknowledgement

The authors would like to gratefully thank Dr. Philip A. Chou for his guidance and contribution to this paper.

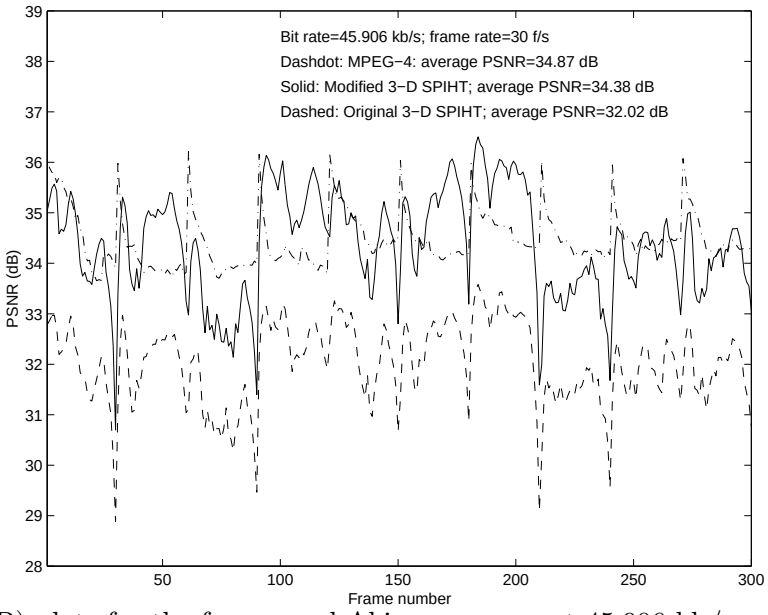


Figure 7: PSNR (dB) plots for the foreground Akiyo sequence at 45.906 kb/s using the original 3-D SPIHT coder, the new modified 3-D SPIHT coder, and the MPEG-4 coder. GOP=30 in all cases.

References

[1] MPEG-4 Requirements Group, "MPEG-4 requirements version 4.0," Tech. Rep., ISO/IEC JTC1/SC29/WG11, Stockholm, July 1997.

[2] MPEG-7 Requirements Group, "MPEG-7 content and objectives," Tech. Rep., ISO/IEC JTC1/SC29/WG11, Sevilla, Feb. 1997.

[3] W. A. Pearlman, B.-J. Kim, and Z. Xiong, "Embedded video subband coding with 3D SPIHT," in *Wavelet Image and Video Compression*, P. Topiwala, Ed. Kluwer, 1998.

[4] B.-J. Kim, Z. Xiong, and W. A. Pearlman, "Very low bit-rate embedded video coding with 3D set partitioning in hierarchical trees (3D SPIHT)," *IEEE Trans. Circuits and Systems for Video Technology*, Oct. 1997, submitted.

[5] A. Said and W. A. Pearlman, "A new, fast, and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 6, no. 3, pp. 243–250, June 1996.

[6] M. Gilge, T. Engelhart, and R. Mehlan, "Coding of arbitrarily shaped image segments based on a generalized orthogonal transform," *Signal Processing: Image Communication*, vol. 1, pp. 153–180, 1989.

[7] J. Apostolopoulos and J. Lim, "Representing arbitrarily-shaped regions: a case study of overcomplete representations," in *Proc. Int'l Conf. Image Processing*, Washington, D.C., Oct. 1995, IEEE.

[8] J. Apostolopoulos and J. Lim, "Critically sampled wavelet representations for multidimensional signals with arbitrary regions of support," in *Proc. Int'l Conf. Acoustics, Speech, and Signal Processing*, Seattle, WA, May 1998, IEEE.

[9] C. Herley, "Boundary filters for finite-length signals and time-varying filter banks," *IEEE Trans. Circuits and Systems*, vol. 42, pp. 102–144, Feb. 1995.

[10] A. Mertins, "Optimized biorthogonal shape adaptive wavelets," in *Proc. Int'l Conf. Image Processing*, Chicago, IL, Oct. 1998, IEEE.

[11] W. Sweldens and P. Schröder, "Building your own wavelets at home," in *Wavelets in Computer Graphics*, pp. 15–87. ACM SIGGRAPH Course notes, 1996.

[12] Z. Wu and T. Kanamaru, "Block-based DCT and wavelet selective coding for arbitrarily shaped images," in *Proc. Visual Communications and Image Processing*, San Jose, CA, Jan. 1997, SPIE, pp. 658–665.

[13] J.-H. Kim, J-Y. Lee, E.-S. Kang, and S.-J. Ko, "Region-based wavelet transform for image compression," *IEEE Trans. Circuits and Systems*, vol. 45, pp. 1137–1140, Aug. 1998.

[14] T. Sikora and B. Makai, "Shape-adaptive DCT for generic coding of video," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 5, pp. 59–62, Feb. 1995.

[15] P. Kauff, B. Makai, S. Rauthenberg, U. Golz, J. Lameillieure, and T. Sikora, "Functional coding of video using a shape-adaptive DCT algorithm and an object-based motion compensation toolbox," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 7, pp. 181–196, Feb. 1997.

[16] H. Barnard, J. Weber, and J. Biemond, "A region-based discrete wavelet transform for image coding," in *Wavelets In Image Communication*, M. Barlaud, Ed. Elsevier, The Netherlands, 1994.

[17] S. Li and W. Li, "Shape-adaptive discrete wavelet transforms for arbitrarily shaped visual object coding," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 10, no. 5, pp. 725–743, Aug. 2000.

[18] J. Li and S. Lei, "Arbitrary shape wavelet transform with phase alignment," in *Proc. Int'l Conf. Image Processing*, Chicago, IL, Oct. 1998, IEEE.

[19] D. C. Youla, "Mathematical theory of image restoration by the method of convex projections," in *Image Recovery: Theory and Applications*, H. Stark, Ed. Academic Press, 1987.

[20] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, "Image coding using wavelet transform," *IEEE Trans. Image Processing*, vol. 1, pp. 205–221, Apr. 1992.

[21] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Trans. Signal Processing*, vol. 41, no. 12, pp. 3445–3463, Dec. 1993.

[22] University of California at Los Angeles (UCLA) Image Communications Lab, "Wavlet image coding: PSNR results," Web site http://www.icsl.ucla.edu/~ipl/psnr_results.html, 1998.

[23] A. Wang, Z. Xiong, P. A. Chou, and S. Mehrotra, "Three-dimensional wavelet coding of video with global motion compensation," in *Proc. Data Compression Conference*, Snowbird, UT, Mar. 1999, IEEE Computer Society.