Shape-Adaptive Embedded Coding of Ocean-Temperature Imagery

Justin T. Rucker and James E. Fowler
Department of Electrical and Computer Engineering
GeoResources Institute (GRI)
Mississippi State University, Mississippi State, MS 39762 USA

***Abstract*— An embedded wavelet-based coder for the shape-adaptive coding of ocean-temperature data is described. The proposed coder, 3D binary set splitting with k-d trees (3D-BISK), is based upon the popular bitplane-coding paradigm and is specifically designed for shape-adaptive coding. Other similar coding methods use octree-based set partitioning; however, 3D-BISK employs a simpler set decomposition based on k-d trees which makes it more flexible when considering shape-adaptive coding. The performance of 3D-BISK is compared to prominent shape-adaptive coders and superior performance is demonstrated for a variety of ocean-temperature datasets.**

I. Introduction

The compression of imagery with arbitrary shape has become an important issue in several multimedia application areas, with the recent MPEG-4 video-coding standard [1] being the prime example. However, certain geoscience applications have also benefited from such shape-adaptive coding. For example, the US Naval Oceanographic Office (NAVOCEANO) generates three-dimensional oceanographic temperature datasets for rectangular regions of sea and land at standard ocean depths. Points that refer to land or lie beyond the bathymetry are considered to have no valid data. Thus, the compression of such ocean-temperature data requires shape-adaptive coding.

The problem of the shape-adaptive coding of ocean-temperature imagery was considered in [2, 3], wherein the modern paradigm of embedded wavelet-based coding—at the time quickly becoming the preferred approach to the compression of 2D images—was adapted to 3D ocean-temperature imagery with arbitrary shape. Since that time, a number of 3D embedded wavelet-based techniques with improved performance have been proposed, albeit in the context traditional, rectangular imagery. Yet, these coders all have a common design built upon three major components—a 3D wavelet transform, significance-map encoding, and successive-approximation quantization (i.e., bitplane coding)—which can be easily made shape adaptive.

A key process in embedded wavelet-based coders is the mapping of the significance state of each wavelet coefficient (i.e., whether or not the coefficient is greater than or less than the current threshold) into a binary-valued significance map with the threshold decreasing for each successive pass through the dataset. Such coders can be made shape adaptive and applied to the ocean-temperature compression problem by employing a 3D shape-adaptive wavelet transform [4] which transforms only the ocean regions; land regions, where no data exists, are permanently considered insignificant in the significance map. The major difference between wavelet-based compression schemes lies in the method for coding the significance map; consequently, the key to shape-adaptive coding is to modify this significance-map encoding to accommodate the presence of land regions wherein no valid data lies.

In this paper, we describe 3D binary set splitting with k-d trees (3D-BISK), which is a 3D extension of the 2D-BISK coder proposed in [5]. BISK is itself a variant of the well-known, state-of-the-art set-partitioning embedded block (SPECK) algorithm [6, 7]; its shape-adaptive version, object-based SPECK (OB-SPECK) [8]; and its 3D extension, 3D-SPECK [9]. Experimental evidence has shown that 3D-SPECK demonstrates performance roughly equivalent to that of the prominent JPEG-2000 standard [10] in tasks such as the compression of hyperspectral image cubes. JPEG-2000, on the other hand, does not support shape-adaptive coding.

The main contribution of this paper is the development of the 3D-BISK algorithm which replaces the octree set-partitioning operation of 3D-SPECK with k-d trees [11], a simpler set decomposition particularly well-suited to shape-adaptive coding due to its greater flexibility at capturing arbitrarily shaped regions. Additionally, 3D-BISK aggressively discards land regions from consideration by shrinking the decomposed sets to the bounding box of their ocean regions. Empirical results demonstrate that 3D-BISK consistently yields rate-distortion performance significantly superior to that of a number of other 3D shape-adaptive embedded coders for the coding of a variety of ocean-temperature volumes.

The remainder of this paper is organized as follows. In Sec. II, we briefly review a number of prior approaches to shape-adaptive coding. Next, we describe the 3D-BISK algorithm in detail in Sec. III and experimentally compare it to other techniques in Sec. IV. Finally, we make some concluding remarks in Sec. V. We note that a preliminary description of 3D-BISK appeared in [12]; here, we give a more thorough presentation of the algorithm, as well as a more comprehensive experimental investigation.

II. Shape-Adaptive Coding

In this section, we review a number of prior approaches to 3D shape-adaptive coding. Essentially, 3D shape-adaptive coders are direct extensions to 3D of algorithms developed for

In *Proceedings of the* 40^{th} *Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove, CA, October 2006, pp. 1887-1891.

2D imagery with arbitrary shape. Regardless of the dimensionality of the data, the straightforward approach to shape-adaptive coding involves applying a transform to only the valid ocean data and treating the remaining land regions as permanently "insignificant." The bitplane-coding passes can then process these land regions in the same way as other insignificant coefficients. While most shape-adaptive coders are based on this general idea, a number of approaches employ various modifications to the significance-map encoding to increase performance.

The transform used is a 3D version of the shape-adaptive wavelet transform of [4]. There are two ways to extend a 2D shape-adaptive transform to 3D. The first is a so-called wavelet-packet transform in which each ocean-depth layer is decomposed using a separable 2D transform followed with a 1D decomposition in the depth direction. Alternatively, there is the widely-used the dyadic transform structure in which the lowpass baseband subband is recursively decomposed for each transform scale. Unless otherwise noted, each of the following shape-adaptive algorithms can use either the dyadic or wavelet-packet transform structures.

A. Shape-Adaptive 3D-SPIHT

Set partitioning in hierarchical trees (SPIHT) [13] is one of the most prominent embedded wavelet-based coders for 2D images; it was extended to 3D in [14] and made shape adaptive in [15]. Significance-map encoding for 3D-SPIHT involves the coding of the insignificance of entire tree-structured sets (zerotrees) across multiple scales of a wavelet transform. The shape-adaptive version of 3D-SPIHT follows the straightforward approach described above by aggregating large land regions together with insignificant ocean regions into zerotree sets. Further refinement to the algorithm can be made by discarding subtrees consisting entirely of land regions (which are permanently insignificant) from further consideration [15].

B. 3D-OB-SPECK

The SPECK algorithm [6, 7] eliminates the cross-scale aggregation of coefficients that occurs in SPIHT and other zerotree-based algorithms and instead applies partitioning to sets of contiguous coefficients within each individual subband. SPECK was made shape adaptive as OB-SPECK in [8] and extended to 3D as 3D-SPECK [9]. To our knowledge, the first 3D shape-adaptive version of SPECK (3D-OB-SPECK) appeared as an implementation in QccPack [16]. In 3D-OB-SPECK, the significance state of an entire set is tested and coded; then, if the set contains at least one significant coefficient, it is split into eight subsets (i.e., octree partitioning), and the process is repeated recursively for each subset. 3D-OB-SPECK is similar to shape-adaptive 3D-SPIHT in that land is considered to be permanently insignificant, and, when a set contains only land coefficients, it is removed from further consideration.

C. 3D-tarp

A unique approach to significance-map coding, tarp coding [17], uses a nonadaptive arithmetic coder coupled with an explicit probability estimate of the significance map. In the tarp coder, the density estimation is efficiently computed by a novel series of 1D filtering operations coined tarp filtering. In contrast to other prominent wavelet-based coders, tarp coding lacks complex context modeling or cross-subband, cross-scale aggregation of symbols such as zerotree structures. Tarp, originally developed for 2D rectangular images, was extended to 3D imagery in [18, 19] and to shape-adaptive coding in [20]. Shape-adaptive tarp coding calls for "skipping" over land regions while maintaining the current probability estimate unchanged. Because the tarp algorithm lacks context modeling and symbol aggregation, this skipping of land leads to efficient performance for shape-adaptive coding.

D. 3D-WDR

In embedded wavelet-based coding, the significance map forms a binary image; consequently, techniques that have been employed for the coding of bilevel images are applicable to significance-map coding. For example, runlength coding has a long history of such binary-coding use. The wavelet difference reduction (WDR) [21] algorithm combines runlength coding of the significance map with an efficient lossless representation of runlength symbols to produce an embedded image coder. WDR was originally developed as a 2D coder, but is straightforwardly extended to 3D, which was done as an implementation in QccPack [16]. In addition, WDR can be made shape adaptive by "skipping" over land regions and not coding any significance information for them or including them in the runlengths. We note that the algorithm described in [2, 3], which is currently used by NAVOCEANO for the coding of ocean-temperature imagery, is very similar to shape-adaptive 3D-WDR.

E. EBCOT

The recent JPEG-2000 standard [10] is the most prominent example of techniques that code the significance map using known significance states of neighboring coefficients to provide a context for the coding of the significance state of the current coefficient with an adaptive arithmetic coder. While the JPEG-2000 standard does not support arbitrarily shaped images, the underlying embedded block coding with optimized truncation (EBCOT) algorithm [22] is easily made shape adaptive. In shape-adaptive EBCOT [23], land regions are ignored and not coded in all coding passes, while anytime that the context for an ocean coefficient overlaps the bathymetry boundary, land coefficients in the context are treated as insignificant.

The significance-map coding in EBCOT is strictly a 2D process. That is, a 2D image is transformed with a 2D wavelet transform, and each subband is partitioned into a number of codeblocks, with an embedded bitstream generated independently for each codeblock. EBCOT can be used for 3D datasets by applying this 2D codeblock-based procedure to each 2D "slice" of the 3D dataset, truncating each codeblock bitstream, and then concatenating the truncated bitstreams together to form the final bitstream. In the 3D-EBCOT coding of [15],

In *Proceedings of the* 40^{th} *Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove, CA, October 2006, pp. 1887-1891.

a Lagrangian rate-distortion-optimal truncation procedure is used as in the original 2D EBCOT formulation [22]. We note that, due to the 2D nature of its codeblock processing, the 3D-EBCOT algorithm must use the wavelet-packet transform, whereas the preceding techniques can use either the wavelet-packet or dyadic decomposition structures.

III. 3D BINARY SET-SPLITTING WITH k-D TREES

In this section, we describe our 3D-BISK algorithm and the k-d tree set-partitioning structure upon which it is based. 3D-BISK is a variant of 3D-OB-SPECK that is well-suited to shape-adaptive coding. The octree-partitioning of 3D-OB-SPECK is replaced by binary set splitting of k-d trees which allows for a more flexible coding of arbitrarily shaped regions. An additional key difference between 3D-OB-SPECK and 3D-BISK is the aggressive "shrinking" of the sets to the bounding box of the ocean coefficients contained in the set, which is responsible for a large part of the performance gain.

A. Set Partitioning with k-d Trees

Octrees and k-d trees [11] are two well-known methods for the partitioning of 3D sets. In an octree, a set (a rectangular prism) is divided along all three dimensions to form eight equally sized subsets. On the other hand, in a k-d tree, a set is divided along a single dimension into two subsets whose sizes are not necessarily equal. It is straightforward to see that k-d trees can achieve a partitioning of a set identical to that resulting from an octree decomposition, although usually a greater number of levels of decomposition are needed. However, we demonstrate below that the k-d trees decomposition is advantageous for shape-adaptive coding of the significance map.

B. The 3D-BISK Algorithm

Following a 3D wavelet transform, the 3D-BISK algorithm begins by placing the subbands of the transformed coefficients into a list of insignificant sets (LIS); subsequently, each subband is "shrunk" to the bounding volume of its ocean coefficients. As in 3D-OB-SPECK, each LIS is indexed, and a given set $\mathcal{S}$ resides in the LIS with index $N(\mathcal{S})$, i.e., in $\text{LIS}_{N(\mathcal{S})}$. The LIS index, $N(\mathcal{S})$, is the total number of decompositions, or splits, that has produced the set. The algorithm continues in the usual bitplane-coding fashion with sorting and refinement passes. The algorithm is as follows:

```
procedure BISK(𝒳)
  Initialization(𝒳)
  n ← max bitplane
  while (true)
    SortingPass()
    RefinementPass()
    n ← n − 1
procedure Initialization(𝒳)
  for each subband 𝒮 in 𝒳
    N(𝒮) ← total number of decompositions
             in all dimensions
    ShrinkSet(𝒮)
    append 𝒮 to LIS_{N(𝒮)}
  LSP ← ∅
procedure SortingPass()
  l = number of LIS lists
  while l > 0
    for each 𝒮 ∈ LIS_{N(𝒮)}
      ProcessSet(𝒮)
    l ← l − 1
procedure RefinementPass()
  for each 𝒮 ∈ LSP
    output n^th bitplane value of coefficient magnitude
```

Like 3D-OB-SPECK, 3D-BISK tests the significance of all the sets in all the LIS lists. A set is considered to be significant if the magnitude of the largest ocean coefficient exceeds a threshold. If a set contains no significant ocean coefficients, it is placed into an LIS and will be processed at the next lower threshold. Since 3D-BISK employs k-d trees, when a set becomes significant, it is split into halves. Each half is then placed in an LIS and processed in the same manner until decomposed to a single pixel. If a set being processed contains no ocean coefficients, the set is removed from its LIS and discarded. The algorithms for processing and partitioning the sets is described below:

```
procedure ProcessSet(𝒮)
  if 𝒮 = ∅
    remove 𝒮 from LIS_{N(𝒮)}
  else
    output Γ_n(𝒮)
    if Γ_n(𝒮) = 1
      remove 𝒮 from LIS_{N(𝒮)}
      if |𝒮| = 1
        output sign of 𝒮
        append 𝒮 to LSP
      else
        CodeSet(𝒮)
procedure CodeSet(𝒮)
  {𝒮_1, 𝒮_2} = PartitionSet(𝒮)
  if 𝒮_1 ≠ ∅
    append 𝒮_1 to LIS_{N(𝒮_1)}
    ProcessSet(𝒮_1)
  append 𝒮_2 to LIS_{N(𝒮_2)}
  ProcessSet(𝒮_2)
procedure PartitionSet(𝒮)
  if z(𝒮) ≥ y(𝒮) ≥ x(𝒮)
    split 𝒮 depth-wise into 𝒮_1 and 𝒮_2:
      𝒮_1: ⌊z(𝒮)/2⌋ × y(𝒮) × x(𝒮)
      𝒮_2: (z(𝒮) − ⌊z(𝒮)/2⌋) × y(𝒮) × x(𝒮)
  else
    if x(𝒮) ≥ y(𝒮) > z(𝒮)
      split 𝒮 horizontally into 𝒮_1 and 𝒮_2:
        𝒮_1: z(𝒮) × ⌊y(𝒮)/2⌋ × x(𝒮)
        𝒮_2: z(𝒮) × (y(𝒮) − ⌊y(𝒮)/2⌋) × x(𝒮)
  else
    if y(𝒮) > z(𝒮) > x(𝒮)
      split 𝒮 vertically into 𝒮_1 and 𝒮_2:
```

$\mathcal{S}_1$: $z(\mathcal{S}) \times y(\mathcal{S}) \times \lfloor x(\mathcal{S})/2 \rfloor$
$\mathcal{S}_2$: $z(\mathcal{S}) \times y(\mathcal{S}) \times \big(x(\mathcal{S}) - \lfloor x(\mathcal{S})/2 \rfloor\big)$
$N(\mathcal{S}_1) = N(\mathcal{S}) + 1$
$N(\mathcal{S}_2) = N(\mathcal{S}) + 1$
`ShrinkSet`($\mathcal{S}_1$)
`ShrinkSet`($\mathcal{S}_2$)

Above, $\Gamma_n(\mathcal{S})$ is the significance state of set $\mathcal{S}$, and $z(\mathcal{S})$, $y(\mathcal{S})$, and $x(\mathcal{S})$ are the number of ocean depths, rows, and columns, respectively, of set $\mathcal{S}$.

The use of k-d trees in 3D-BISK is advantageous for the adaptive arithmetic coder used to code the significance information. Specifically, when set $\mathcal{S}$ is split into $\mathcal{S}_1$ and $\mathcal{S}_2$, and $\mathcal{S}_1$ is known to be insignificant (or empty), the significance state of $\mathcal{S}_2$ is guaranteed to be significant. In this case, the significance state $\Gamma_n(\mathcal{S}_2)$ is not coded into the bitstream. In the other case, the coding of $\Gamma_n(\mathcal{S}_2)$ is conditioned with the knowledge that $\mathcal{S}_1$ is significant.[1] By contrast, the first seven subsets of an octree decomposition must be known to be insignificant for 3D-OB-SPECK to benefit from the same strategy. The contexts used for set-significance coding within the BISK algorithm are as follows:

$c(\mathcal{S}_1) \leftarrow$ `CONTEXT_S1`
if $\mathcal{S}_1 = \emptyset$ or $\Gamma_n(\mathcal{S}_1) = 0$
  $c(\mathcal{S}_2) \leftarrow$ `CONTEXT_NOCODE`
else
  $c(\mathcal{S}_2) \leftarrow$ `CONTEXT_S2`

Above, $c(\mathcal{S}_i)$ denotes the context that will be used to code $\Gamma_n(\mathcal{S}_i)$.

In `PartitionSet`() above, a set is split into roughly equal-sized halves along the dimension of the set which is the longest. We refer to this strategy for determining the location and dimension of the k-d tree set-splitting operation as "longest dimension" (LD). We have investigated several alternative set-splitting approaches: 1) split a set into halves along the dimension that results in the smallest-sized sets after the set-shrinking operation (i.e., "smallest set" (SS)), 2) split a set along its longest dimension at the "center of mass" of its ocean points (i.e., "center of mass" (CM)), and 3) split a set at the center of mass along the dimension and that results in the smallest set sizes after the set-shrinking operation (i.e., "smallest set/center of mass" (SSCM)). In the experimental results that follow in the next section, we empirically compare the performance of these set-splitting strategies.

IV. Experimental Results

The performance of our 3D-BISK algorithm is measured using the ocean-temperature data from the study in [2, 3]. We use a three-level wavelet transform with the popular 9/7 biorthogonal filters [24], and find that the dyadic transform structure significantly outperforms the corresponding wavelet-packet transform. Consequently, all algorithms use this dyadic transform, with the sole exception being 3D-EBCOT, which cannot use the dyadic transform. All results for 3D-EBCOT are thus for the wavelet-packet transform. Throughout the experiments, rate is measured in bits per voxel (bpv), and distortion is measured as signal-to-noise ratio (SNR) in dB.

Table I compares the distortion performance at a given rate for the four set-splitting strategies for 3D-BISK outlined in Sec. III-B. We see that, although the performances for all four approaches are rather similar, the LD strategy—which is the simplest of the four to implement—performs nearly as well as or better than all the other splitting policies. Consequently, we use the LD splitting strategy exclusively throughout the remainder of the results.

Fig. 1 presents the distortion obtained for two datasets over a range of rates. We see that 3D-BISK outperforms 3D-EBCOT by a wide margin (3–10 dB); this performance gap is due largely to the fact that 3D-EBCOT is constrained to use the wavelet-packet decomposition structure. Distortion performance for a given rate is tabulated in Table II for all the algorithms which use the dyadic transform. We see that 3D-BISK almost always yields the best distortion performance, being outperformed by tarp for only one dataset.

V. Conclusions

In this paper, we described 3D-BISK, an embedded, wavelet-based, 3D shape-adaptive coder for the compression of ocean-temperature imagery. We compared 3D-BISK to a variety of prominent 3D wavelet-based coding techniques, and experimental results demonstrated superior performance for 3D-BISK for a variety of ocean-temperature datasets. The performance gain was attributed to aggressive discarding of land-only sets and the simpler, more flexible partitioning of the sets that results from the binary k-d tree set-decomposition structure.

Interestingly, k-d trees often require more decompositions to represent a dataset than octrees. This suggests that binary set splitting allows for a simpler, more efficient entropy coding of the significance map. To confirm this hypothesis, we coded full datasets with no land so that `ShrinkSet`() had no effect on the coding process. In the case of coding these datasets, 3D-BISK and 3D-OB-SPECK exhibited virtually identical performance despite that fact that 3D-BISK incurred roughly seven times as many set-decomposition operations.

[1] Due to how sets are partitioned (see `PartitionSet`()), $\mathcal{S}_2$ is guaranteed to be nonempty while $\mathcal{S}_1$ may or may not be empty.

References

[1] *Information Technology—Coding of Audio-Visual Objects—Part 2: Visual*, ISO/IEC 14496-2, 1999, MPEG-4 Coding Standard.

[2] J. E. Fowler and D. N. Fox, "Embedded wavelet-based coding of three-dimensional oceanographic images with land masses," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 39, no. 2, pp. 284–290, February 2001.

[3] ——, "Wavelet-based coding of three-dimensional oceanographic images around land masses," in *Proceedings of the International Conference on Image Processing*, vol. 2, Vancouver, Canada, September 2000, pp. 431–434.

[4] S. Li and W. Li, "Shape-adaptive discrete wavelet transforms for arbitrary shaped visual object coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, no. 5, pp. 725–743, August 2000.

In *Proceedings of the* 40^{th} *Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove, CA, October 2006, pp. 1887-1891.

[5] J. E. Fowler, "Shape-adaptive coding using binary set splitting with k-d trees," in *Proceedings of the International Conference on Image Processing*, vol. 2, Singapore, October 2004, pp. 1301–1304.
[6] A. Islam and W. A. Pearlman, "An embedded and efficient low-complexity hierarchical image coder," in *Visual Communications and Image Processing*, K. Aizawa, R. L. Stevenson, and Y.-Q. Zhang, Eds. San Jose, CA: Proc. SPIE 3653, January 1999, pp. 294–305.
[7] W. A. Pearlman, A. Islam, N. Nagaraj, and A. Said, "Efficient, low-complexity image coding with a set-partitioning embedded block coder," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 11, pp. 1219–1235, November 2004.
[8] Z. Lu and W. A. Pearlman, "Wavelet video coding of video object by object-based SPECK algorithm," in *Proceedings of the Picture Coding Symposium*, Seoul, Korea, April 2001, pp. 413–416.
[9] X. Tang, W. A. Pearlman, and J. W. Modestino, "Hyperspectral image compression using three-dimensional wavelet coding," in *Image and Video Communications and Processing*, B. Vasudev, T. R. Hsing, A. G. Tescher, and T. Ebrahimi, Eds. Santa Clara, CA: Proc. SPIE 5022, January 2003, pp. 1037–1047.
[10] *Information Technology—JPEG 2000 Image Coding System—Part 1: Core Coding System*, ISO/IEC 15444-1, 2000.
[11] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Communications of the ACM*, vol. 18, no. 9, pp. 509–517, September 1975.
[12] J. T. Rucker and J. E. Fowler, "Coding of ocean-temperature volumes using binary set splitting with k-d trees," in *Proceedings of the International Geoscience and Remote Sensing Symposium*, vol. 1, Anchorage, AK, September 2004, pp. 289–292.
[13] A. Said and W. A. Pearlman, "A new, fast, and efficient image codec based on set partitioning in hierarchical trees," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, no. 3, pp. 243–250, June 1996.
[14] 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 Transactions on Circuits and Systems for Video Technology*, vol. 10, no. 8, pp. 1374–1387, December 2000.
[15] G. Minami, Z. Xiong, A. Wang, and S. Mehrota, "3-D wavelet coding of video with arbitrary regions of support," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 9, pp. 1063–1068, September 2001.
[16] J. E. Fowler, "QccPack: An open-source software library for quantization, compression, and coding," in *Applications of Digital Image Processing XXIII*, A. G. Tescher, Ed. San Diego, CA: Proc. SPIE 4115, August 2000, pp. 294–301.
[17] P. Simard, D. Steinkraus, and H. Malvar, "On-line adaptation in image coding with a 2-D tarp filter," in *Proceedings of the IEEE Data Compression Conference*, J. A. Storer and M. Cohn, Eds., Snowbird, UT, April 2002, pp. 23–32.
[18] Y. Wang, J. T. Rucker, and J. E. Fowler, "3D tarp coding for the compression of hyperspectral images," *IEEE Geoscience and Remote Sensing Letters*, vol. 1, no. 2, pp. 136–140, April 2004.
[19] ——, "Embedded wavelet-based compression of hyperspectral imagery using tarp coding," in *Proceedings of the International Geoscience and Remote Sensing Symposium*, vol. 3, Toulouse, France, July 2003, pp. 2027–2029.
[20] J. E. Fowler, "Shape-adaptive tarp coding," in *Proceedings of the International Conference on Image Processing*, vol. 1, Barcelona, Spain, September 2003, pp. 621–624.
[21] J. Tian and R. Wells, Jr., "Embedded image coding using wavelet difference reduction," in *Wavelet Image and Video Compression*, P. N. Topiwala, Ed. Boston, MA: Kluwer Academic Publishers, 1998, ch. 17, pp. 289–301.
[22] D. Taubman, "High performance scalable image compression with EBCOT," *IEEE Transactions on Image Processing*, vol. 9, no. 7, pp. 1158–1170, July 2000.
[23] Z. Liu, J. Hua, Z. Xiong, and K. Castleman, "Lossy-to-lossless ROI coding of chromosome images using modified SPIHT and EBCOT," in *Proceedings of the IEEE International Symposium on Biomedical Imaging*, Washington, DC, July 2002, pp. 317–320.
[24] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, "Image coding using wavelet transform," *IEEE Transactions on Image Processing*, vol. 1, no. 2, pp. 205–220, April 1992.

TABLE I

COMPARISON OF 3D-BISK SET-SPLITTING STRATEGIES AT 1.0 BPV

	SNR (dB)			
Dataset	LD	CM	SS	SSCM
bisca	53.1	53.1	**53.2**	53.1
nwlan	**58.7**	58.6	**58.7**	58.5
okina	61.6	61.6	61.5	**61.7**
ylsoj	**54.2**	54.1	54.1	**54.2**

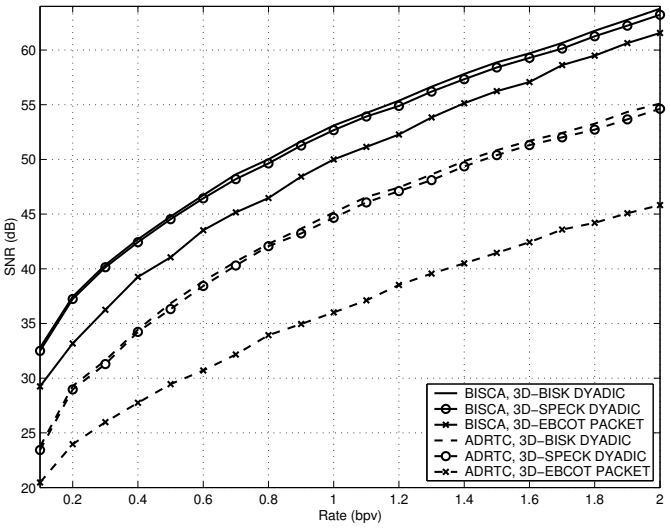

Fig. 1. Rate-distortion performance for adrtc and bisca.

TABLE II

DISTORTION PERFORMANCE AT 1.0 BPV

	SNR (dB)				
Dataset	BISK	SPECK	SPIHT	TARP	WDR
adrtc	**45.2**	44.7	40.1	44.1	44.7
bisca	**53.1**	52.7	51.6	52.8	51.8
ginse	**48.1**	47.8	46.7	**48.1**	47.3
guama	**70.7**	69.7	68.7	70.3	70.1
hawai	**60.3**	59.9	59.0	59.7	60.1
med	**49.3**	48.9	46.7	48.9	48.8
nwlan	**58.7**	58.4	57.4	58.3	57.4
okina	61.6	60.7	57.3	**62.3**	61.1
socal	**45.3**	44.7	38.4	44.2	44.5
ylsoj	**54.2**	53.9	52.7	54.0	53.3