

International Journal of Wavelets, Multiresolution and Information Processing
© World Scientific Publishing Company

COMPRESSION OF SEGMENTED 3D SEISMIC DATA

VALERY A. ZHELUDEV

*School of Computer Science, Tel Aviv University
Tel Aviv, 69978, Israel
zhel@post.tau.ac.il*

DAN D. KOSLOFF

*Department of Earth and Planetary Sciences,
Tel Aviv University, Tel Aviv, 69978, Israel
dan@seismo.tau.ac.il*

EUGENE Y. RAGOZA

*Company Paradigm Geophysical Ltd.,
Hertzlia, 46120, Israel.
eugene@paradigmgeo.com*

Received (Day Month Year)

Revised (Day Month Year)

Communicated by (xxxxxxxxxx)

We present a preliminary investigation of compression of segmented 3D seismic volumes for the rendering purposes. Promising results are obtained on the base of 3D discrete cosine transforms followed by the SPIHT coding scheme. An accelerated version of the algorithm combines 1D discrete cosine transform in vertical direction with the 2D wavelet transform of horizontal slices. In this case the SPIHT scheme is used for coding the mixed sets of cosine-wavelet coefficients.

Keywords: seismic; compression, DCT; wavelet.

AMS Subject Classification: 94A08, 68P30

1. Introduction

In contemporary 3D seismic processing very large data arrays are involved. Efficient compression algorithms in order to rapidly transmit and store the data are needed. In the past decade a number of researchers investigated the problem of the compression of seismic data arrays. Wavelet transforms were very successful with compression of still images. Therefore the efforts to use these techniques for seismic compression were quite natural^{6 10 16}. However, results turned out much less impressive than with image compression because of the great variability of seismic data and the noisy background inherent in it even within the same volume. Moreover, recently some researchers argued that the seismic signals are not wavelet-friendly

2 *Valery A. Zheludev, Dan D. Kosloff and Eugene Y. Ragoza*

at all ^{1 12}. This happens because of the presence of oscillatory patterns in seismic signals. By these reasons, local cosine transforms appear advantageous compared to wavelets in seismic compression. In ^{1 12 17} local cosine bases using lapped DCT-IV transforms ⁵ and adaptive segmentation were applied to the compression of 2D seismic sections.

Another important research field concerned with seismic compression is processing the compressed data. Recently the authors of the present paper reported successful application of wavelet transform to the acceleration of 3D Kirchhoff migration ²⁰.

In this paper we address an intrinsically complex problem of compression of seismic data for the purposes of rendering the subsurface structures. To efficiently visualize 3D structures, fast access to the data must be provided. Therefore the large volume is segmented into comparatively small bricks, typically of size $32 \times 32 \times 32$ samples. In order to produce a picture of a reasonable size on a computer monitor, as many bricks as possible should be placed into the memory. It is hardly possible to achieve that without a significant compression of the data bricks because of limited capacity of the memory of even powerful contemporary computers.

We present an approach that is based on application of 3D DCT-II to the data bricks and coding the produced coefficients by the SPIHT codec ¹³. An accelerated version applies DCT-II only in the vertical direction while horizontal slices are processed by 2D wavelet transforms.

2. Description of the algorithm

The compression of 3D seismic data for rendering purposes dictates a number of specific requirements to the algorithm:

- The compression must be implemented on 3D data.
- A significant compression rate for the typical stacked or migrated seismic data has to be achieved.
- Each small brick has to be compressed and uncompressed separately.
- While uncompressed, boundary discrepancies between adjacent bricks must be minimized.
- (Last but not least.) Any developed algorithm is of no value for rendering if the compression and, especially the decompression procedures are not implemented in an extremely fast manner.

The above requirements prevented us from utilizing compression algorithms, which have been reported in the literature. The main obstacle was that all the algorithms handle unsegmented seismic arrays, set aside that almost all of them are targeted on 2D seismic sections.

A typical data compression scheme consists of three stages: 1.Transform. 2.Quantization. 3.Coding. Due to specifics of our task we added one more stage: 4.Handling the boundaries between bricks.

2.1. Transform

Experiments confirmed the suggestion in ^{1 12} that the the seismic signals are trigonometric- rather than wavelet-friendly. Even while compressing unsegmented data arrays, trigonometric transforms produced better results than the results with wavelets. On the segmented arrays the advantage of trigonometric transforms over the wavelet ones becomes overwhelming. By this reason we chose to apply the discrete cosine transform (DCT).

The following two kinds of DCT are used in image compression:

- (1) DCT-II of a signal $\mathbf{f} = \{f_n\}_{n=0}^{N-1}$, $N = 2^p$ is:

$$\hat{f}^{II}(k) = b(k) \sum_{n=0}^{N-1} f_n \cos \left[\frac{k\pi}{N} \left(n + \frac{1}{2} \right) \right], \quad b(k) = \begin{cases} 1/\sqrt{2}, & \text{if } k = 0; \\ 1, & \text{otherwise,} \end{cases}$$

$$f_n = \frac{2}{N} \sum_{k=0}^{N-1} b(k) \hat{f}^{II}(k) \cos \left[\frac{k\pi}{N} \left(n + \frac{1}{2} \right) \right].$$

- (2) DCT-IV of the signal \mathbf{f} is:

$$\hat{f}^{IV}(k) = \sum_{n=0}^{N-1} f_n \cos \left[\frac{\pi}{N} \left(k + \frac{1}{2} \right) \left(n + \frac{1}{2} \right) \right],$$

$$f_n = \frac{2}{N} \sum_{k=0}^{N-1} \hat{f}^{IV}(k) \cos \left[\frac{\pi}{N} \left(k + \frac{1}{2} \right) \left(n + \frac{1}{2} \right) \right].$$

The signal $\mathbf{f}^{II} = \{\hat{f}_n^{II}\}$, which is restored from the DCT coefficients $\{\hat{f}^{II}(k)\}$ is $2N$ -periodic and

$$f_n^{II} = \begin{cases} f_n, & \text{if } n = 0, \dots, N-1; \\ f_{-n-1}, & \text{if } n = -N, \dots, -1. \end{cases} \quad (2.1)$$

Thus, it provides an even extension of the signal \mathbf{f} across its boundaries. Therefore application of this transform to segmented images and 3D volumes reduces the blocking effect. On the contrary, the restored signal $\mathbf{f}^{IV} = \{f_n^{IV}\}$ is $4N$ -periodic. It is even across $-\frac{1}{2}$ and odd across $N - \frac{1}{2}$. Therefore, application of the DCT-IV transforms to a segmented data leads to severe boundary discrepancies. However, this transform serves as a base for the so called local cosine bases ⁵, which are windowed overlapped DCT-IV transforms. These bases were successfully exploited in ^{1 12 17} for the compression of seismic data. However, because we had to compress each brick separately, we could not use these lapped DCT-IV transforms. Our final choice was to use the 3D DCT-II transforms of the whole $32 \times 32 \times 32$ brick. We recall that the JPEG image compression standard is based on 2D DCT-II transforms of 8×8 cells. Fortunately, this JPEG application stimulated development of a diversity of fast algorithms for implementation of forward and inverse DCT-II. We perform the 3D transform as subsequent application of 1D transform in each dimension. The 1D transform is implemented using a fast algorithm described by Feig and

4 *Valery A. Zheludev, Dan D. Kosloff and Eugene Y. Ragoza*

Winograd in ⁹. The forward or the inverse 3D transforms of 1000 $32 \times 32 \times 32$ bricks takes about 7 seconds on the SGI Octane R12000 workstation with usage of only one of two available processors.

A way to accelerate the implementation without a significant depletion of the performance lies in combining DCT with a wavelet transform. It is justified by the observation that the oscillatory patterns are mainly encountered along the vertical axis of a seismic volume. Therefore it seems reasonable to apply the DCT only in vertical direction and to apply a 2D wavelet transform to the horizontal slices of $32 \times 32 \times 32$ bricks. In our experiments we tested two types of wavelet transforms: the well-known 9/7 biorthogonal transforms by Daubechies ⁷ which are most frequently used in image processing (these transforms are accepted by the new JPEG 2000 image compression standard) and a recently constructed transforms which we label as 11/5 transform ³, (see Appendix). Both these transforms can be implemented in a fast lifting manner (⁸ and Appendix).

2.2. Quantization and Coding

Many contemporary schemes of image compression are based on wavelet transform of the image and the concept of zerotree coding by Shapiro ¹⁴. This concept takes advantage of the self-similarity of wavelet coefficients across the decomposition scales and their decay toward high frequency scales and relates the coefficients with quad-trees. It establishes the ancestor-descendant relationship between wavelet coefficients of different scales, which are located at the same spatial area. The coefficients of finer scales in this relation are descendants of coefficients of sparser scales. One of most efficient algorithms based on the zerotree concept is the so called SPIHT algorithm by Said and Pearlman ¹³, who added to the Shapiro's algorithm a set partitioning technique. This algorithm combines adaptive quantization of wavelet coefficients with coding. The algorithm presents a scheme for progressive coding of coefficient values when most significant bits are coded first. This property allows to flexibly control the compression rate. Moreover, the SPIHT coding procedure is very fast and, that is of special importance to us, the decoding procedure is even faster.

Recently Xiong et al. ^{18 19} argued that coefficients of the DCT-II of 8×8 cells possess some properties similar to the properties of coefficients of the 3-scale 2D wavelet transform. After labelling the 64 DCT coefficients as in Figure 1, the parent-children relationship between DCT coefficients is established as follows: the parent of a coefficient i is $\lfloor \frac{i}{4} \rfloor$, $i = 1 \dots 63$, while the set of four children associated with the coefficient j is $\{4j, 4j + 1, 4j + 2, 4j + 3\}$, $j = 1 \dots 15$. By this reason the usage of SPIHT codec in combination with the DCT-II could be efficient. Using this observation, we applied 3D SPIHT algorithm ¹¹ for coding DCT-II coefficients of the $32 \times 32 \times 32$ bricks. This is similar to a 5-scale 3D wavelet transform. Note that results in ¹⁹ of image coding with usage of wavelets were superior to the DCT results. On the contrary, in our experiments with seismic data the DCT-based

0	1	4	5	16	17	20	21
2	3	6	7	18	19	22	23
8	9	12	13	24	25	28	29
10	11	14	15	26	27	30	31
32	33	36	37	48	49	52	53
34	35	38	39	50	51	54	55
40	41	44	45	56	57	60	61
42	43	46	47	58	59	62	63

Fig. 1. A 8×8 DCT block can be treated as a depth-3 tree of coefficients .

scheme combined with the SPIHT codec produced better results.

A similar structure have the set of coefficients of the combined DCT-wavelet transforms of the $32 \times 32 \times 32$ bricks. Here we applied a 5-scale 2D wavelet transform to horizontal 32×32 slices and the DCT-II to vertical segments of seismic traces of length 32. SPIHT successfully coded such sets of coefficients.

2.3. Handling the boundaries

No one lossy algorithm which compresses fragments of a one- or multidimensional signal separately can not completely avoid the so called blocking effect, i. e. the boundary discrepancies between adjacent fragments. To reduce this effect, various schemes of extension of signals are applied⁴. Due to its symmetric properties, DCT-II implicitly performs an even extension of a signal across boundaries. Nevertheless, when a reasonable compression rate is achieved, the blocking effect becomes visible. We succeeded in substantial reduction of this effect using some kind of interpolation after decompression of adjacent bricks. To be specific, suppose that a signal \mathbf{x} of length 64 consists of two smooth fragments $\mathbf{x} = \{x_k^1\}_{k=1}^{32} \cup \{x_k^2\}_{k=33}^{64}$ and x_{32}^1 differs significantly from x_{33}^2 . Then we carry out two darning-stitches:

Mend of x_{32}^1 : We construct a cubic polynomial $P^1(t)$, which interpolates the samples $x_{29}^1, x_{31}^1, x_{33}^2, x_{35}^2$ at the points 29, 31, 33, 35 and calculate an improved sample x_{32}^3 as the value of $P^1(t)$ at the point 32:

$$x_{32}^3 = \frac{-x_{29}^1 + 9x_{31}^1 + 9x_{33}^2 - x_{35}^2}{16}.$$

Mend of x_{33}^2 : A similar procedure we apply in order to adjust the sample x_{33}^2 . We construct a cubic polynomial $P^2(t)$, which interpolates the samples $x_{30}^1, x_{32}^3, x_{34}^2, x_{36}^2$ at the points 30, 32, 34, 36 and calculate an improved sam-

6 *Valery A. Zheludev, Dan D. Kosloff and Eugene Y. Ragoza*

ple x_{33}^3 as the value of $P^2(t)$ at the point 32:

$$x_{33}^3 = \frac{-x_{30}^1 + 9x_{32}^3 + 9x_{34}^2 - x_{36}^2}{16}.$$

The rest of samples of the improved signal \mathbf{x}^3 are left equal to the corresponding samples of the signal \mathbf{x} . If the discrepancy is large, it is advisable to mend more samples at the vicinity of the break using polynomials of higher degrees and more stitches. We illustrate this procedure in Figure 2.

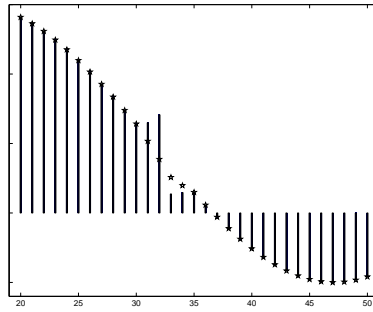


Fig. 2. Amendment procedure applied to the samples x_k , $k = 31, 32, 33, 34$.

3. Examples

We performed numerous experiments on a number of stacked SMP data volumes with the sampling rate of $25m \times 25m \times 4msec$ using various wavelet and trigonometric transforms and coding schemes. The bit allocation for the initial data storage was 32 bit per sample.

The results were evaluated both visually and through the so called peak signal to noise ratio (PSNR) in decibels which is commonly used in image processing:

$$PSNR = 10 \log_{10} \left(\frac{N M^2}{\sum_{k=1}^N (x_k - \tilde{x}_k)^2} \right) dB, \quad (3.2)$$

where N is the total number of samples in the volume, x_k is an original sample and \tilde{x}_k is a reconstructed sample, $M = \max_k(x_k) - \min_k(x_k)$.

The experiments gave a strong evidence in favor of our approach. Typically, implying the procedures described above, we achieved compression rate (CR) of the 3D volumes at 1:128 (0.25 bit per pixel) practically without visible distortions of images and blocking effects. To better illustrate results of compression, we present a few figures which display a fragment of a stacked 2D seismic section. The original fragment is displayed in Figure 3. In the following figures we illustrate results of compression of this fragment with CR=1:100 (0.32 bit per pixel) using various

transforms of the data. In all cases coefficients of the transforms were coded using the SPIHT algorithm.

Figure 4: We display result of compression of the entire non-segmented section using lapped DCT-IV with basic cell of size 32×32 (left picture) and the 9/7 2D wavelet transform (right picture). The quality of the image reconstructed from the wavelet transform is satisfactory (PSNR=29.64). But much better quality is achieved by application of the lapped DCT-IV (PSNR=30.30). The reconstructed image is hardly distinguishable from the original one. Regretfully, both these methods can not be exploited for our final purpose (does not satisfy to Requirement 3) , but it is advisable to use the lapped DCT-IV for the compression of unsegmented seismic volumes.

Figure 5: We display result of application of the wavelet transform to the segmented 32×32 section. We used the 5-scale the 9/7 wavelet transforms. Left picture displays the reconstructed image without smoothing of boundaries, (PSNR=27.72). The result with the smoothed boundaries is shown in the right picture. It is better than the former one (PSNR=28.16) but still is remarkably lower than in two previous examples. Moreover the visual appearance of the images is poor. Application of the 11/5 wavelet transform and other wavelet transforms produces a similar result.

Much better performance demonstrate the combined wavelet(horizontal)-DCT-II(vertical) transforms. In this case different wavelet transforms produced different results. In most experiments the 11/5 transform outperformed the 9/7 transform.

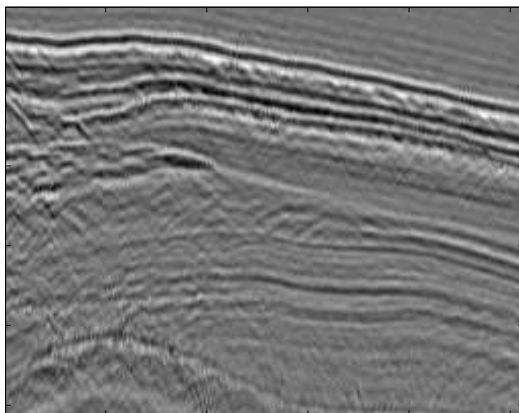
Figure 6: We display result of application of the combined 9/7 wavelet -DCT-II transform to the segmented 32×32 section. Left picture displays the reconstructed image without smoothing of boundaries, (PSNR=28.68). The result with the smoothed boundaries is shown in the right picture (PSNR=28.89).

Figure 7: We display result of application of the combined 11/5 wavelet -DCT-II transform to the segmented 32×32 section. Left picture displays the reconstructed image without smoothing of boundaries, (PSNR=28.83). The result with the smoothed boundaries is shown in the right picture (PSNR=29.11).

Note that although PSNR for the combined transforms is inferior to PSNR produced by the non-fragmented wavelet transform (Figure 4), the visual perception of the images in Figure 6 and especially in Figure 7 is much better than of the right image in Figure 4. Fine structures are revealed by the combined transforms more distinctly than by the non-fragmented wavelet transforms, not to mention the fragmented wavelet transforms. It confirms the observation that the DCT in vertical direction is more relevant to seismic data than wavelet transforms. The combination of the 5/11 transform with the DCT produces higher PSNR than such a combination with the 9/7 instead of the 5/11 transform. The visual perception of the right image in Figure 7 is a little bit better than that in Figure 6.

8 *Valery A. Zheludev, Dan D. Kosloff and Eugene Y. Ragoza*

Figure 8: The best performance among transforms of segmented 32×32 section demonstrated 2D DCT-II. Left picture displays the reconstructed image without smoothing of boundaries, (PSNR=29.65). The result with the smoothed boundaries is shown in the right picture (PSNR=29.75). This PSNR is a little bit higher than PSNR achieved by the non-segmented wavelet transform but visual perception of the image is much better. The procedure of smoothing the boundaries almost completely eliminated the blocking effect. This is our basic algorithm.



1

Fig. 3. A fragment of a stacked SMP 2D seismic section with the sampling rate of $25m \times 4msec$.

Conclusions

We present preliminary results of our investigation of compression of 3D seismic volumes for the rendering purposes. Set aside complications intrinsic in any kind of seismic compression, this special objective imposes a number of additional restrictions. Nevertheless, we obtain good compression results applying 3D DCT-II to the $32 \times 32 \times 32$ data bricks and coding the coefficients of the transforms by 3D SPIHT algorithm. This algorithm, which was originally constructed for coding coefficients of wavelet transforms, proved to be efficient for coding the coefficients of the DCT. Our technique of smoothing boundaries between segments also proved to be helpful. It removes almost completely the blocking effect. The algorithm is fast but to provide rendering efficiency, further acceleration must be achieved. One of the ways to make the transforms faster is to combine the DCT-II in vertical direction with the 2D wavelet transforms to the horizontal slices. The 3D SPIHT algorithm was used for coding the mixed sets of the coefficients. As a result we

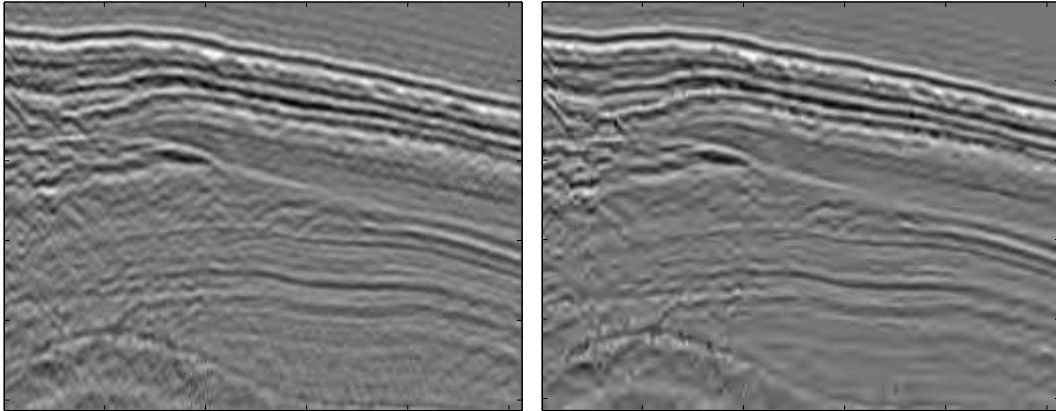


Fig. 4. Compression of the entire section. Left: lapped DCT-IV CR= 1:100, PSNR=30.30. Basic cell is 32×32 . Right: 9/7 2D wavelet transform, CR=1:100, PSNR=29.64.

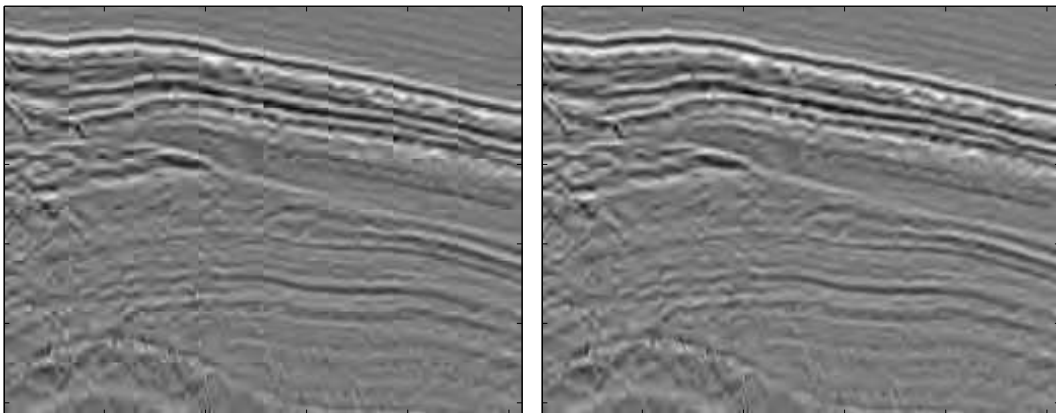


Fig. 5. Wavelet compression of the segmented 32×32 section, CR= 1:100. Left: without smoothing of boundaries , PSNR=27.72. Right: Boundaries smoothed PSNR=28.16.

achieved a noticeable reduction of the computational cost of the implementation with a minimal degradation of the quality. In most experiments the newly devised 5/11 wavelet transform outperformed the popular 9/7 transform, having the same computational complexity as the latter one.

10 *Valery A. Zheludev, Dan D. Kosloff and Eugene Y. Ragoza*

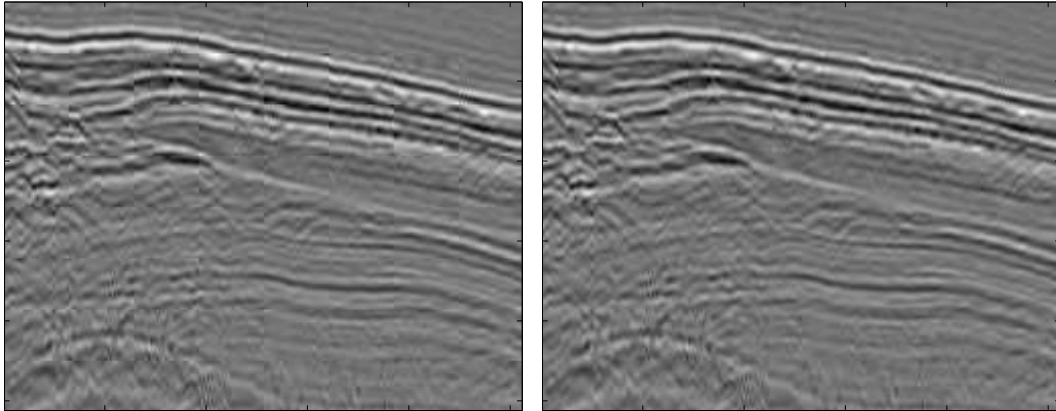


Fig. 6. Combined 9/7 wavelet(horizontal)-DCT-II(vertical) compression of the segmented 32×32 section, CR= 1:100. Left: without smoothing of boundaries , PSNR=28.68. Right: Boundaries smoothed PSNR=28.89.

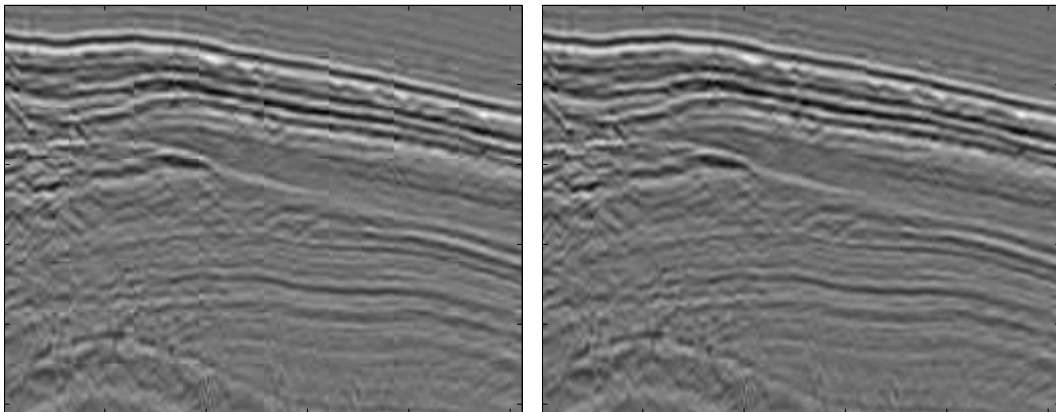


Fig. 7. Combined 11/5 wavelet(horizontal)-DCT-II(vertical) compression of the segmented 32×32 section, CR= 1:100. Left: without smoothing of boundaries , PSNR=28.83. Right: Boundaries smoothed PSNR=29.11.

Acknowledgment

The research supported by the grant of Israel Science Foundation –1999-2003, No.258/99-1, Application of the Wavelet Transform to 3D Seismic Imaging.

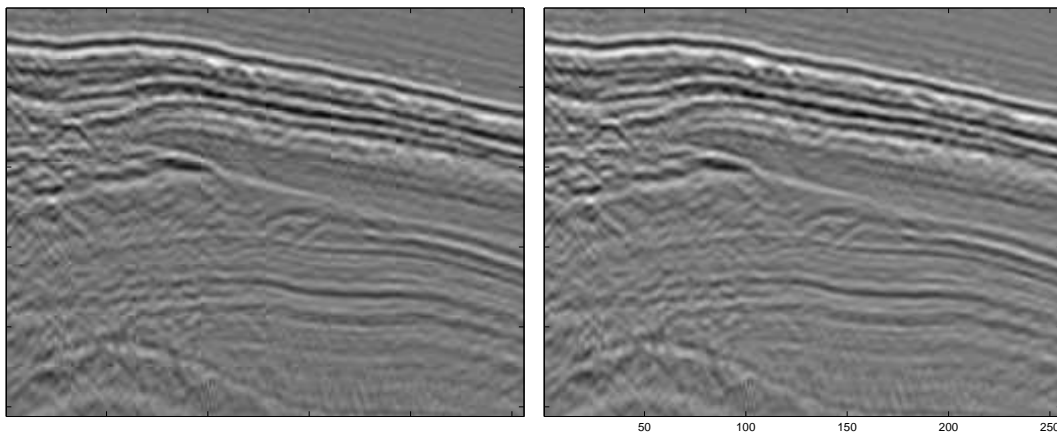


Fig. 8. 2D DCT-II compression of the segmented 32×32 section, CR= 1:100. Left: without smoothing of boundaries , PSNR=29.65. Right: Boundaries smoothed PSNR=29.75.

Appendix A. Lifting scheme of wavelet transform and the 5/11 filter

The lifting scheme of the biorthogonal wavelet transform of a signal \mathbf{x} was introduced by Sweldens¹⁵. It provides tools for the design and efficient implementation of the transforms. The lifting scheme of the decomposition of the signal $\mathbf{x} = \{x_k\}$, whose z -transform $X(z) = \sum_{k=-\infty}^{\infty} z^{-k} x_k$, consists of three steps:

Split The signal \mathbf{x} is split into even and odd subarrays: $\mathbf{e} = \{e(k) = x(2k)\}$, $\mathbf{o} = \{o(k) = x(2k+1)\}$, $k \in \mathbb{Z}$. In the z -transform domain this operation corresponds to the following relation: $E(z^2) = (X(z) + X(-z))/2$, $O(z^2) = z(X(z) - X(-z))/2$, where $E(z)$ and $O(z)$ denote the z transforms of \mathbf{e} and \mathbf{o} , respectively.

Predict The even array \mathbf{e} is used to predict the odd array \mathbf{o} . Then, the new odd array \mathbf{d} is defined as the difference between the existing array \mathbf{o} and the predicted one. To be specific, we apply some *prediction* filter U to the array \mathbf{e} , in order for the result to approximate the array \mathbf{o} . Then, we subtract this result from the array \mathbf{o} : $D(z) = O(z) - U(z)E(z)$. Provided that the filter U is properly chosen, this step results in decorrelation of the signal.

Update (lifting) The even array is updated using the new odd array that is being convolved with the *update* filter whose transfer function we denote by $V(z)/2$: $S(z) = E(z) + \frac{1}{2}V(z)D(z)$. Generally, the goal of this step is to eliminate aliasing which appears when the original signal \mathbf{x} is downsampled into \mathbf{e} . By doing so \mathbf{e} is transformed into a downsampled and smoothed (low-pass filtering) replica \mathbf{s} of \mathbf{x} . Note that \mathbf{d} is the array of details, which complements the smoothed array \mathbf{s} .

12 *Valery A. Zheludev, Dan D. Kosloff and Eugene Y. Ragoza*

The key issue in this scheme is how to properly choose the filters U and V . Varying these filter we are able to design a diversity of wavelet transforms ².

One of advantages of the lifting implementations of wavelet transforms is that the reconstruction of the signal from the wavelet coefficients \mathbf{s} and \mathbf{d} is carried out by the exact inversion of the decomposition steps:

Undo Lifting The even array $E(z) = S(z) - V(z)/2D(z)$ is restored.

Undo Predict The odd array $O(z) = D(z) + U(z)E(z)$ is restored.

Undo Split It is the standard restoration of the signal from its even and odd components. In the z domain it appears as: $X(z) = E(z^2) + z^{-1}O(z^2)$.

The lifting implementation of the biorthogonal wavelet transform is equivalent to the standard implementation ⁷, which uses the analysis low- and high-pass filters $\tilde{H}(z) = \sqrt{2}(1 + zV(z^2)\Phi(-z))$ and $\tilde{G}(z) = \sqrt{2}z\Phi(-z)$, respectively. The synthesis low- and high-pass filters are $H(z) = \sqrt{2}\Phi(z)$ and $G(z) = \sqrt{2}z^{-1}(1 - zV(z^2)\Phi(z))$, respectively.

The 11/5 filter bank. For the design of the filters U and V we use the following algorithm.

To predict a sample $o_k = x_{2k}$ we construct a cubic polynomial $P(t)$, which interpolates the samples $e_{k-1} = x_{2k-2}$, $e_k = x_{2k}$, $e_{k+1} = x_{2k+2}$, $e_{k+2} = x_{2k+4}$ at the points $2k - 2$, $2k$, $2k + 2$, $2k + 4$ and calculate the prediction as the value of $P(t)$ at the point $2k + 1$:

$$d_k = o_k - \frac{-e_{k-1} + 9e_k + 9e_{k+1} - e_{k+2}}{16}.$$

The update filter in our scheme is the halved predict filter. So, we have:

$$s_k = e_k - \frac{-d_{k-2} + 9d_{k-1} + 9d_k - d_{k+1}}{32}.$$

These detail \mathbf{d} and smoothed \mathbf{s} subarrays can be produced by the direct high-pass and low-pass filtering the signal \mathbf{x} by filters \tilde{G} and \tilde{H} of length 5 and 11, respectively, followed by downsampling. The reconstruction can be implemented by filtering the upsampled signals \mathbf{d} and \mathbf{s} by the synthesis filters G and H of length 11 and 5, respectively. However, the lifting implementation yields a significant saving of computational cost. To correctly implement the wavelet transform near boundaries, the signal must be extended across the boundaries ⁴. We do it in a manner similar to the extension, which is carried out by DCT-II as it is described in (2.1).

References

1. A. Z. Averbuch, F. Meyer, J-O. Stromberg, R. Coifman, A. Vassiliou, *Efficient Compression for Seismic Data*, IEEE Trans. on Image Processing, **10**, no. 12, (2001) 1801-1814.

2. A. Z. Averbuch, V. A. Zheludev, *A new family of spline-based biorthogonal wavelet transforms and their application to image compression*, to appear in *IEEE Trans. on Image Proc.*.
3. A. Averbuch, V. Zheludev, *Splines: a new contribution to wavelet analysis* in Proc. Conference *Algorithms for Approximation IV*, Huddersfield, England, 2001.
4. C. M. Brislawn, *Classification of Nonexpansive Symmetric extension transforms for multirate filter banks* *Applied and Computational Harmonic Analysis*, **3** (1996), No. 4, 337-357. 337-357.
5. R. R. Coifman and Y. Meyer, *Remarques sur l'analyse de Fourier a fenêtre*, *C. R. Acad. Sci.*, (1991) 259-261.
6. P. L. Donoho, R. A. Ergas, and J. D. Villasenor, *High-performance seismic trace compression*, In 65th Ann. Internat. Mtg., Soc. Expl. Geophys., *Expanded Abstracts*, (1995), 160-163.
7. I. Daubechies, *Ten lectures on wavelets*, (SIAM, 1992).
8. I. Daubechies, W. Sweldens, *Factoring wavelet transforms into lifting steps*, *J. Fourier Anal. Appl.*, **4** (1998) 247-269.
9. E. Feig and S. Winograd, *Fast algorithms for the discrete cosine transform*, *IEEE Trans. Sign. Proc.*, **40**, (1992) 2174-2193.
10. M. F. Khéne, S. H. Abdul-Jauwad, *Efficient seismic compression using the lifting scheme*, In 70th Ann. Internat. Mtg., Soc. Expl. Geophys., *Expanded Abstracts*, (2000), 2052-2054.
11. B.-J. Kim, Z. Xiong, and W. A. Pearlman, *Low Bit-Rate Scalable Video Coding with 3D Set Partitioning in Hierarchical Trees (3D SPIHT)*, *IEEE Trans. Circuits and Systems for Video Technology*, **10**, (2000) 1374-1387.
12. F. G. Meyer, *Fast compression of seismic data with local trigonometric bases*, in Proc. SPIE **3813**, *Wavelet Applications in Signal and Image Processing VII*, eds. A. Aldroubi, A. F. Laine; M. A. Unser; (1999), 648-658.
13. A. Said and W. W. Pearlman, *A new, fast and efficient image codec based on set partitioning in hierarchical trees*, *IEEE Trans. on Circ. and Syst. for Video Tech.*, **6**, (1996) 243-250.
14. J. M. Shapiro, *Embedded image coding using zerotree of wavelet coefficients*, *IEEE Trans. Sign. Proc.*, **41**, (1993) 3445-3462.
15. W. Sweldens *The lifting scheme: A custom design construction of biorthogonal wavelets*, *Appl. Comput. Harm. Anal.* **3**(2), (1996), 186-200.
16. A. Vassiliou, and M. V. Wickerhauser, *Comparison of wavelet image coding schemes for seismic data compression*, In 67th Ann. Internat. Mtg., Soc. Expl. Geophys., *Expanded Abstracts*, (1997),1334-1337.
17. Y. Wang, R.-S. Wu, *Seismic data compression by an adaptive local cosine/sine transform and its effect on migration*, *Geophysical Prospecting*, **48**, (2000) 1009-1031.
18. Z. Xiong, O. Guleryuz and M. T. Orchard, *A DCT-based embedded image coder*, *IEEE Signal Processing Letters.*, **3**, (1996) 289-290.
19. Z. Xiong, K. Ramchadran, M. T. Orchard, and Y.-Q. Zhang, *A comparative study of DCT- and wavelet-based image coding*, *IEEE Trans. on Circ. and Syst. for Video Tech.*, **9**, (1999) 692-695.
20. V. A. Zheludev, D. Kosloff, E. Ragoza *Fast Kirchhoff migration in wavelet domain*, *Exploration Geophysics*, **33**, (2002) 23-27.