ON COMPRESSION OF SEGMENTED 3D SEISMIC DATA

A. Valery A. Zheludev, B. Dan D. Kosloff

A. School of Computer Science
B. Department of Earth and Planetary Sciences
Tel Aviv University
Tel Aviv, 69978, ISRAEL

Eugene Y. Ragoza

Company Paradigm Geophysical Ltd.
Herzlia, 46120, Israel

ABSTRACT

We present preliminary investigation of compression of segmented 3D seismic volumes for the rendering purposes. Promising results are obtained on the base of discrete cosine transforms and SPIHT coding scheme.

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 [3, 6, 10]. However, results turned out much less impressive than in 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 at all [1, 7]. 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, 7, 11] local cosine bases using lapped DCT-4 transforms [2] 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 [13].

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.

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.

2. DESCRIPTION OF THE APPROACH

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

1. The compression must be implemented on 3D data.
2. A significant compression rate for the typical stacked or migrated seismic data has to be achieved.
3. Each small brick has to be compressed and uncompressed separately.
4. While uncompressed, boundary discrepancies between adjacent bricks must be minimized.
5. (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.

1. Transform The experiments confirmed the suggestion in [1, 7] 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). Since we had to compress each brick separately, we could not use the lapped DCT-4 transforms as it was done in [1, 7, 11]. However, the usage of non-overlapped DCT-4 transforms led to severe boundary discrepancies. 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 Winograd in [5]. 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.

2-3.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 [9]. 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. One of most efficient algorithms based on the zerotree concept is the so called SPIHT algorithm by Said and Pearlman [8], 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. 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 [12]. By this reason the usage of SPIHT coder in combination with the DCT-II could be efficient. Using this observation, we applied 3D SPIHT algorithm for coding DCT-II coefficients of the $32 \times 32 \times 32$ bricks. Note that results in [12] 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 scheme produced better results.

4.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 expansion of signals are applied. Due to its symmetric properties. DCT-II implicitly performs a sort of even extension of a signal through 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.

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:

$$PSNR = 10 \log_{10} \left(\frac{N\, 255^2}{\sum_{k=1}^{N} (x(k) - \tilde{x}(k))^2} \right) \, dB, \quad (1)$$

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.

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 1. 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 2: We display result of application of the wavelet transform to the entire non-segmented section. In this case PSNR=29.69. The quality of the image is satisfactory but the method can not be exploited for our final purpose (does not satisfy to Requirement 3.

Figure 3: We display result of application of the wavelet transform to the segmented 32×32 section. We used the 9/7 biorthogonal wavelets [4] which are most frequently used in image processing. Boundaries between the segments are smoothed. In this case PSNR=27.97 that is remarkably lower than in the previous scheme. Moreover the visual appearance of the image much worse than before.

Figure 4: We display result of application of the DCT-2 transform to the segmented 32×32 section. Boundaries between the segments are smoothed. CR= 1:100, PSNR=29.69, which is almost equal to that in Figure 2. The same is true about the quality of the image, but, nevertheless, some fine structures are revealed here more distinctly than in Figure 2. This is our basic algorithm.

Figure 5: The experiment which is illustrated here is similar to the experiment that was displayed in Figure 4. The only difference is that smoothing the boundaries was not applied. Here CR= 1:100, PSNR=29.57, which is insignificantly lower than in Figure 4. Generally, the quality of the image did not degrade but the blocking effect is visible in some places.

Figure 6: We display result of application of the lapped DCT-4 compression of the entire fragment. The transforms were implemented by folding over the cells of size 32×32 samples. Here CR= 1:100, PSNR=30.18 which is highest amongst all the experiments. The quality of the image is also superior to the others. The method can not be exploited for our final purpose since does not satisfy to Requirement 3, but it is advisable to use it for the compression of unsegmented seismic volumes

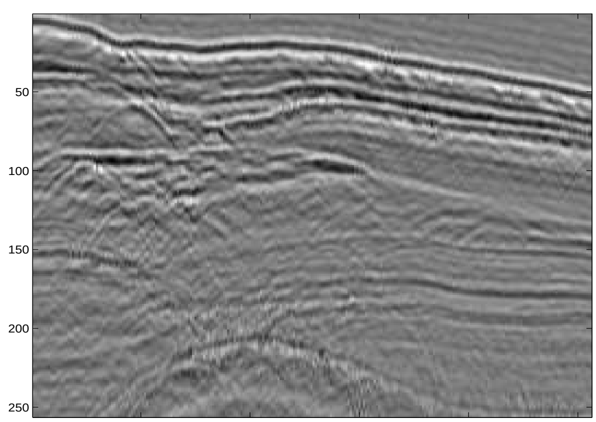

Fig. 1. A fragment of a stacked SMP 2D seismic section with the sampling rate of $25m \times 4msec$.

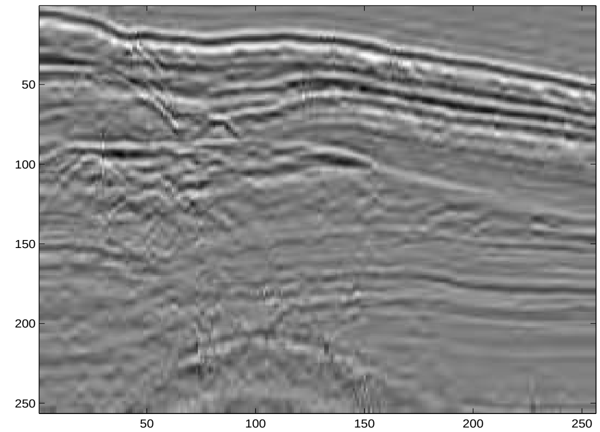

Fig. 2. Wavelet compression of the entire fragment, compression rate=1:100, PSNR=29.70.

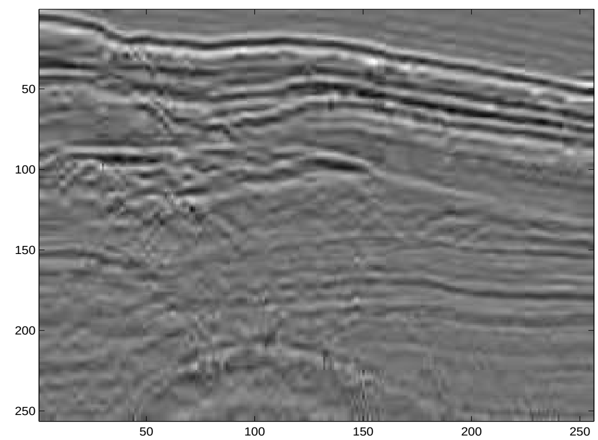

Fig. 3. Wavelet compression of the segmented 32×32 fragment, CR= 1:100, PSNR=27.97.

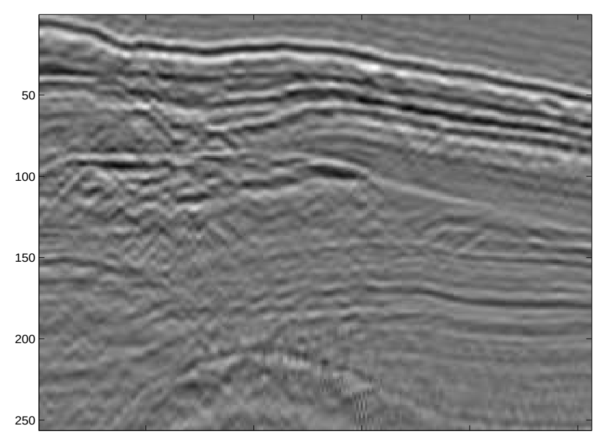

Fig. 4. DCT-2 compression of the of the segmented 32×32 fragment, CR= 1:100, PSNR=29.69. Boundaries smoothed.

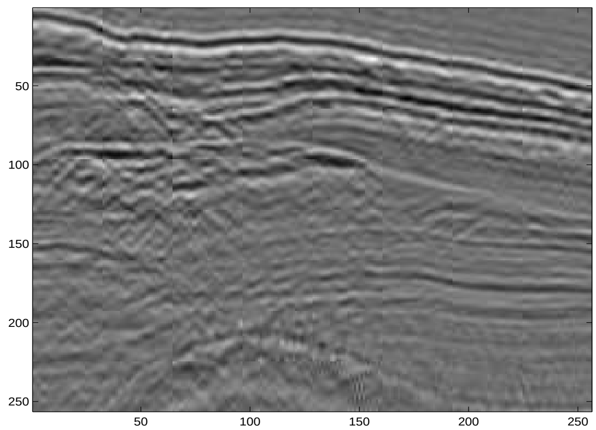

Fig. 5. The same algorithm as in Figure 4except for smoothing the boundaries, CR= 1:100, PSNR=29.57.

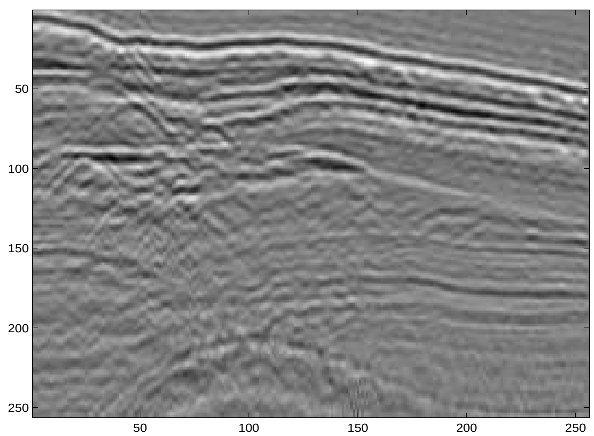

Fig. 6. Lapped DCT-4 compression of the entire fragment. CR= 1:100, PSNR=30.18. Basic cell is 32×32.

4. 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 promising results applying DCT-2 transforms to segments of data volumes and coding the coefficients of the transforms by SPIHT algorithm. The technique of smoothing boundaries between segments also proved to be helpful. The algorithm is fast but to provide rendering efficiency, further acceleration must be achieved.

5. REFERENCES

[1] A. Z. Averbuch, F. Meyer, J-O. Stromberg, R.Coifman, A. Vassiliou, ”Efficient Compression for Seismic Data”, IEEE Trans. on Image Processing, vol. 10, no. 12, pp. 1801-1814, 2001.

[2] R. R. Coifman and Y. Meyer, ”Remarques sur l’analyse de Fourier a fenêtre”, C. R. Acad. Sci., pp. 259-261, 1991.

[3] 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.

[4] I. Daubechies, 1992,Ten lectures on wavelets. SIAM.

[5] E. Feig and S. Winograd, ”Fast algorithms for the discrete cosine transform”, IEEE Trans. Sign. Proc., vol. 40, pp. 2174-2193, 1992.

[6] 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.

[7] F. G. Meyer, ”Fast compression of seismic data with local trigonometric bases,” Proc. SPIE **3813**, Wavelet Applications in Signal and Image Processing VII, (A. Aldroubi, A. F. Laine; M. A. Unser; Eds.) (1999), 648-658.

[8] A. Said and W. W. Pearlman, ”A new, fast and efficientimage codec based on set partitioning in hierarchical trees” IEEE Trans. on Circ. and Syst. for Video Tech., vol. 6, pp. 243-250, 1996.

[9] J. M. Shapiro, ”Embedded image coding using zerotree of wavelet coefficients”, IEEE Trans. Sign. Proc., vol. 41, pp. 3445-3462, 1993.

[10] 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.

[11] Y. Wang, R.-S. Wu, ”Seismic data compression by an adaptive local cosine/sine transform and its effect on migration”, Geophysical Prospecting, vol. 48, pp. 1009-1031, 2000.

[12] 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., vol. 9, pp. 692-695, 1999.

[13] V. A. Zheludev, D. Kosloff, E. Ragoza ”Fast Kirchhoff migration in wavelet domain”, Exploration Geophysics, vol. 33, pp. 23-27, 2002.