Fast Compression of Seismic Data with Local Trigonometric Bases

François G. Meyer

Yale University, New Haven, USA

ABSTRACT

Our goal in this paper is to provide a fast numerical implementation of the local trigonometric bases algorithm[1] in order to demonstrate that an advantage can be gained by constructing a biorthogonal basis adapted to a target image. Different choices for the bells are proposed, and an extensive evaluation of the algorithm was performed on synthetic and seismic data. Because of its ability to reproduce textures so well, the coder performs very well, even at high bitrate.

Keywords: Local trigonometric transforms, image compression, seismic data

1. INTRODUCTION

Wavelets with many vanishing moments yield sparse decompositions of piece-wise smooth surfaces, and are very effective for coding piece-wise smooth images.[2,3] Seismic data, however, are composed of oscillatory patterns. Rapid variations of intensity can only be described by the small scale wavelet coefficients. Long oscillatory patterns thus require many of such fine scale coefficients. Unfortunately, those small scale coefficients carry very little energy, and are often quantized to zero, even at high bit rates. While some early work on wavelet compression of seismic data sounds promising,[4–6] we advocate in this paper an approach based on LCT.

Our goal in this paper is to provide a fast numerical implementation of the best LCT algorithm, in order to demonstrate that an advantage can be gained by constructing a basis adapted to a target image. Many classes of images, or multidimensional signals, have very diffuse representations in a standard wavelet basis. Fingerprints, or seismic signals are few examples of non wavelet-friendly signals. An adapted LCT basis can often provide a very sparse representation of such images. Fast algorithms for choosing a best basis are therefore of fundamental importance. Emphasis in this paper has been placed on developing algorithms that are computationally efficient.

2. ADAPTIVE SMOOTH LOCAL COSINE TRANSFORMS

We review here the construction of smooth Localized Cosine Transforms (LCT)[1,7,8] in one dimension. We consider the more general setting where we have two biorthogonal bases.[9–11] Let $\bigcup_{n=-\infty}^{n=+\infty}[a_n, a_{n+1}[$ be a cover of $\mathbb{R}$. We define a neighborhood around each point a_n: $[a_n - \alpha_n, a_n + \beta_n]$. We only consider the two overlapping case, and α_n, and β_n, are positive numbers such that (see Fig. 1)

$$a_n + \beta_n < a_{n+1} - \alpha_{n+1}. \tag{1}$$

Let b_n be a sequence of bell functions such that:

$$\begin{aligned} b_n(x)\, b_{n-1}(2a_n - x) + b_n(2a_n - x)\, b_{n-1}(x) \neq 0 \quad &\text{if} \quad x \in [a_n - \alpha_n, a_n + \beta_n] \\ b_n(x) \neq 0 \quad &\text{if} \quad x \in [a_n + \beta_n, a_{n+1} - \alpha_{n+1}] \end{aligned} \tag{2}$$

and let

$$\theta_n(x) = \frac{1}{b_n(x)\, b_{n-1}(2a_n - x) + b_n(2a_n - x)\, b_{n-1}(x)} \tag{3}$$

The dual bells are defined as follows:

François G. Meyer is with the Department of Diagnostic Radiology, Yale University School of Medicine, 333 Cedar Street, P.O. Box 208042, New Haven, CT 06520-8042, e-mail: francois.meyer@yale.edu

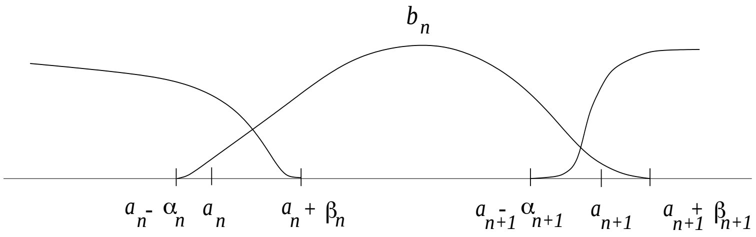

Figure 1: The bell b_n lives over the interval $[a_n - \alpha_n, a_{n+1} + \beta_{n+1}]$.

$$\tilde{b}_n(x) = \begin{cases} \theta_n(x) b_{n-1}(2a_n - x) & \text{if} \quad x \in [a_n - \alpha_n, a_n + \beta_n] \\ \dfrac{1}{b_n(x)} & \text{if} \quad x \in [a_n + \beta_n, a_{n+1} - \alpha_{n+1}] \\ \theta_{n+1}(x) b_{n+1}(2a_{n+1} - x) & \text{if} \quad x \in [a_{n+1} - \alpha_{n+1}, a_{n+1} + \beta_{n+1}] \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

Let $c_{n.k}(x)$ be the family of basis functions of the DCT-IV:

$$c_{n,k}(x) = \sqrt{\frac{2}{a_{n+1} - a_n}} \cos\left((k + 1/2)\frac{x - a_n}{a_{n+1} - a_n}\right) \tag{5}$$

We define the family

$$w_{n,k} = b_n(x)\, c_{n,k}(x) \tag{6}$$

and the dual family is defined as

$$w_{n,k} = \tilde{b}_n(x)\, c_{n,k}(x) \tag{7}$$

We have the following result

LEMMA 2.1. *$w_{j,n}$ and $\tilde{w}_{j,n}$ are Riesz biorthogonal bases:*

$$\int w_{j,n}(x)\, \tilde{w}_{k,m}(x)\, dx = \delta_{j,k} \delta_{n,m} \tag{8}$$

$\forall \mathbf{x} \in L^2(\mathbb{R})$,

$$\mathbf{x}(x) = \sum_{j,n} x_{j,n} w_{j,n}(x) \qquad \textit{with} \quad x_{j,n} = \int \mathbf{x}(x) \tilde{w}_{j,n}(x) dx \tag{9}$$

$$\mathbf{x}(x) = \sum_{j,n} \tilde{x}_{j,n} \tilde{w}_{j,n}(x) \qquad \textit{with} \quad \tilde{x}_{j,n} = \int \mathbf{x}(x) w_{j,n}(x) dx \tag{10}$$

2.1. Implementation by folding

In practice, in order to expand a function $\mathbf{x}$ into the basis $\{w_{j,n}\}$ we do not calculate the correlation between $\mathbf{x}$ and the basis functions $w_{j,n}$. Instead we transform $\mathbf{x}$ restricted to $[a_n - \alpha_n, a_{n+1} + \beta_{n+1}]$ into a smooth function onto $[a_n, a_{n+1}]$ with the adequate polarity (even around a_n, and odd around a_{n+1}), and we expand it into the basis $\{c_{n,k}\}$. To do this we fold the overlapping parts of the bells $b(x - a_n)$ and $b(a_{n+1} - x)$ back into the interval, across the endpoints of the interval, with some folding and unfolding operators. The advantage of the procedure is that we can pre-process the data with the folding operators, and then use a fast DCT to calculate the expansion into the basis $\{c_{n,k}\}$.

2.2. Choice of the bell function

The construction of the biorthogonal bases described above leaves us with more freedom to select the bells b_n than the orthogonal setting.[8] As is shown in the experiments, the bell b_n is a key parameter if the transform, and it should be carefully optimized. Because seismic data contain many oscillations, we use the bells of Matviyenko,[10] that were designed to efficiently compress trigonometric functions. The bells depend on a parameter N, and have the following expression:

$$b^N(x) = \begin{cases} \frac{1}{2}\left(1 + \sum_{n=0}^{N-1} g_n \sin(n+1/2)\pi x\right) & \text{if} \quad x \in [-1/2, 1/2] \\ \frac{1}{2}\left(1 + \sum_{n=0}^{N-1} (-1)^n g_n \cos(n+1/2)\pi x\right) & \text{if} \quad x \in [1/2 : 3/2] \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

The coefficients g_n are calculated numerically for all practical values of N in.[10]

Other optimized bells have been designed to reproduce constants,[12] or polynomials.[13]

2.3. Adaptive segmentation

We adaptively select the size and location of the windows. We define two partitions of $\mathbb{R}$, $\bigcup_{n=-\infty}^{n=+\infty}[a_n, a_{n+1}[$, and $\bigcup_{k=-\infty}^{k=+\infty}[b_m, b_{m+1}[$. We then consider the tiling obtained by the lattice cubes $[a_n, a_{n+1}[\otimes[b_m, b_{m+1}[$. We consider the separable tensor products of the bases $w_{n,k}$, and $w_{m,j}$. The sequence $w_{n,k} \otimes w_{m,j}$ is an orthonormal basis for $L^2(\mathbb{R}^2)$. We build a tree decomposition of an image. We divide the image into four sub-squares, and we consider the local cosine basis associated with this tiling. We then further decompose each square into four sub-squares, and consider the local cosine bases associated with this finer tiling. By applying this decomposition recursively we obtain a homogeneous quadtree-structured decomposition as shown in Fig. 2. Anisotropic segmentations can also be obtained as explained in.**?**

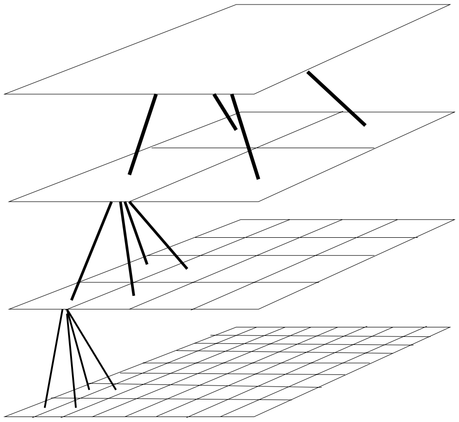

Figure 2: Within each block the image is expanded into a local DCT. The tree is then pruned in order to keep those coefficients that require the least number of bits to code.

3. CHOICE OF A COST FUNCTION

The metric defined by the objective function defines the optimality criterion. In this work, one basis is better than another if it provides a better reconstruction quality for the same number of bits spent in coding the coefficients, or if it requires less bits to achieve the same reconstruction quality. Initially Coifman and Wickerhauser[14] proposed to use the entropy of the vector $\mathbf{x}$:

$$h(\mathbf{x}) = -\sum_k \frac{|x_k|^2}{\|\mathbf{x}\|^2} \log \frac{|x_k|^2}{\|\mathbf{x}\|^2} \tag{12}$$

It is important to realize that (12) bears no connection with the entropy H $(\mathbf{x})$ of the probability distribution of the $\{x_k\}$. For instance, if all x_k are equal, then $h(\mathbf{x})$ is maximal, but the entropy of the distribution H $(\mathbf{x})$ is minimal. In practice, we have noticed that the cost function (12) is usually of little value because it fails to discover any meaningful bases. Because $h(\mathbf{x})$ is not related to the theoretical number of bits required to code the coefficients $<\mathbf{x}, w_{n,k}>$, a more meaningful measure considers the number of bits needed to approximate the image with a given error. Ramchandran and Vetterli[15] followed this path, and wedded the bit allocation algorithm of Shoham and Gersho[16] to the best basis algorithm.[14] Unfortunately their approach is extremely computationally intensive: the problem in[15] involves 3 layers of non linear approximations, only one of which lends itself to a fast algorithm). Finally, we note that the results published in[15] correspond to hypothetical compression rates, since the first order entropy was chosen to measure the rate.

Instead of using the rate distortion framework, we designed a cost function that returns an estimate of the actual rate achieved by each node. The cost function mimics the actual scalar quantization, and entropy coding, which are presented in Section 5.1. However, the cost function is much faster to compute. It is composed of two complementary terms:

- $c_1(\mathbf{x})$, the cost of coding the sign and the magnitude of the non zero output levels of the scalar quantizer,
- $c_2(\mathbf{x})$, the cost of coding the locations of the non zero output levels (significance map),

If $\mathbf{x} = \{x_k\}$ is an N dimensional vector, a first order approximation of the cost of coding the magnitude of the output levels $\{|Q(x_k)|\}$ is given by the number of bits needed to represent the set $\{|Q(x_k)|, k/Q(x_k) \neq 0\}$:

$$c_1(\mathbf{x}) = \sum_{k/Q(x_k)\neq 0} \max\left(\log_2 |Q(x_k)|, 0\right) \tag{13}$$

A fast implementation of c_1 can be devised using the standard representation of floating numbers: $\log_2 |Q(x_k)|$ is obtained using a logical “AND” and a mask. The sign of $Q(x_k)$ is extracted in a similar manner.

The second term provides an estimate of the number of bits needed to code the significance map. This term is calculated using the first order entropy of a Bernoulli process: each coefficient x_k is significant with a probability p, and we assume that the significance of the coefficients are independent events. This memoryless property is obviously not true, but since we do not take advantage of the correlation across subbands in the entropy coding, this hypothesis yield a good estimate of our actual coding cost. We get

$$c_2(\mathbf{x}) = -N\left(p\log_2(p) + (1-p)\log_2(1-p)\right) \tag{14}$$

The computation of the cost function requires to quantize the coefficients. We use the scalar quantizer described in section 5.1. An initial estimate of the quantization step is required to compute the cost function. This estimate can be refined if needed, after a first compression. We noticed that the best basis selected with this cost function varies with the particular choice of this initial quantization step, but the overall compression result varies slowly as a function of this parameter.

4. FAST DCT-IV

We use an implementation of the DCT-IV that requires to calculate a FFT of half length. It is straightforward to check that the DCT-IV coefficients, $\hat{x}(j), \quad j = 0, \ldots, N-1$ of the sequence $x(n), \quad n = 0, \ldots, N-1$ are given by:

$$\begin{aligned} \hat{x}(j) =& \mathrm{Re}\left(e^{-\dfrac{ij\pi}{2N}} \sum_{n=0}^{N/2-1} y(n) e^{-\dfrac{2i\pi jn}{N}}\right) \\ \hat{x}(N-j-1) =& -\mathrm{Im}\left(e^{-\dfrac{ij\pi}{2N}} \sum_{n=0}^{N/2-1} y(n) e^{-\dfrac{2i\pi jn}{N}}\right) \end{aligned} \tag{15}$$

where the sequence $y(n)$ is computed as follows:

$$y(n) = \left(x(2n) + i\,x(N-2n-1)\right) e^{-\dfrac{i(n+1/4)\pi}{N}} \tag{16}$$

We can then use one of the many fast implementation of the FFT to compute the DCT-IV.

5. QUANTIZATION AND ENTROPY CODING

After several experiments we have noticed that the most efficient way of organizing the coefficients consists in scanning them by increasing frequency. We divide each LCT block into a fixed number of frequency subsets ; the coefficients in a subset have similar two-dimensional frequency. We then gather from all the LCT blocks all the coefficients that are in the same subset. We start with the subset with the smallest frequency, continue until we scan the subset with the highest frequency.

5.1. Laplacian based scalar quantization

Within each frequency subset the distribution of the cosine coefficients is approximated with a Laplacian distribution. As shown in[17] Generalized Gaussian models provide a better fit than Laplacian models, but they only outperform the Laplacian models by a small margin.[17] Furthermore, Laplacian distribution yields tractable computations of the optimal entropy constrained scalar quantizers,[18] as well some near optimal scalar quantizer.[18] A particularly efficient near optimal scalar quantizer relies on the three ingredients:

- $[-\Delta+\delta, \Delta-\delta]$, the symmetric dead-zone ,
- Δ, the quantizer step size,
- δ, the reconstruction offset

The principle of the quantizer is shown in Fig 3. The optimal (for mean square error) reconstruction offset is given by[18] : $\delta = 1 - \Delta\frac{e^{-\Delta}}{1-e^{-\Delta}}$. The theoretical performance of the quantizer is very close to the optimal behavior of the entropy constrained scalar quantizer,[18] but has a much simpler rule for reconstruction. Finally, we apply a dichotomic search to find the optimal value of Δ in order to exactly match the budget.

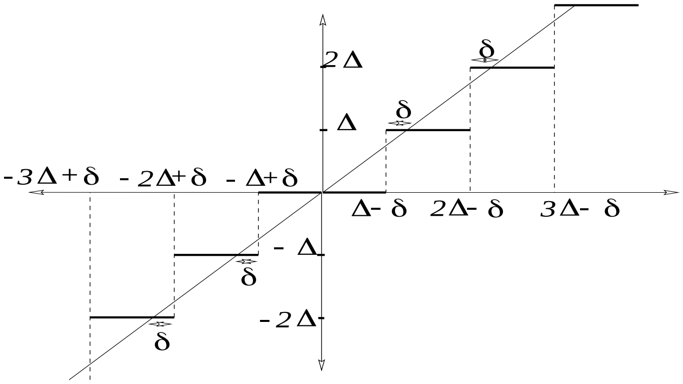

Figure 3: Scalar quantizer, with a dead zone.

5.2. Ordering of the coefficients

After scalar quantization, the positions of the non-zero output levels are recorded in a significance map. Because large output levels often appear in clusters, one can exploit the spatial correlation between neighboring pixels for the lossless compression of the significance map. Correlations also exist across frequency subsets. In the case of the wavelet basis several authors have exploited these correlations to describe with quadtrees large regions where the quantized coefficients are equal to zero.[19,2,3] Such partitioning techniques take full advantage of the self similar structure of natural images across scales.[20] While correlations also exist across frequency in a LCT basis, in general we cannot condition the probability of significance of a given pixel on the probability of significance of the pixels in its parent frequency. Because the LCT basis is adapted to the frequency content of the target image, we can expect significant high frequency coefficients.

Another technical difficulty comes from the fact that with a general LCT tree we usually cannot define the “parent” frequency. Some attempts have been made to use zerotrees with DCT, but the approach in[21] effectively requires a wavelet-like structure for the definition of the frequency subsets. Because of all these issues, we limit the context to be a spatial context.

The spatial context is defined as follows. For each frequency subset, we scan the pixels inside that subset using a Hilbert space filling curve (see Fig. 4). This self similar curve exploits the coherence in two dimensions, and it guarantees that[22]: (i) all pixels in an area are visited before moving on to another area, (ii) the sequentially visited pixels are always adjacent. The spatial context is then defined as the n_C pixels that appear before the current pixel in the Hilbert scan. In the experiments we use $n_C = 3$. We then use a n_C order arithmetic coder to encode the Hilbert scan of the significance map. This efficient context modeling permits to condition the probability of significance of a given pixel on the probability of significance of its neighbors.

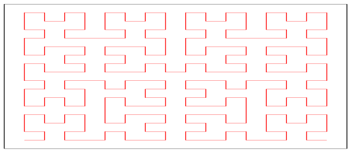

Figure 4: Hilbert space filling curve.

5.3. Entropy coding

The significance map is coded with a n_C order arithmetic coder. The signs of the output levels are not entropy coded, and are simply packed. The magnitude of the output levels are variable length encoded, using an arithmetic coder to encode the length. The best basis geometry is described by a quadtree. We code the quadtree, with an adaptive arithmetic coder.

6. EXPERIMENTS AND DISCUSSION

We implemented the Fast Local Cosine Transform (FLCT) coder and decoder, and an actual bit stream is generated by the coder. Note that for all experiments we generated a compressed file with a size equal to the targeted budget. The FLCT code that was used for the experiments is available from http : //noodle.med.yale.edu/ ∼ meyer/profile.html. For all experiments we have used the optimized bell of Matviyenko.[10] We present the results of the compression algorithm, using two test images: 192 × 640 "image-1", and 192 × 640 "image-2". Image-1 is a synthetic image, and image-2 is an image extracted from a seismic data set. Both are difficult to compress because they contain non planar oscillatory patterns. For both images we investigated the influence of N, the parameter of the bell, and α_n, β_n, the size of the bell, on the performance of the compression. We work with 32 bit images, and we define the Peak Signal to Noise Ratio (PSNR) of the compressed image I_c as $PSNR = 10 \log_{10} \frac{\max_{i,j} |I(i,j)|^2}{\frac{1}{N^2} \sum_{i,j=0}^{N-1} |I(i,j) - I_c(i,j)|^2}$.

Synthetic data

Fig. 5 shows the original synthetic image. Fig. 6 shows the geometry of the best LCT basis chosen for a compression of 32. Because the LCT will perform a local approximation of the image to a planar wave, the best basis algorithm automatically divides the image into smaller boxes in regions where the curvature changes rapidly. Because our best basis algorithm minimizes the amount of memory, it explores the tree with a "depth-first approach. The selection of the optimal segmentation proceeds from the upper left corner to the lower left corner. As a result, the optimal segmentation needs not be symmetric, even if the original image is symmetric, as shown in Fig. 6. Fig. 7 shows the decoded image for a compression ratio of 32, and with a PSNR=41.98 dB.

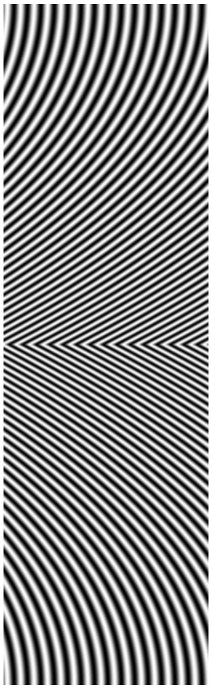

Figure 5: Original data set (size: 192 × 640.

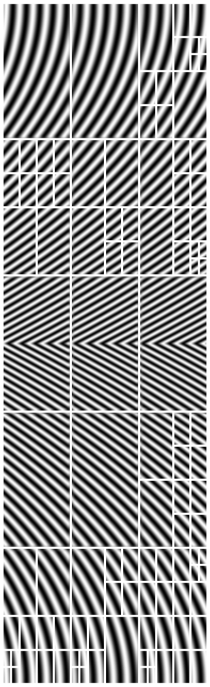

Figure 6: Best basis, $N = 2$, $\alpha_n = \beta_n = 4$, compression ratio = 32,

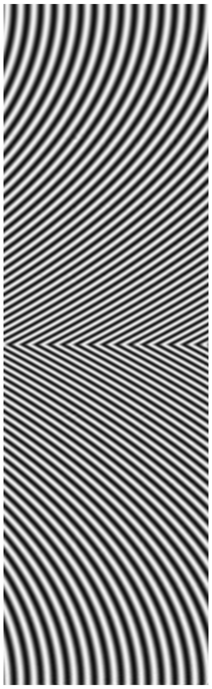

Figure 7: Decoded image, compression ratio = 32, PSNR = 41.98 dB, $N = 2$, $\alpha_n = \beta_n = 4$,

In Fig. 8 we study the influence of the parameter of the bell on the overall compression. The size of the support of the bell was kept constant ($\alpha_n = \beta_n = 4$) for all bells. We show only the first 6 best bells. The optimal bell depends on the compression ratio: for a compression from 8 to 48 $N = 2$ is optimal, and for a compression of 48 to 128, $N = 1$ is optimal. This is in contradiction with the theoretical results, and experiments on 1-D signals performed by Matviyenko.[10]

In Fig. 9 we study the influence of the size of the bell on the overall compression. The parameter of the bell was kept constant ($N = 2$) for all interval sizes. The optimal choice is always $\alpha_n = \beta_n = 4$. For such a size, the bell covers half of the interval on either side at the lowest level.

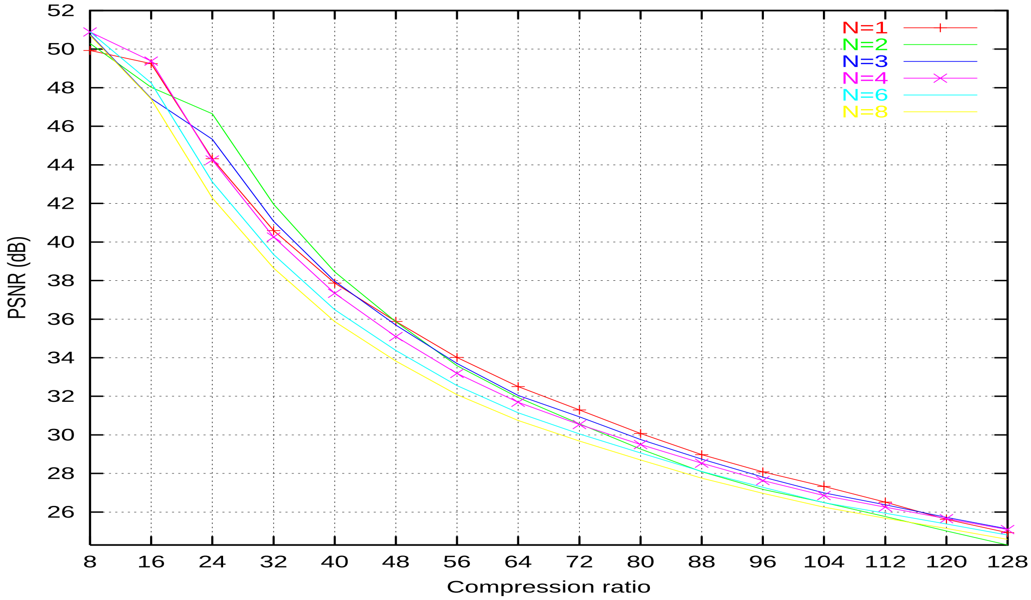

Figure 8: Influence of N, the parameter of the bell. α_n, β_n are kept equal to 4.

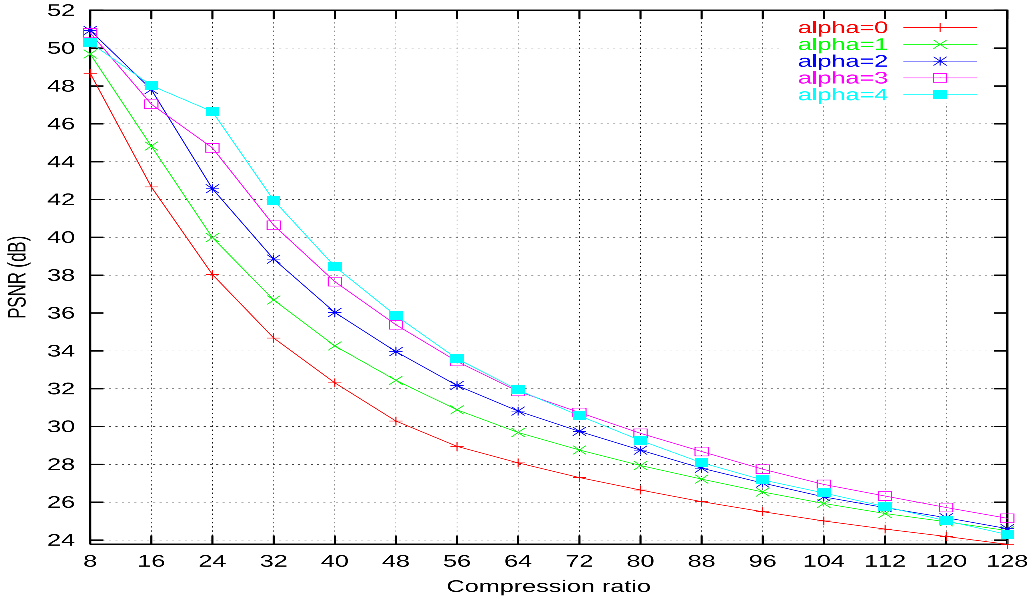

Figure 9: Influence of the size of the bell: $\alpha_n = \beta_n$. N is kept equal to 2.

Real data

Fig. 10 shows the original image from a seismic data set. Fig. 11 shows the geometry of the best LCT basis chosen for a compression of 32. As mentioned above the best basis algorithm automatically chooses smaller boxes in regions where the curvature changes rapidly. Fig. 12 shows the decoded image for a compression ratio of 32, and with a PSNR=22.03 dB.

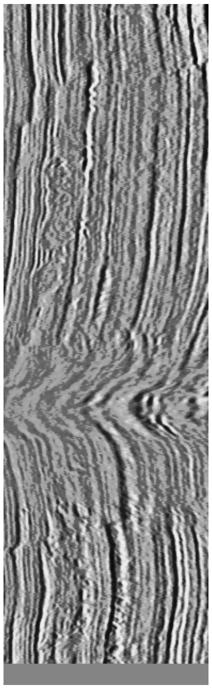

Figure 10: Original data set (size: 192×640.

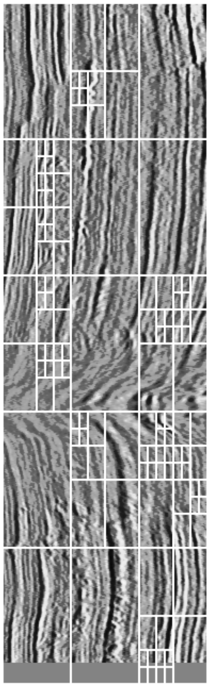

Figure 11: Best basis, $N = 4$, $\alpha_n = \beta_n = 2$, compression ratio = 32,

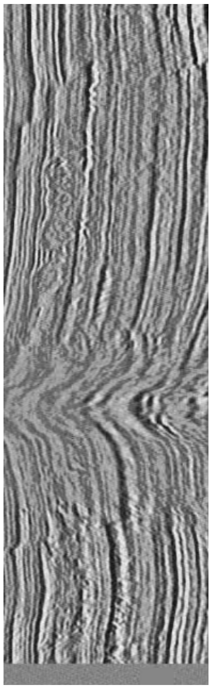

Figure 12: Decoded image, compression ratio = 32, PSNR = 22.03 dB, $N = 4$, $\alpha_n = \beta_n = 2$,

In Fig. 13 we study the influence of the parameter N of the bell on the overall compression. The size of the support of the bell was kept constant ($\alpha_n = \beta_n = 1$) for all bells. We show only the first 4 best bells. The optimal bell is reached for $N = 4$, but the PSNR is not very sensitive to the choice of N. In Fig. 14 we study the influence of the size of the bell on the overall compression. The parameter of the bell was kept constant ($N = 4$) for all interval sizes. The optimal choice is always $\alpha_n = \beta_n = 1$. This choice corresponds to a very steep bell.

A comparison of the LCT algorithm with a wavelet packet coder[23] would permit to better understand the advantage of the LCT over wavelet packets. We are currently investigating the relative merits of the two algorithms, and we will soon be able report our results.

7. ACKNOWLEDGMENTS

The author thanks Anthony Vassiliou for providing the seismic data.

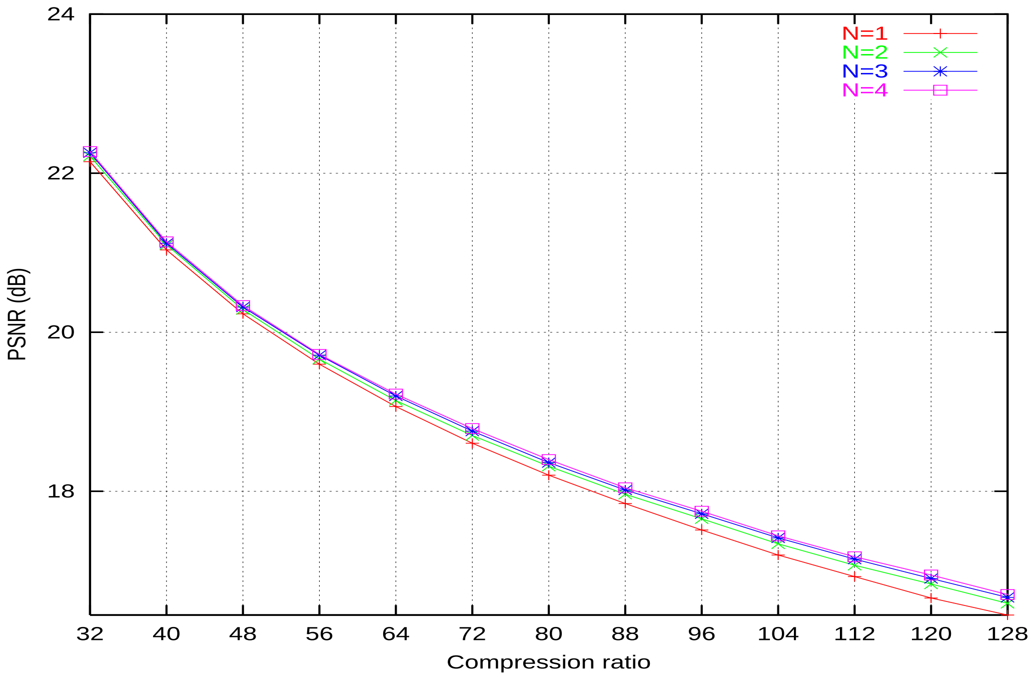

Figure 13: Influence of N, the parameter of the bell. α_n, β_n are kept equal to 1.

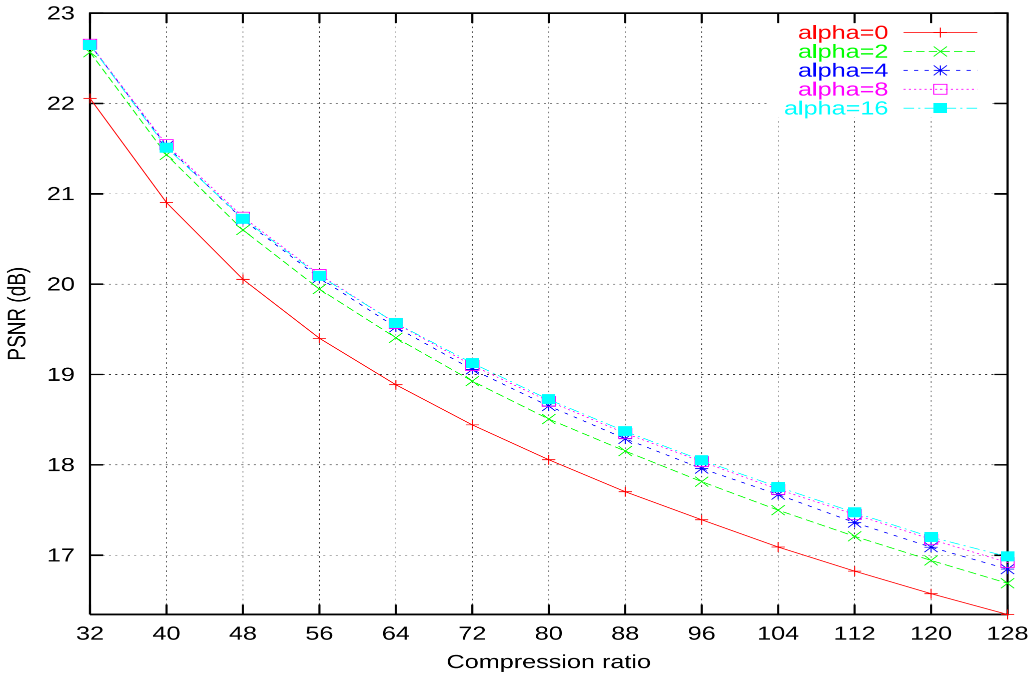

Figure 14: Influence of the size of the bell. N is kept equal to 4.

REFERENCES

1. P. Auscher, G. Weiss, and M. Wickerhauser, “Local sine and cosine bases of Coifman and Meyer,” in *Wavelets-A Tutorial*, pp. 237–56, Academic Press, 1992.
2. A. Said and W. Pearlman, “A new fast and efficient image codec based on set partioning in hierarchical trees,” *IEEE Trans. Circuits. Syst. Video Tech.* **6**, pp. 243–50, June 1996.
3. J. Shapiro, “Embedded image coding using zerotrees of wavelet coefficients,” *IEEE Trans. on Signal Processing* **41**, pp. 3445–3462, Dec. 1993.
4. R. Ergas, P. Donoho, and J. Villassenor, “High-performance seismic trace compression,” in *Soc. Expl. Geophys., 65th Intern. Conv. Soc. Expl. Geophys.*, 1995.
5. E. Reiter and P. Heller, “Wavelet transformation-based compression of nmo-corrected cdp gathers,” in *Soc. Expl. Geophys., 64th Intern. Conv. Soc. Expl. Geophys.*, 1994.
6. A. Vassiliou and M. Wickerhauser, “Comparison of wavelet image coding schemes for seismic data compression,” in *Soc. Expl. Geophys., 67th Intern. Conv. Soc. Expl. Geophys.*, 1997.
7. R. Coifman and Y. Meyer, “Remarques sur l’analyse de Fourier à fenêtre,” *C.R. Acad. Sci. Paris I* , pp. 259–61, 1991.
8. M. Wickerhauser, “Smooth localized orthonormal bases,” *C.R. Acad. Sci. Paris I* , pp. 423–427, 1993.
9. C. Chui and X. Shi, “Characterization of biorthogonal cosine wavelets,” *J. Fourier Anal. Appl.* **3**(5), pp. 560–75, 1997.
10. G. Matviyenko, “Optimized local trigonometric bases,” *Applied and Computational Harmonic Analysis* **3**, pp. 301–23, 1996.
11. F. Meyer and R. Coifman, “Brushlets: a tool for directional image analysis and image compression,” *Applied and Computational Harmonic Analysis* **4**, pp. 147–187, 1997.
12. B. Jawerth and W. Sweldens, “Biorthogonal local trigonometric bases,” *J. Fourier Anal. Appl.* **2**(2), pp. 109–133, 1995.
13. K. Bittner, “Error estimates and reproduction of polynomials for biorthogonal local trigonometric bases,” *Applied and Computational Harmonic Analysis* **6**, pp. 75–102, 1999.
14. R. Coifman and M. Wickerhauser, “Entropy-based algorithms for best basis selection,” *IEEE Trans. Information Theory* **38**, pp. 713–718, March 1992.
15. K. Ramchandran and M. Vetterli, “Best wavelet packet bases in a rate-distortion sense,” *IEEE Trans. Image Processing* , pp. 160–175, April 1993.
16. Y. Shoham and A. Gersho, “Efficient bit allocation for an arbitrary set of quantizers,” *IEEE Trans. Acoustics, Speech, and Signal Process.* **36**, pp. 1445–1453, Sept. 1988.
17. K. Birney and T. Fischer, “On the modeling of DCT and subbdand image data for compression,” *IEEE Trans. on Image Process.* **4**(2), pp. 186–193, 1995.
18. G. Sullivan, “Efficient scalar quantization of exponential and Laplacian random variables,” *IEEE Trans. Information Theory* **42**, pp. 1365–1374, Sept. 1996.
19. A. Lewis and G. Knowles, “Image compression using the 2-D wavelet transform,” *IEEE Trans. Image Processing* **1,(2)**, pp. 244–250, 1992.
20. G. Davis, “A wavelet-based analysis of fractal image compression,” *IEEE Trans. Image Processing* **7**(2), pp. 141–154, 1998.
21. G. Davis and S. Chawla, “Image coding using optimized significance tree,” in *IEEE Data Compression Conference -DCC’97*, pp. 387–396, 1997.
22. T. Bially, “Space filling curves: Their generation and their application to bandwith reduction,” *IEEE Trans. Inf. Theor.* **15,(6)**, pp. 658–664, 1969.
23. F. Meyer, A. Averbuch, and J.-O. Strömberg, “Fast adaptive wavelet packet image compression,” *IEEE Transactions on Image Processing* , pp. 792–800, May 2000.