An LCT-wavelet based algorithms for data compression

Amir Z. Averbuch[1], Valery A. Zheludev[1], Moshe Guttmann[1], Dan D. Kosloff[2]
[1]School of Computer Science
Tel Aviv University, Tel Aviv 69978, Israel

[2]Department of Earth and Planetary Sciences
Tel Aviv University, Tel Aviv 69978, Israel

Abstract

We present an algorithm that compresses two-dimensional data arrays, which are piece-wise smooth in one direction and have oscillatory events in the other direction. Seismic and hyper-spectral data have this mixed structure. The transform part of the compression process is an algorithm that combines wavelet and the local cosine transform (LCT). The quantization and the entropy coding parts in the compression process were taken from the SPIHT codec. To efficiently apply the SPIHT codec to a mixed coefficients array, reordering of the LCT coefficients takes place. When oscillating events are present in different directions as in fingerprints or when the image comprises a fine texture, the 2D LCT with reordering of coefficients is applied. These algorithms outperform algorithms that are solely based on the 2D wavelet transforms and SPIHT coding including JPEG2000 compression standard. The algorithms retain fine oscillating events including texture even at a low bitrate. Its compression capabilities are also demonstrated on multimedia images that have a fine texture. The wavelet part in the mixed transform of the hybrid algorithm utilizes the library of Butterworth wavelet transforms that outperforms the 9/7 biorthogonal wavelet library.

1 Introduction

3D seismic data and hyper-spectral images are considered to be very large datasets. Efficient algorithms are needed to compress these datasets in order to store or transmit them from planes, boats and satellites to base stations for further analysis. The compression ratio should be as high as possible without damaging the analysis capabilities that take place after decompression. The compression of these datasets should preserve fine details. Since wavelet transforms have successful history in achieving high compression ratios for still images, then these techniques were ported to handle seismic compression [21, 26, 34]. However, the outcomes were less satisfactory due to the great variability of

seismic data, the inherent noisy background and their oscillatory nature. Moreover, some researchers argued that seismic signals are not wavelet-friendly [1, 27]. This happens because of the oscillatory patterns that are present in seismic data.

These oscillatory patterns can be properly handled by the cosine transforms. The local cosine transform (LCT) [1, 27, 35], which uses the lapped DCT-IV transform (LCT) [19] with adaptive partition, was applied to compress 2D seismic sections. The LCT catches well oscillatory patterns. Although these methods provided an excellent energy compaction, the bottleneck was the absence of an efficient quantization–coding scheme comparable with the schemes designed for the wavelet transforms such as EZW [32] and SPIHT [29] or JPEG 2000 standard [25].

It was observed in [36] that the cosine transforms coefficients can be rearranged in a way that their structure becomes similar to the structure of wavelet transform coefficients. This observation paved the way to combine the cosine transforms with the wavelet–oriented coding schemes (WOCS) [9, 13]. The DCT-II –based algorithm coupled with the SPIHT coder for compressing the segmented seismic cube was presented in [8]. An embedded tarp filtering combined with the classification of the reordered DCT-II coefficients (TCE) was applied to the image compression is presented in [10].

Seismic data has different structure in its vertical and horizontal directions. While horizontal structure is piece-wise smooth, the vertical traces comprise oscillatory patterns. To some extent, the same can be said about the hyper-spectral data where each spatial pixel is represented by a vector (also called multipixel) of the intensities in all the available wavebands (typically, ≈ 200). Compression of hyper-spectral data should retain the spectral characteristic features of the multipixels. In seismic compression, the oscillating events, which bear the information on the subsurface layers, must be retained.

Seismic and hyper-spectral images have different structures in different directions. This triggered the idea to apply different transform in different direction. One realization of this approach was presented in [12] where 3D hyper-spectral data cubes were compressed by the application of the 2D wavelet transform in the spatial direction and the Karhunen-Loéve transform in the spectral direction. The transforms coefficients were coded by the JPEG2000. Another scheme for the 3D hyper-spectral data compression was presented in [11] where different types of wavelet transforms were applied to the spatial and spectral directions, which were followed by JPEG2000 coding.

As was mentioned above, compression of seismic and hyper-spectral data pursues two goals: 1. Transmission of the data from the capturing devices to the analysis station (so-called onboard compression) and 2. Storage of huge data volumes (on-the-ground compression). In this paper, we focus on the compression of 2D rectangles, which comprise one spatial direction and one spectral (or in-depth) direction. Such rectangles result from the data acquisition schemes in hyper-spectral imaging and in seismic exploration. Thus, the proposed algorithms can be directly used for the onboard compression.

For the on-the-ground compression, 3D methods produce better results but our scheme can be readily extended to the 3D setting.

We exploit the fact that wavelets provide a sparse representation of such a mixed-structure data in the horizontal direction while the LCT handles well oscillatory patterns. Thus, we propose to apply the wavelet transform to the horizontal direction and the LCT to the vertical direction.

The proposed transform produces mixed LCT and wavelet coefficients. In order to be submitted to a WOCS, the coefficients array should be reshaped to mimic the structure of a 2D array of wavelet coefficients. To achieve it, we supply the LCT coefficients with spatial meaning by partitioning the data in the vertical direction. The joint coefficients array is organized in a tree-like way by reordering the LCT coefficients. Therefore, this schemes is called an *hybrid compression algorithm* (HCA). If there are oscillations in both directions, then the LCT is applied to both directions followed by reordering of the transform coefficients and then the WOCS is applied. This method is abbreviated as RLCTA.

We compared the performance of the HCA and RLCTA using the SPIHT codec with the performance of JPEG2000 and so also with the performance of 2D wavelet- based algorithms using SPIHT. In most experiments with the hyper-spectral and seismic data HCA and RLCTA using the SPIHT was an advantage over JPEG2000 and over wavelet- based algorithms with the SPIHT. In addition, our experiments demonstrated that the biorthogonal wavelet transforms, which are based on the infinite impulse response (IIR) Butterworth filters [2, 3], provide better compressed images than what the transforms based on the popular 9/7 transforms [18], which, in particular, are utilized in the JPEG2000 standard. This is true for the 2D wavelet transforms as well as for the hybrid transforms.

Another successful application of the HCA and RLCTA is compression of fingerprints images. The FBI uses the compression standard [6] that is based on the 9/7 2D wavelet transform. This transform, which uses finite impulse response (FIR) filters of length 7 and 9, is included also in JPEG2000 image compression standard. The HCA and RLCTA produce higher PSNR results and retain better the structure of the fingerprints compared to the outputs from the application of 2D wavelet transforms and JPEG2000.

For multimedia type images, which are relatively smooth such as "Lena" and "Fabrics", the HCA performance was close (sometimes even inferior) to the performance of the 2D wavelet- based algorithms. On the other hand, the HCA and RLCTA outperform the 2D wavelet based algorithms on images that have fine texture or oscillating patterns such as "Barbara", "Elaine", "Wood fence", etc. The HCA and RLCTA retain the texture and oscillations even in a very low bit rate.

The paper is organized as follows. In Section 2, we recall some known facts about LCT, wavelet transforms and SPIHT coding. Section 3 describes the hybrid algorithm. Section 4 presents the experimental results on compression of seismic sections, hyper-spectral data, fingerprints and multimedia images.

2 Preliminaries

2.1 Local cosine transform (LCT)

The discrete cosine transform of type IV (DCT-IV) ([20]) of the signal $\mathbf{f}$ and its inverse are defined as follows:

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

The basis signals

$$\left\{\cos\left[\frac{\pi}{N}\left(k+\frac{1}{2}\right)\left(n+\frac{1}{2}\right)\right]\right\}, \quad n = 0, N-1,$$

of DCT-IV are even on the left side with respect to $-\frac{1}{2}$ and odd on the right side with respect to $N-\frac{1}{2}$. Therefore, direct application of the DCT-IV to a partitioned data leads to severe boundary discrepancies. However, this transform serves as a base for the so called local cosine bases [19], which are the windowed lapped DCT-IV transforms. These bases were successfully exploited in [1, 4, 27, 31, 35] for image compression in general and seismic data compression in particular.

Assume we have a signal $\mathbf{S} \stackrel{\text{def}}{=} \{s_k\}_{k=0}^{N-1}$ and some partition P of the interval $0 : N-1$. The idea behind the lapped DCT-IV transform, also called the local cosine transform (LCT) of a signal, is to apply overlapped bells to adjacent sub-intervals. Then, the overlapping parts are folded back to the sub-intervals across the endpoints of the sub-intervals and each sub-interval is expanded by the application of the DCT-IV transform. In the reconstruction phase, the transform coefficients are unfolded. For details, see [1, 4, 19, 30]. It is important that LCT produces no discrepancies between adjacent intervals. The choice of a bell is discussed in [4, 31]. The bell we chose is given in section 4. There are fast algorithms that implement the LCT.

2.2 Wavelet transforms

Currently, wavelet transforms constitute a recognized tool for image processing applications. In particular, they have gained a proven success in image compression. We summarize here some well known facts that are needed later.

The multiscale wavelet transform of a signal is implemented via iterated multirate filtering by a pair L (low-pass) and H (high-pass) filters. Once implemented, the wavelet transform of a signal produces a partition of the Nyquist frequency band in a logarithmic mode. The diagram of a three-scale wavelet transform and the layout of the transform coefficients are displayed in Fig. 2.1.

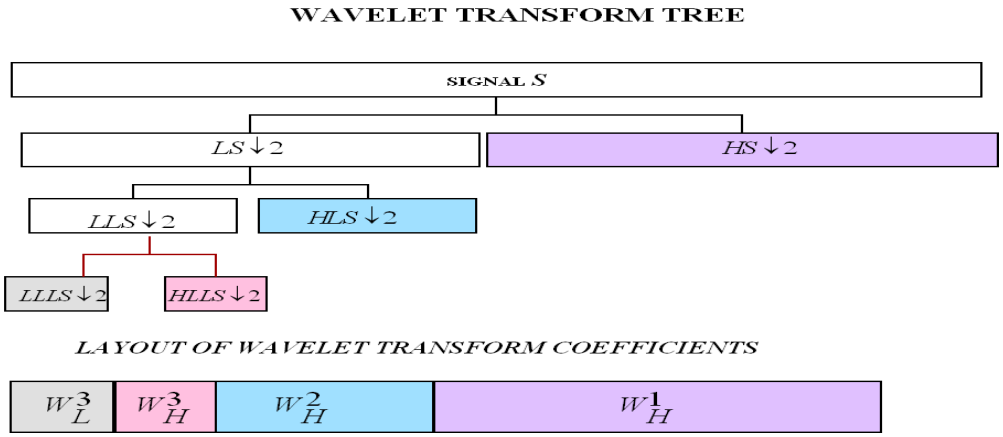

Figure 2.1: Three-scale wavelet transform and the layout of the transform coefficients

On the other hand, the wavelet coefficients have a spatial meaning. A transform coefficient from a certain decomposition scale is a correlation coefficient of the signal with a translation of the waveform (wavelet) related to this scale. The wavelets from the first decomposition scale are translated by two-sample steps, the steps for the second scale are four samples and so on.

The wavelets $\psi^j(k),\ j = 1, ...4,\ \Phi^4(k)$ and their frequency responses are displayed in Fig. 2.2.

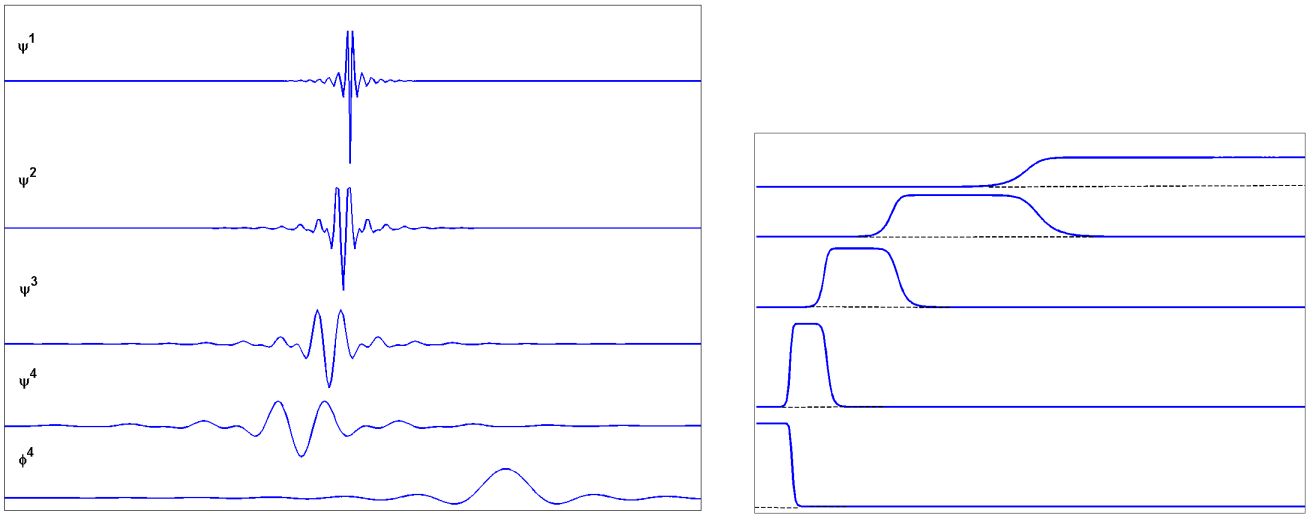

Figure 2.2: Left: the High-frequency wavelets $\psi^j(k),\ j = 1, ...4$, and the low-frequency wavelet $\Phi^4(k)$. Right: their frequency responses.

2.2.1 Tree structure of the wavelet transform coefficients

Assume that the number of scales is $J = 3$. The high-frequency coefficient h_m^3 from the coarsest scale is related to the waveform ψ_{8m+4}^3, which is centered around the sample $8m + 4$. The high-frequency coefficients from the second scale h_{2m}^2 and h_{2m+1}^2 are related to the waveforms ψ_{8m+2}^2 and ψ_{8m+6}^2, which are centered around the samples $8m + 2$ and $8m + 6$, respectively. These waveforms occupy, approximately, the same area as the waveform $\psi^3(k)_{8m+4}$. In that sense, we claim that the finer scale coefficients h_{2m}^2 and h_{2m+1}^2 are the "offsprings" of the coarsest scale coefficient h_m^3. In turn, their "offsprings" are the fine scale coefficients h_{4m}^1, h_{4m+1}^1, h_{4m+2}^1 and h_{2m+3}^1. Thus, the coarsest scale coefficient h_m^3 is the root of the tree

$$\begin{array}{ccccc} & & h_{4m}^1 \\ & h_{2m}^2 \nearrow\searrow & \\ h_m^3 \nearrow\searrow & & h_{4m+1}^2 \\ & h_{2m+1}^2 \nearrow\searrow & \\ & & h_{4m+2}^1 \\ & & h_{4m+3}^2 \end{array} \quad . \tag{2.1}$$

The wavelet transform of a two-dimensional array $\mathbf{T} = \{t_{n,m}\}$ of size $N \times M$ is implemented as a tensor product. First, the pair of filters L and H is applied to the columns of $\mathbf{T}$ and the results are downsampled. The coefficients arrays $\mathbf{W}_L^1$ and $\mathbf{W}_H^1$ of size $N/2 \times M$ are produced. Then, the filters L and H are applied to the rows of $\mathbf{W}_L^1$ and $\mathbf{W}_H^1$. This filtering is followed by downsampling which results in four sub-arrays coefficients $\mathbf{W}_{LL}^1$, $\mathbf{W}_{LH}^1$, $\mathbf{W}_{HL}^1$ and $\mathbf{W}_{HH}^1$ of size $N/2 \times M/2$. The 2D Nyquist frequency domain is split accordingly. Then, the above procedure is applied to the coefficient array $\mathbf{W}_{LL}^1$ to decompose it into the subarrays $\mathbf{W}_{LL}^2$, $\mathbf{W}_{LH}^2$, $\mathbf{W}_{HL}^2$ and $\mathbf{W}_{HH}^2$ of size $N/4 \times M/4$. Then, this procedure is iterated using $\mathbf{W}_{LL}^2$ instead of $\mathbf{W}_{LL}^1$ and so on. The layout of the transform coefficients, which corresponds to the Nyquist frequency partition, for the three-scale wavelet transform, is displayed in Fig. 2.3.

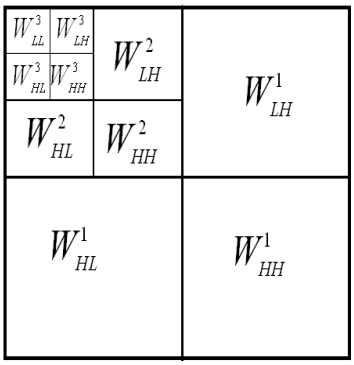

Figure 2.3: Layout of coefficients of the three-scale of a 2D wavelet transform

Similarly to the 1D case, each coefficient $lh^J_{n,m} \in \mathbf{W}^J_{LH}$, $hl^J_{n,m} \in \mathbf{W}^J_{HL}$ and $hh^J_{n,m} \in \mathbf{W}^J_{HH}$ from the coarse scale has four "offsprings" from the finer scale $J-1$.

$$lh^J_{n,m} \swarrow \begin{matrix} lh^{J-1}_{2n,2m}, \\ lh^{J-1}_{2n,2m} \\ lh^{J-1}_{2n+1,2m} \\ lh^{J-1}_{2n+1,2m+1} \end{matrix} \qquad hl^J_{n,m} \swarrow \begin{matrix} hl^{J-1}_{2n,2m}, \\ hl^{J-1}_{2n,2m} \\ hl^{J-1}_{2n+1,2m} \\ hl^{J-1}_{2n+1,2m+1} \end{matrix} \qquad hh^J_{n,m} \swarrow \begin{matrix} hh^{J-1}_{2n,2m}, \\ hh^{J-1}_{2n,2m} \\ hh^{J-1}_{2n+1,2m} \\ hh^{J-1}_{2n+1,2m+1} \end{matrix} \quad . \tag{2.2}$$

Similar relations exist between the coefficients from the scales $J-1$ and $J-2$ and so on. These relations for $J=3$ are illustrated in Fig. 2.4.

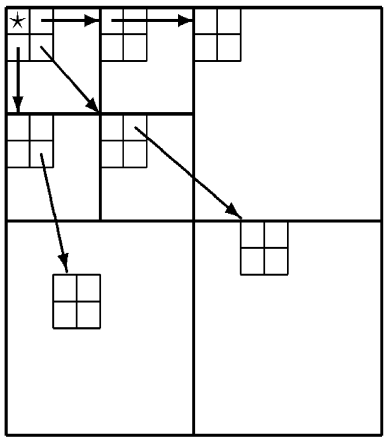

Figure 2.4: Relationship among wavelet coefficients from different scales

Thus, each coefficients $lh^J_{n,m} \in \mathbf{W}^J_{LH}$, $hl^J_{n,m} \in \mathbf{W}^J_{HL}$ or $hh^J_{n,m} \in \mathbf{W}^J_{HH}$ from the coarse scale can be regarded as the root of a quad-tree of coefficients. This relationship between wavelet coefficients in different scales is exploited in the embedded zerotree wavelet (EZW) codec [32]. This codec takes advantage of the self-similarity between wavelet coefficients across the decomposed scales and their decay toward high frequency scales. One of most efficient algorithms, which is based on the zerotree concept, is the SPIHT algorithm [29]. This algorithm combines adaptive quantization of the wavelet coefficients with coding. The produced bitstream is further compressed losslessly by the application of an adaptive arithmetic coding. The SPIHT coding procedure is fast and its decoding procedure is even faster.

3 The hybrid algorithm

We propose to apply the wavelet and the LCT transforms in the horizontal and vertical directions, respectively. The mixed wavelet and LCT transforms coefficients are encoded by the SPIHT algorithm

or by JPEG2000 coder. For this, we organize the array of the mixed transform coefficients in a wavelet-like way. It requires reordering of the LCT coefficients.

The array of DCT coefficients of a signal of length 2^k produces a natural logarithmic split of the frequency band once being separated into $k+1$ blocks:

$$\Big(\; c_0 \mid \;\; c_1 \mid \;\; c_2 \, c_3 \mid \;\; c_4 \, c_5 \, c_6 \, c_7 \mid \;\; c_8 \, c_9 \, c_{10} \, c_{11} \, c_{12} \, c_{13} \, c_{14} \, c_{15} \mid \;\; ... \; \Big)^T. \tag{3.1}$$

The partition in Eq. (3.1) appears automatically when the coefficients indices are presented in a binary mode:

$$\Big(\; c_0 \mid \;\; c_1 \mid \;\; c_{10} \, c_{11} \mid \;\; c_{100} \, c_{101} \, c_{110} \, c_{111} \mid \;\; c_{1000} \, c_{1001} \, c_{1010} \, c_{1011} \, c_{1100} \, c_{1101} \, c_{1110} \, c_{1111} \mid \;\; ... \; \Big)^T. \tag{3.2}$$

Thus, the array is partitioned according to the number of bits in the coefficients indices. We call this a bit-wise partition.

Assume we are given $N \times M$ data array $\mathbf{T}$, where $N = 2^k Q$, $M = 2^J R$. We define the partition P of the interval $I \stackrel{\text{def}}{=} [0, 1, ..., N-1]$ by splitting it into Q subintervals $I = \bigcup_{i=1}^{Q} I^i$ of length 2^k each. We apply the P-based LCT transform to each column of $\mathbf{T}$. Thus, the array $\mathbf{T}$ is transformed into the array $\mathbf{C}$ of LCT coefficients.

For each column, we get the array $\mathbf{c}$ of N LCT coefficients, which consists of Q blocks $\mathbf{c} = \bigcup_{i=1}^{Q} \mathbf{c}^i$, where $\mathbf{c}^i \stackrel{\text{def}}{=} \left\{ c_n^i \right\}_{n=0}^{2^k-1}$. Each of them can be bit-wise partitioned as in Eq. (3.2): $\mathbf{c}^i = \bigcup_{\beta=0}^{k} \mathbf{b}_\beta^i$, where $\mathbf{b}_0^i = c_0^i$, $\mathbf{b}_1^i = c_1^i$ and $\mathbf{b}_\beta^i$ is the set of coefficients c_n^i, whose indices can be represented by β bits.

In order to obtain a wavelet-like structure of the array $\mathbf{c}$ of N LCT coefficients, we rearrange it in a bit-wise mode

$$\mathbf{c} \longrightarrow \mathbf{b} \stackrel{\text{def}}{=} \bigcup_{\beta=0}^{k} \mathbf{b}_\beta, \quad \mathbf{b}_\beta \stackrel{\text{def}}{=} \bigcup_{i=1}^{Q} \mathbf{b}_\beta^i. \tag{3.3}$$

Thus, we get $\mathbf{b}_0 \stackrel{\text{def}}{=} \left(c_0^1, c_0^2, ..., c_0^Q \right)^T$, $\mathbf{b}_1 \stackrel{\text{def}}{=} \left(c_1^1, c_1^2, ..., c_1^Q \right)^T$, $\mathbf{b}_2 \stackrel{\text{def}}{=} \left(c_2^1, c_3^1; c_2^2, c_3^2; ...; c_2^Q, c_3^Q \right)^T$, and so on. This rearrangement is illustrated in Fig. 3.1. The structure of the array $\mathbf{b}$ is similar to the structure of the coefficients array $\mathbf{w}$ of the wavelet transform, where the signal was decomposed into the $k-1$ scale. The similarity relations are

$$\mathbf{b}_0 \sim \mathbf{w}_L^{k-1}, \; \mathbf{b}_1 \sim \mathbf{w}_H^{k-1}, \; \mathbf{b}_2 \sim \mathbf{w}_H^{k-2}, \; ..., \; \mathbf{b}_{k-1} \sim \mathbf{w}_H^1. \tag{3.4}$$

The ancestor-descendant relationship in the array $\mathbf{b}$ are similar to that in $\mathbf{w}$.

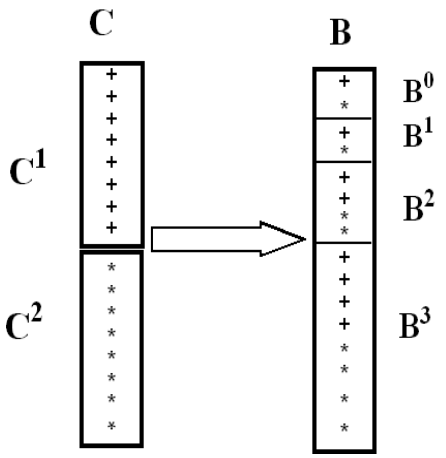

Figure 3.1: Reordering scheme of the LCT coefficients

We perform this reordering of the LCT coefficients for all the columns of the array **C**. Thus, we get

$$\mathbf{C} \longrightarrow \mathbf{B} = \bigcup_{\beta=0}^{k} \mathbf{B}_\beta \quad \mathbf{B}_\beta \stackrel{\text{def}}{=} \bigcup_{i=1}^{Q} \mathbf{B}_\beta^i. \tag{3.5}$$

Then, each row in the array **B** is decomposed down to scale J by the application of the wavelet transform. This produces the hybrid LCT–wavelet coefficients array denoted as **CW**. The structure of **CW** is similar to the the structure of a 2D wavelet coefficients array, where the transform on the columns was decomposed to scale $k-1$, while the transform on the rows was decomposed till scale J. Three scales decomposition structure of the **CW** array is illustrated in Fig. 3.2.

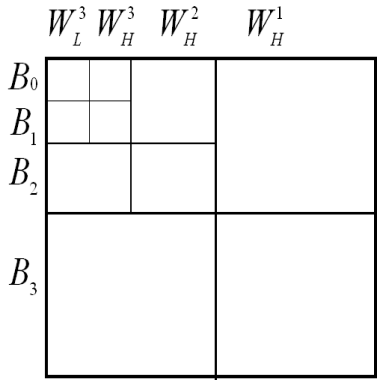

Figure 3.2: Layout of the LCT–wavelet coefficients in three scales

The array **CW** is the input to SPIHT or JPEG2000 coders that implement quantization and entropy coding.

If the data arrays have oscillating structures in both vertical and horizontal directions then ap-

plication of the LCT in both directions followed by reordering of the transform coefficients in both directions is recommended. We call this transform the reordered LCT (RLCTA).

4 Experimental results

The hybrid and RLCTA algorithms were applied to compress different data types: seismic, hyperspectral, fingerprints and multimedia images.

We compared between the performance of these algorithms where the quantization and entropy coding was borrowed from the SPIHT codec and the performance of JPEG2000 compression standard. For the latter, we used the MATLAB script `imwrite`. We also compared between the compression results from HCA and RLCTA algorithms and the results from the application of the 2D wavelet transform using the same SPIHT codec for all the transforms. An additional experimental goal was to compare between the performance of the Butterworth wavelet transforms [2, 3] and that of the the 9/7 wavelet transform [18].

The choice of a bell in the LCT construction (see section 2.1) has some effect on the performance of the LCT and, consequently, on the hybrid algorithm. A library of bells was introduced in [31]. A comparative study of their effects on the performance of an LCT–based image compression algorithm was given in [28]. However, such a comparison is beyond the scope of this paper and we did not check the effects of different bells on the performance of the algorithm. Our goal was to demonstrate the capabilities of the new method. For this purpose, it was sufficient to implement LCT with the simple "sine" bell $b(x) = \sin \frac{\pi}{2}(x + 1/2)$.

All the used transforms have the following notation:

Jpeg2000 standard: JP2k.

2D wavelet transforms: W9/7 – the 9/7 transform; **WButt/M** – the Butterworth wavelet transform with M vanishing moments.

Hybrid transforms: The 1D wavelet transforms in the horizontal direction and the LCT in the vertical direction, which used to partition the vertical section into Q horizontal rectangles. The transform coefficients were reordered to fit SPIHT encoding as was explained in section 3. **H9/7/Q** – the 9/7 wavelet transform in the horizontal direction; **HButt/M/Q** – the Butterworth wavelet transform with M vanishing moments.

2D RLCTA: RLCTA/Q/P –2D LCT transform was applied, which used partition of the image into $Q \times P$ rectangles. The transform coefficients were reordered to fit SPIHT encoding.

All the images that were used in the experiments are 8-bit grayscale images. Assume that the image $\mathbf{X}$ was subjected to lossy compression by some method $\mathbf{M}$ and, then, reconstructed to the image $\tilde{\mathbf{X}}$, whose array of samples is $\tilde{\mathbf{x}} \stackrel{\text{def}}{=} \{\tilde{x}_k\}_{k=1}^{N}$. We compare the quality of different compression methods by the peak signal to noise ratio (PSNR) in decibels

$$PSNR \stackrel{\text{def}}{=} 10 \log_{10} \left(\frac{N\, 255^2}{\sum_{k=1}^{N} (x_k - \tilde{x}_k)^2} \right) \; dB, \tag{4.1}$$

in addition to visual inspection. The SPIHT codec, which we utilized, was enhanced with an adaptive arithmetic coder. The bolded numbers in the following tables indicate the best achieved results.

4.1 Seismic sections compression

Compression of seismic data, which is used for the reconstruction of subsurface layers structure, should retain all the seismic events in the traces.

4.1.1 Stacked CMP data section

We used a stacked common mid point (CMP) data section in the experiments. We display the original section of size 512×512 and a fragment of size 200×200 from it in Fig. 4.1.

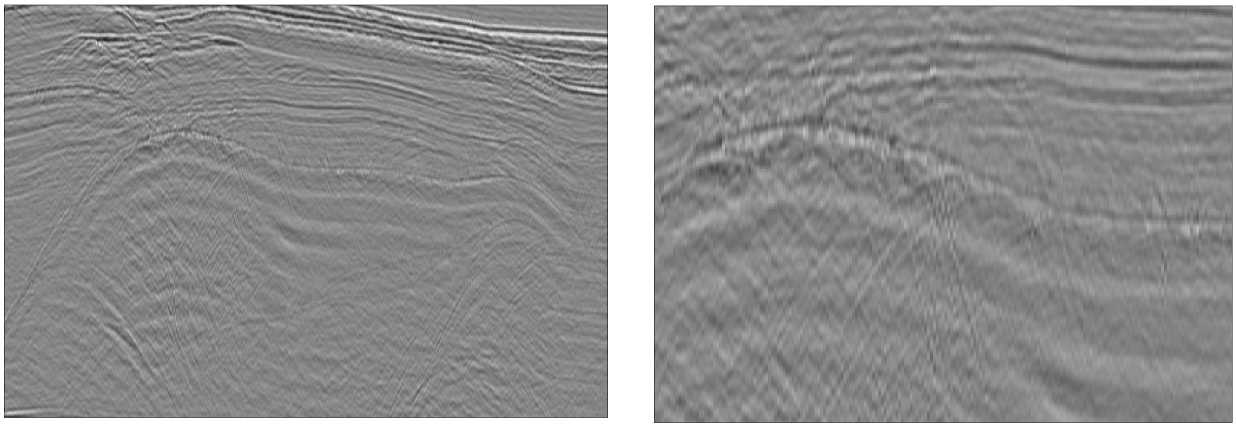

Figure 4.1: The original stacked CMP section. Left: the whole section. Right: a fragment from the left image. The X-axis corresponds to the horizontal direction along the earth surface while the Y-axis corresponds to the vertical (depth) direction.

Five types of transforms were used: **W9/7**, **WButt/10**, **H9/7/8**, **HButt/10/8** and **RLCTA/16/16**, where the transform coefficients were coded by SPIHT. The results were compared with the results from the application of **JP2k**. The achieved PSNR values are presented in Table 4.1.

bit/pixel	JP2k	W9/7	WButt/10	H9/7/8	HButt/10/8	RLCTA/8/8
1/8	31.03	30.86	31.01	31.19	31.39	**31.46**
1/4	33.12	32.79	32.99	33.39	33.79	**33.83**
1/2	35.91	35.39	35.84	36.68	**37**	**37**
1	40.61	39.83	40.42	41.8	42.26	**42.31**
2	48.66	46.91	48.82	51.26	51.6	**51.63**

Table 4.1: The PSNR values after decompression of the stacked CMP seismic section

We observe that the PSNR values after the application of the hybrid transforms and the RLCTA are significantly higher than those after application of the 2D wavelet transforms whether **JP2k** or SPIHT are used. It means that the hybrid transforms and the RLCTA fit better seismic images in comparison to the 2D wavelet transforms. The best PSNR values appear after application of **RLCTA/8/8**, which only slightly exceed the values produced by **HButt/10/8**. The reason for that is in the presence of oscillations in the vertical and so also in the horizontal directions. Note that the 2D wavelet and hybrid transforms, which used the Butterworth wavelets, produced higher PSNR compared to the 9/7 transform. The hybrid transforms and the RLCTA retain seismic events much better than the 2D wavelet transforms. It is illustrated in the following two figures.

Figure 4.2 displays fragments from the reconstruction of CMP section of the image 4.1 (right) after the application of **JP2k** compression algorithm where the compression ratio is 1/4 bit per pixel (left image) and from the application of the **RLCTA/8/8**–SPIHT compression (right image). The **RLCTA/8/8**–SPIHT compression retains the main structure of the data unlike the outcome from the application of the **JP2k** algorithm.

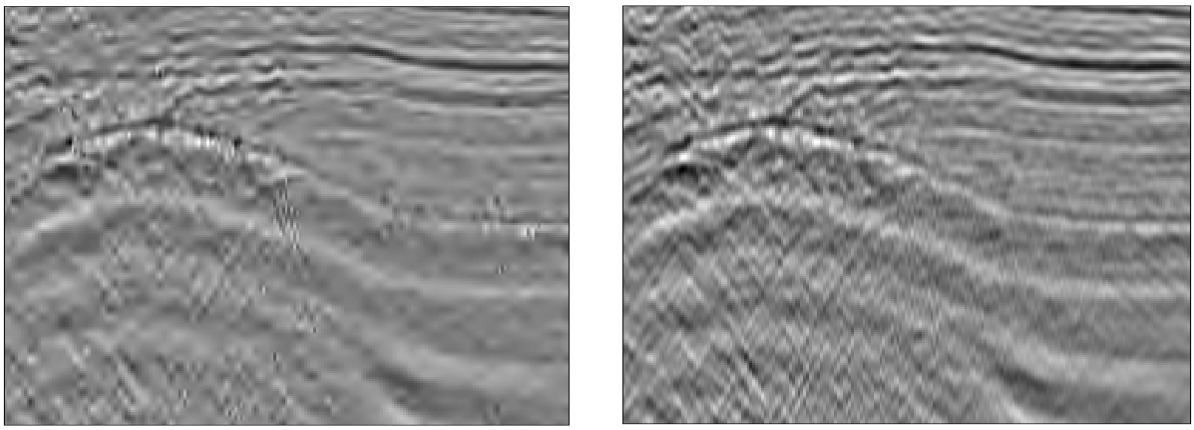

Figure 4.2: A fragment of the reconstructed CMP section in Fig. 4.1 (right). Left: Output from the application of the **JP2k** algorithm. Right: The output from the application of the **RLCTA/8/8**–SPIHT algorithm. The transforms coefficients were encoded by 1/4 bit per pixel. The X-axis corresponds to the horizontal direction along the earth surface while the Y-axis corresponds to the vertical (depth) direction.

Figure 4.3 displays a line from the vertical trace #110 from the original section versus a line from the reconstructed section after the 1/2 bit per pixel compression by the applications of **JP2k** and **HButt/10/8**–SPIHT algorithms.

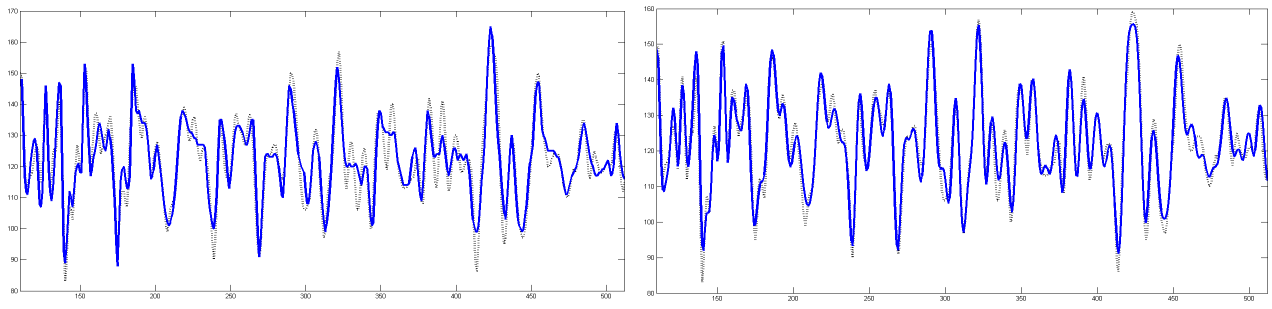

Figure 4.3: The differences between an original line from the vertical trace #110 in the CMP section (Fig. 4.1 (right)) and the reconstructed line after the applications of different compression algorithms. Dotted line: Original. Solid line: Restored. Left: Restored after the application of **JP2k** algorithm. Right: Restored after the application of the **HButt/10/8**–SPIHT algorithm. Both were compressed to 1/2 bit per pixel

We observe that the reconstructed trace in the right image in Fig. 4.3 is very close to the original trace. This is not the case for the trace in the left image.

4.1.2 Marine shot gather data section

For another seismic experiments, we used a marine shot gather (MSG) data section of size 512×512. As before, we compared between the performance of the 2D wavelet transforms and the hybrid transforms that use different wavelets. Each data pixel is quantized to 8 bits. We display the original section of size 512×512 and fragment of size 200×200 from it in Fig. 4.4.

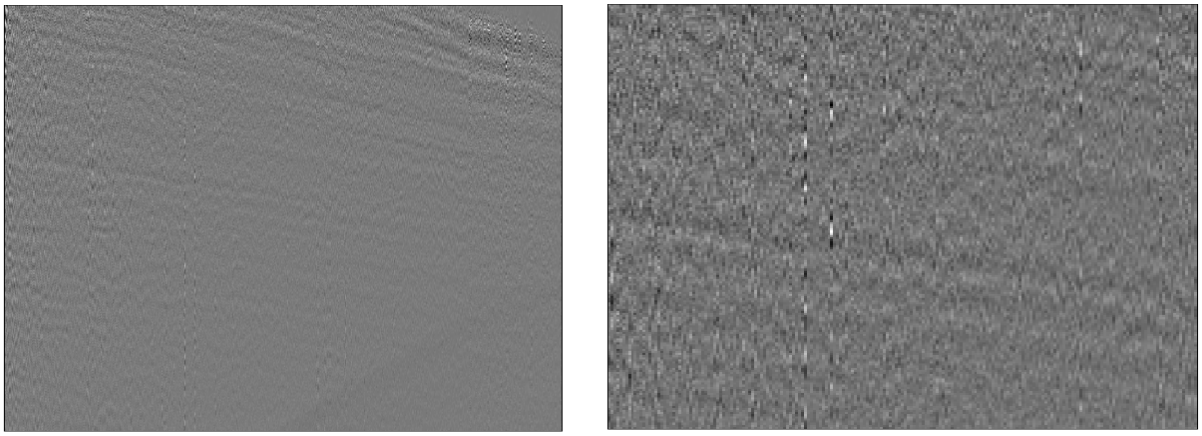

Figure 4.4: The original marine shot gather (MSG) section. Left: The whole section. Right: A fragment from the section on the left. The X-axis corresponds to the horizontal direction along the earth surface while the Y-axis corresponds to the vertical (depth) direction.

Five types of transforms were applied: **W9/7**, **WButt/10**, **H9/7/8**, **HButt/10/8** and **RLCTA/8/8**, where the transform coefficients were coded by SPIHT. The results were compared with the results from the application of **JP2k**. The achieved PSNR values are presented in Table 4.2.

bit/pixel	JP2k	W9/7	WButt/10	H9/7/8	HButt/10/8	RLCTA/8/8
1/4	32.82	32.76	32.69	33.07	**33.22**	33.11
1/2	34.65	34.42	34.31	35.11	**35.23**	35.13
1	38	37.69	37.64	39.10	**39.17**	39.16
2	44	43.28	43.25	45.55	**45.67**	45.62

Table 4.2: The PSNR values after decompression of the MSG seismic section (Fig. 4.4 (right))

The compression results from the applications of **W9/7**, **WButt/10** and **JP2k** wavelet transforms are close to each other where **JP2k** has a small advantage over the other two. However, the hybrid transforms and **JP2k** produce much better results. The **HButt/10/8** transform outperforms **H9/7/10** and **JP2k**. Figure 4.5 displays the fragments of the reconstructed MSG section after the application of **JP2k** (left image) and Hybrid **HButt/10/10** with SPIHT coding (right image). The

compression ratio for both is 1 bit per pixel. **HButt/10/8** transform retains the structure of the data much better compared to **JP2k** .

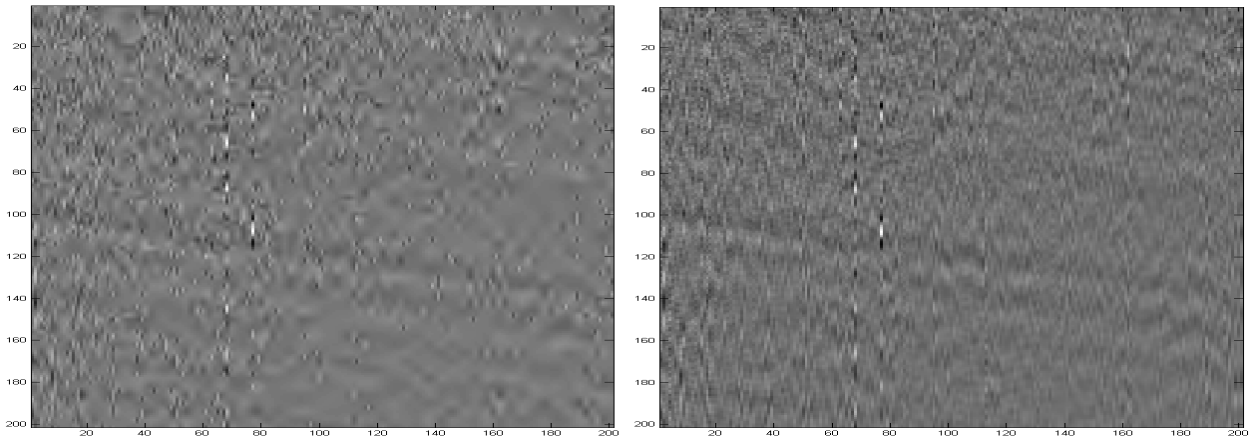

Figure 4.5: A fragment from the reconstructed MSG section (Fig. 4.4 (right)). Left: Reconstruction from the application of **JP2k**, PSNR=34.63 dB. Right: The reconstruction from the application of the Hybrid **HButt/10/8**+SPIHT, PSNR=35.23 dB. The compression ratio is 1 bit per pixel. The X-axis corresponds to the horizontal direction along the earth surface while the Y-axis corresponds to the vertical (depth) direction

Figure 4.6 displays a line from the vertical trace #270 from the original section taken from Fig. 4.4 (right) versus the lines from the section reconstructed after the application of **JP2k** and **HButt/10/8**+SPIHT algorithms.

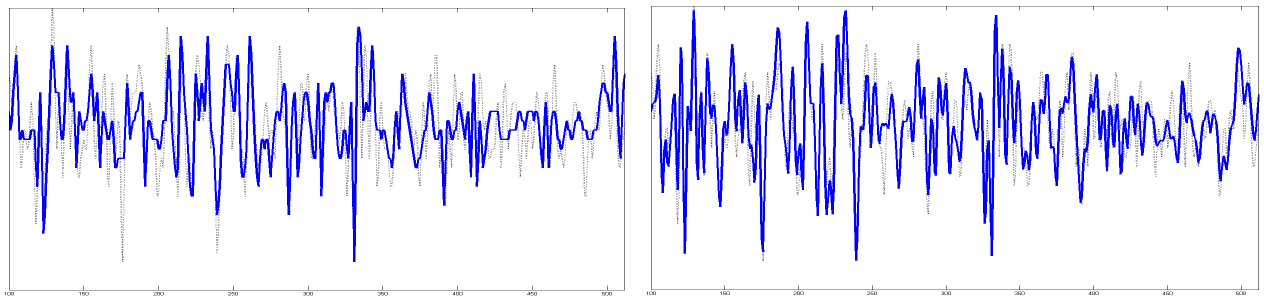

Figure 4.6: Line from the vertical trace #270 in the MSG section. Dotted lines: Original. Solid line: Restored. Left: After the application the **JP2k** algorithm. Right: The same after the application of the **HButt/10/8**+SPIHT algorithm. Both are compressed to 1 bit bit per pixel

We see that the trace, which was reconstructed from **HButt/10/8**+SPIHT encoding/decoding algorithm, is much closer to the original trace in comparison to the trace reconstructed after the

application of the **JP2k** algorithm that missed many details.

Conclusion: For seismic data, the Hybrid compression algorithm significantly outperforms compression algorithms that are based on the 2D wavelet transforms including JPEG2000 coding scheme.

4.2 Compression of hyper-spectral images

The hybrid compression was applied to hyper-spectral data cubes. These cubes were captured from a plane that took simultaneously pictures in many (≈ 200) wavebands from the ground surface. This type of camera can also be placed in a satellite. We used the SPECIM camera ([40]). In this camera, the spectrum of intensities for all the wavebands is assigned to each spatial pixel, which is called multipixel, at once. The proposed compression methodology fits this capturing type and no buffering is needed in order to compress the hyper-spectral data. An hyper-spectral image is treated as a 3D cube where X,Y are the spatial axes and Z is the wavelength axis. The camera captures parallel lines of multipixels that are placed in the X/Z planes. The structure of the Z direction is very different from that of the X direction. This justifies the application of the hybrid transform. This processing enables real-time transmission of hyper-spectral data without the need to buffer the data before the application of the hybrid compression algorithm. The results are compared with the results from the application of JPEG2000 compression standard.

When hyper-spectral data is compressed, it is important to preserve the spectral characteristic features of the multipixels. We applied the compression algorithm to different hyper-spectral data cubes scenarios. Here, we present the results from the application of the compression algorithm to an urban scenery that has many details and thus contains many oscillations.

Figure 4.7 displays an urban ground scene and one of the X/Z multipixel planes. This image was captured by the Specim camera placed on a airplane flying 10,000 feet above sea level. The resolution is 1.5 meter/pixel, 3000 $\times$ 300 pixels per waveband with 180 wavebands.

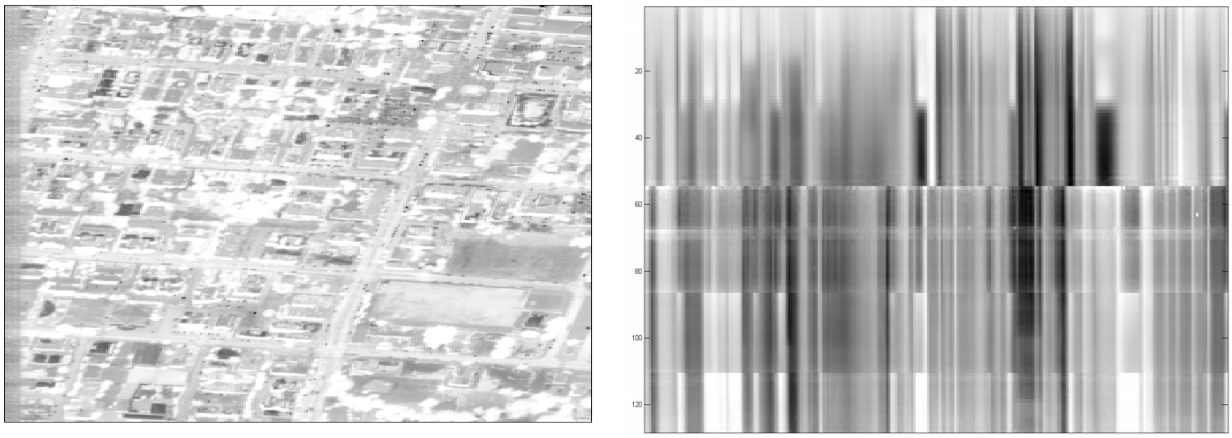

Figure 4.7: An hyper-spectral image. Left: One horizontal plane from the data cube. Right: A multipixel plane. The X-axis corresponds to the direction along a horizontal line of pixels in the left image while the Y-axis indicates the wavebands.

Five types of transforms were applied: **W9/7**, **WButt/4**, **H9/7/8**, **HButt/4/8** and **RLCTA/8/8**, where the transform coefficients were coded by the SPIHT mechanism. The results were compared with the results from the application of **JP2k**. Table 4.3 illustrate results of the comparison on the X/Z plane #200. The performance is given in PSNR values.

bit/pixel	JP2k	W9/7	WButt/4	H9/7/8	HButt/4/8	RLCTA/8/8
1/4	36.79	35.39	35.32	37.73	**38.18**	38.03
1/2	40.9	39	38.83	42.03	**42.25**	41.95
1	46	43.62	43.58	**46.64**	46.51	46.21
2	51.65	49.97	50.01	52.1	**52.24**	51.69

Table 4.3: PSNR values of hyper-spectral compression of X/Z plane #200 (plane 200 from the 3000 planes of the whole hyper-spectral cube) from the urban scene.

We observe that the hybrid transforms with SPIHT produce PSNR values, which are much higher compared to results from the applications of the **JP2k** algorithm, which, in turn, outperforms the wavelet transforms. For all the compression rates except for 1 bpp, the hybrid transform **HButt/4/8** outperforms all. In addition, the hybrid transforms retain the spectral features much better than the wavelet transforms and **JP2k**. Figure 4.8 displays a multipixel from the plane #200 from 3000 planes in Fig. 4.7 (left), which was reconstructed after the application of **JP2k** (left image) and **HButt/4/8** (right inmage) transforms followed by SPIHT encoding/decoding where the compression ratio is 0.25 bit per pixel.

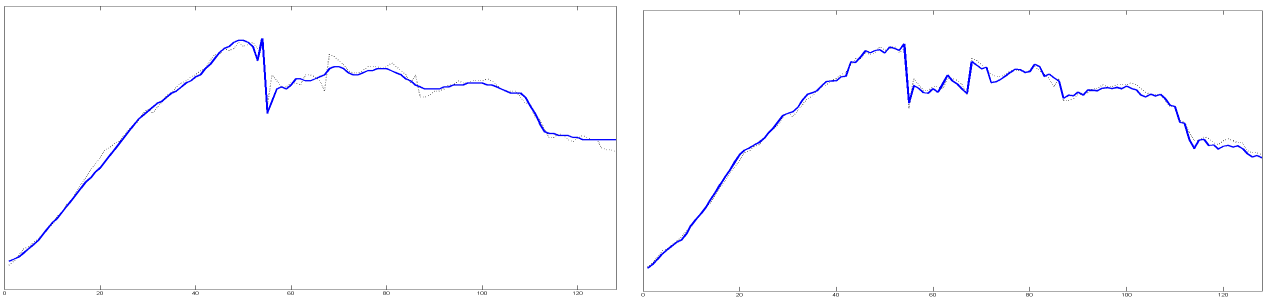

Figure 4.8: One multipixel from the plane X/Z #200. Dotted lines: Original. Solid line: Restored. Left: After the application of – **JP2k**. Right: After the application of **HButt/10/8**+SPIHT encoding/decoding. The compression ratio for both is 0.25 bits per pixel

Almost all the "events" in the original data are present in the reconstructed signal after the application of the hybrid transform even under high compression. This is not the case following the application of the **JP2k** algorithm.

4.3 Fingerprints

The 2D wavelet, hybrid and LCT transforms followed by the SPIHT coding were applied to compres a fingerprint image of size 512 × 512, shown in Fig. 4.9. It was downloaded from C. Brislawn's web page http://www.c3.lanl.gov/∼brislawn/index.html). It has an oscillating structure in each direction. Each pixel has 8 bits.

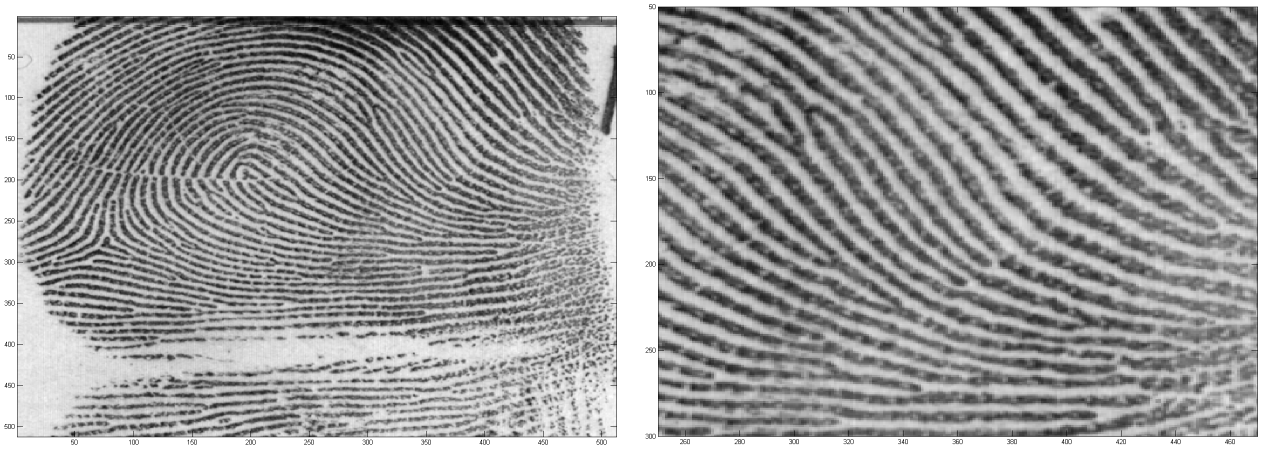

Figure 4.9: The original fingerprint image. Left: The whole image. Right: A fragment from the left image

The performance of the SPIHT-coded algorithms was compared with the performance of JPEG2000

compression standard. The LCT transforms, used the partition of the image into 16 horizontal and vertical rectangles. The achieved PSNR values are given in Table 4.4.

bit/pixel	JP2k	W9/7	WButt/10	H9/7/16	HButt/10/16	RLCTA/16/16
1/8	23.54	23.05	24.03	23.87	24.38	**24.55**
1/4	26.23	25.51	26.42	26.53	26.77	**26.99**
1/2	29.84	29.33	29.91	30.06	29.99	**30.23**
1	33.94	33.3	33.83	33.97	34.27	**34.83**

Table 4.4: PSNR values for fingerprint compression.

We can see from the table that **RLCTA/16/16**+SPIHT outperforms all the other algorithms at all the bitrates. This happens because of better handling of the oscillating events compared to the algorithms that only wavelet transforms. The hybrid transform **HButt/10/16** produces PSNR values close to **RLCTA/16/16**. **WButt/10**+SPIHT, which uses the 2D Butterworth wavelet transform, achieves a good performance, which demonstrates advantage over **W9/7**.

Figure 4.10 displays fragments from the reconstructed fingerprint image after the applications of **JP2k** (left) and **RLCTA/16/16**+SPIHT (right). The compression ratio is 1/8 bit per pixel. The right fragment retains much better the structure of the data than the fragment from **JP2k** compression (left).

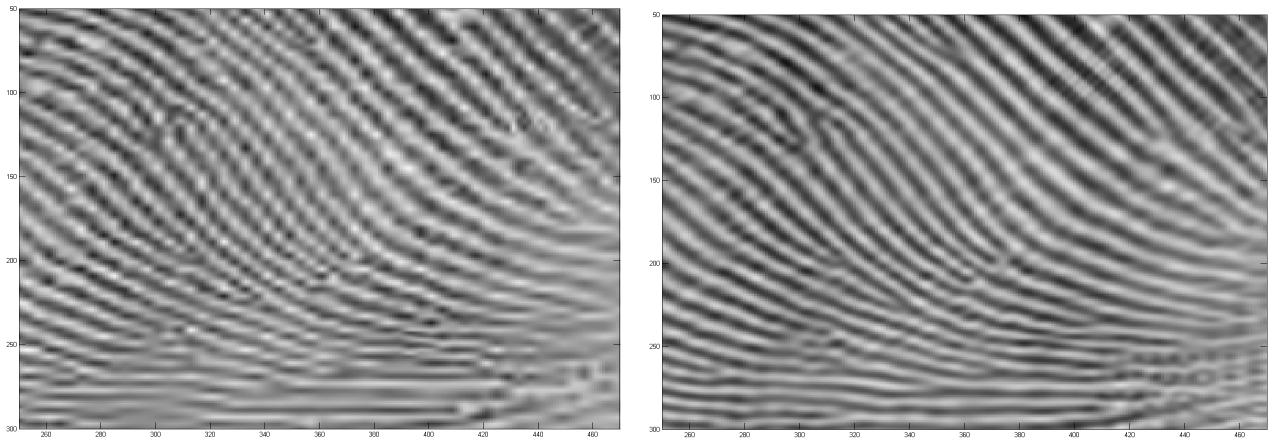

Figure 4.10: A fragment from the reconstructed fingerprint image. Left: After the application of **JP2k**, PSNR=23.54. Right: After the application of **RLCTA/16/16**+SPIHT, PSNR=24.55. The compression ration is 1/8 bit per pixel

Figure 4.11 displays the column line #300 from the original section versus this column line from the reconstructed sections after the applications of **JP2k** (left) and **RLCTA/16/16**+SPIHT (right).

The compression ratio is 1/8 bit per pixel.

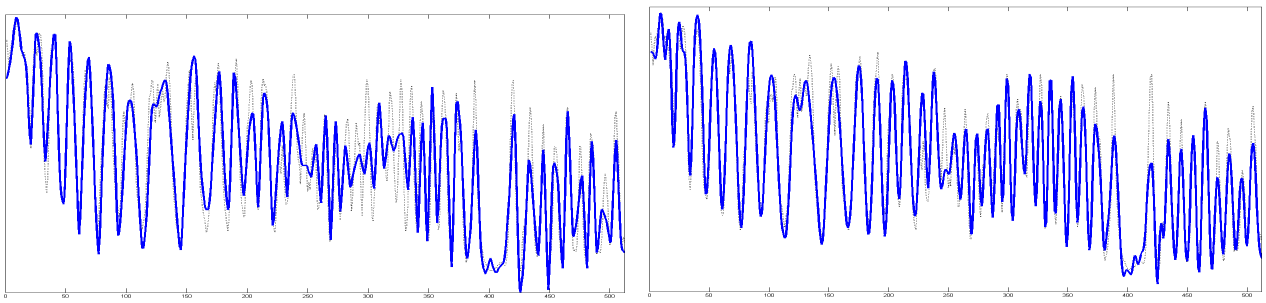

Figure 4.11: Column line #300 from the fingerprint image. Dotted line: Original. Solid line: restored. Left: The two lines after the application of **JP2k**. Right: The two lines after the application of **RLCTA/16/16**+SPIHT (right). The compression ratio is 1/8 bit per pixel

We observe that, unlike **JP2k**, **RLCTA/16/16**+SPIHT restores the curve very close to its original form even when the compression ratio is very low.

4.4 Compression of multimedia images

The hybrid compression algorithms also proved to be efficient for multimedia images. Its superior performance is more evident for images that have oscillating texture. The algorithm restores the texture even at a very low bitrate. On the other hand, it sometimes produces artifacts on the boundaries between smooth and texture areas.

Barbara image "Barbara" image of size 512×512 and its fragment of size 280×280 are displayed in Fig. 4.12. Each pixel in this image has 8 bits.

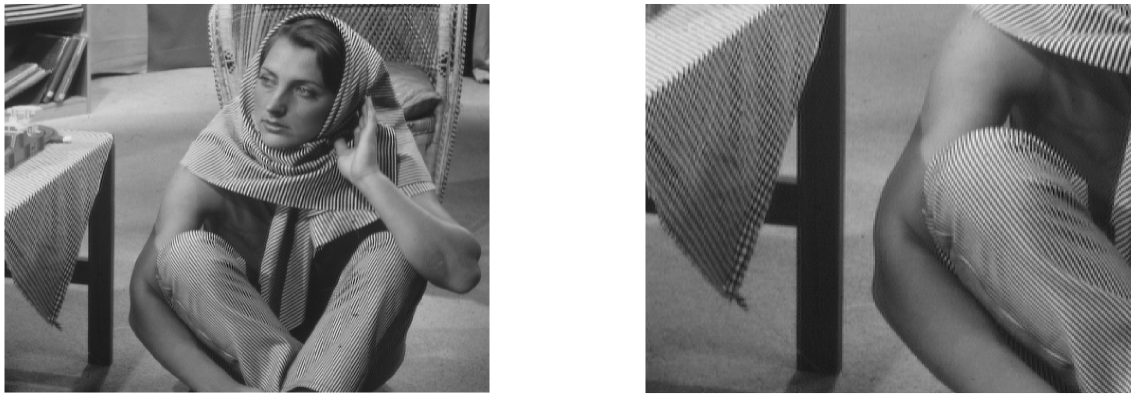

Figure 4.12: The original "Barbara" image . Left: The whole image. Right: A fragment

This image comprises areas with oscillating textures. We compared between the performance of 6 transforms where the hybrid and RLCTA transforms partitions the vertical direction into $Q = 16$ horizontal rectangles each of height 32. The achieved PSNR values are presented in Table 4.5.

bit/pixel	JP2k	W9/7	WButt/10	H9/7/16	HButt/10/16	RLCTA/16/16
1/8	25.78	25.18	25.14	25.71	26.08	**26.78**
1/4	28.68	27.85	28.29	28.63	29.18	**29.54**
1/2	32.71	31.57	32.32	32.72	33.28	**33.28**
1	37.95	36.82	37.9	38	**38.38**	38.15

Table 4.5: The PSNR values from the compression/decompression of "Barbara"

RLCTA/16/16+SPIHT, which produced the best results except 1 bit per pixel, retains the fine texture of the image much better than **JP2k**. This fact is illustrated in Fig. 4.13, which displays fragments of the reconstructed "Barbara" image after the applications of **JP2k** and **RLCTA/16/16**+SPIHT algorithms. The compression ratio is 1/8 bit per pixel.

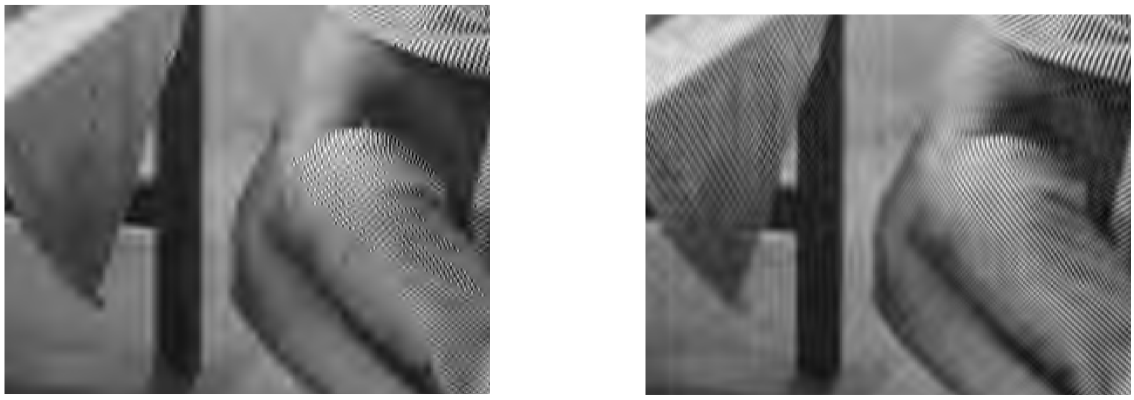

Figure 4.13: A fragment from the reconstructed "Barbara" image after the applications of: Left: **JP2k**. Right: **RLCTA/16/16**+SPIHT. The compression ratio is 1/8 bit per pixel

Figure 4.14 displays the column line #246 from the original section versus this column linr from the reconstructed sections after the application of **JP2k** (left) and **RLCTA/16/16**+SPIHT (right). The compression ratio is 1/4 bit per pixel.

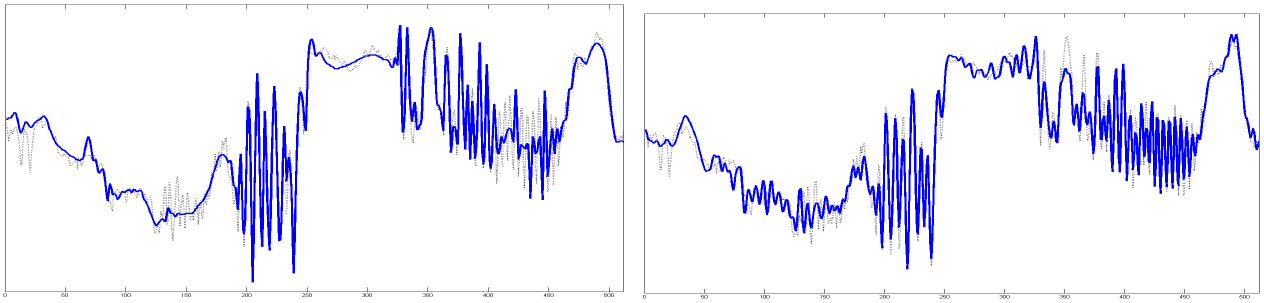

Figure 4.14: Column line #246 from the Barbara image. Dotted line: Original. Solid line: restored. Left: Restored after the application of **JP2k**. Right: Restored after the application of **RLCTA/16/16**+SPIHT (right). The compression ratio is 1/4 bit per pixel

We observe that **RLCTA/16/16**+SPIHT algorithm retains most of the oscillators events that were missed by **JP2k**.

Elaine image "Elaine" image of size 512×512 and its fragment of size 236×256 are displayed in Fig. 4.15. Each pixel has 8 bits.

Figure 4.15: The original "Elaine" image . Left: The whole image. Right: A fragment the right image

We compared between the performance of 6 transforms where the hybrid and RLCTA transforms partitions the vertical direction into $Q = 8$ horizontal rectangles each of height 64. The achieved PSNR values are presented in Table 4.6.

bit/pixel	JP2k	W9/7	WButt/4	H9/7/16	HButt/10/16	RLCTA/16/16
1/8	31.11	31.1	**31.12**	30.41	30.65	30.77
1/4	32.29	32.33	32.33	31.96	32.22	**32.8**
1/2	33.48	33.48	33.36	33.94	34.15	**34.9**
1	36.06	35.75	35.46	36.41	36.79	**37.72**

Table 4.6: The PSNR values from the compression/decompression of "Elaine".

The highest PSNR values in all the bitrates except for 1/8 bit per pixel were produced by **RLCTA/8/8**+SPIHT, while the 2D wavelet transform with the Butterworth wavelet **WButt/4**+SPIHT was advantageous with 1/8 bit per pixel. The advantage of **RLCTA/8/8**+SPIHT over **JP2k** at bitrate 1/2 and 1 bit per pixel was overwhelming. At bitrates 1/8 and 1/4 bit per pixel, reconstruction from **RLCTA/8/8**+SPIHT and hybrid algorithms produced better visual quality of the images compared to **JP2k**, which oversmoothed the images. This fact is illustrated in in Fig. 4.16, which displays fragments of "Elaine" image that were reconstructed after application of **JP2k** and of **RLCTA/8/8**+SPIHT. The compression ratio is 1/8 bit per pixel.

Figure 4.16: A fragment of the reconstructed "Elaine" image after the applications of: Left: **JP2k**. Right: **RLCTA/8/8**+SPIHT. The compression ratio is 1/4 bit per pixel

Oversmoothing by **JP2k** is clearly seen in Fig. 4.17, which displays column line #280 from the original section versus this column line from the reconstructed sections after the applications of **JP2k** (left) and **RLCTA/8/8**+SPIHT (right). The compression ratio is 1/4 bit per pixel.

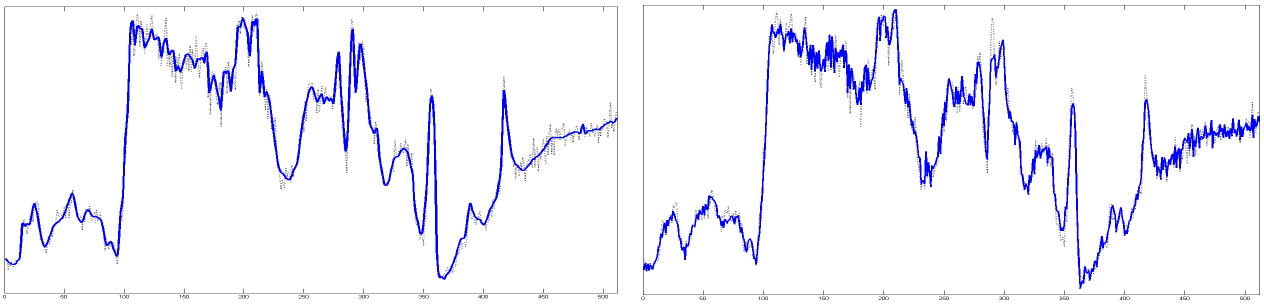

Figure 4.17: Column line #280 in the Elaine image. Dotted line: Original. Solid line: restored. Left: Restored after the application of **JP2k**. Right: Restored after the application of **RLCTA/8/8**+SPIHT (right). The compression ratio is 1/4 bit per pixel

We observe that **RLCTA/8/8**+SPIHT algorithm retains most of the fine texture that was blurred by **JP2k**.

5 Conclusions

The experimental results strongly support our assumption that the hybrid wavelet– LCT – SPIHT and 2D LCT algorithms with reordering of the transform coefficients is a new powerful tool to compress seismic data, hyper-spectral images, fingerprints and other data that comprise of oscillatory structures and(or) a fine texture. Although all the components of the algorithm are well known, their combined operation via reordering of the transform coefficients significantly outperforms compression schemes that are based on multidimensional wavelet transforms with SPIHT coding and JPEG2000. Additional flexibility of the method stems from the use of the library of Butterworth wavelet transforms of different orders. In almost all the experiments, the hybrid transforms with Butterworth wavelets outperform the transforms with the popular 7/9 wavelets. The 2D LCT algorithm with reordering of the transform coefficients demonstrates an excellent performance when oscillating events are present in different directions as in fingerprints or when the image comprise a fine texture as for example in the "Elaine" image.

The algorithm retains fine oscillating events even at a low bitrate that is important for seismic, hyper-spectral and fingerprint images processing and less critical for multimedia type images. The extension of the algorithm to compress 3D seismic or hyper-spectral cubes is straightforward. In the horizontal planes, the 2D wavelet transform is applied, while the LCT is applied to vertical directions followed by reordering the coefficients. The joint array of coefficients is the input to SPIHT encoding. The algorithm is completely automatic and the encoding/decoding operations can be implemented in a fast way.

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] A. Z. Averbuch, A. B. Pevnyi and V. A. Zheludev *Butterworth wavelet transforms derived from discrete interpolatory splines: Recursive implementation*, Signal Processing, **81**, (2001), 2363-2382.

[3] A. Averbuch, V. Zheludev, *Wavelet and frame transforms originated from continuous and discrete splines*, in "Advances in Signal Transforms: Theory and Applications",J. Astola and L. Yaroslavsky eds. , pp. 1-56, Hindawi Publishing Corp., NY, 2007.

[4] A. Averbuch, G. Aharoni, R. Coifman,M. Israeli, *Local cosine transform - A method for the reduction of blocking effects in JPEG*, Journal of Mathematical Imaging and Vision, Special Issue on Wavelets, Vol. 3, pp. 7-38, 1993.

[5] 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.

[6] C. M. Brislawn, *The FBI Fingerprint Image Compression Specification*, in "Wavelet Image and Video Compression", ed. P. Topivala, Springer, 2002, 271-288.

[7] Tseng, Y.H., H.K. Shih, and P.H. Hsu, *Hyperspectral Image Compression Using Three-dimensional Wavelet Transformation*, Proceedings of the 21th Asia Conference on Remote Sensing, 2000, pp. 809-814.

[8] V. A. Zheludev, D. D. Kosloff, E. Y. Ragoza, Compression of segmented 3D seismic data, International Journal of Wavelets, Multiresolution and Information Processing, vol. 2, no. 3, (2004), 269-281.

[9] Y.-A. Jeong and C.-K. Cheong, A DCT-based embedded image coder using wavelet structure of DCT for very low bit rate video codec, IEEE Transactions on Consumer Electronics, vol. 44, no. 3, (1998),500–508.

[10] V. P. Shah, J. E. Fowler, and N. H. Younan, Tarp Filtering of Block-Transform Coefficients for Embedded Image Coding, in Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, Toulouse, France, May 2006, vol. 2, pp. 21-24.

[11] B. Penna, T. Tillo, E. Magli, and G. Olmo, Progressive 3-D coding of hyperspectral images based on JPEG 2000, IEEE Geoscience and Remote Sensing Letters, vol.3, no. 1, (2006), 125–129.

[12] Q. Du and J. E. Fowler, Hyperspectral Image Compression Using JPEG2000 and Principal Component Analysis,IEEE Geoscience and Remote Sensing Letters, vol. 4, pp. 201-205, April 2007..

[13] H. S. Malvar, Fast progressive image coding without wavelets, in Proceedings of the IEEE Data Compression Conference, J. A. Storer and M. Cohn, Eds., Snowbird,UT, March 2000, pp. 243.252.

[14] James E. Fowler, Justin T. Rucker, *Three-Dimensional Wavelet-Based Compression of Hyperspectral Imagery*, in "Hyperspectral Data Exploitation". Ed. Chein-I Chang, John Wiley & Sons, (2007), 379-407.

[15] A. Cohen, I. Daubechies and J.-C. Feauveau, Biorthogonal bases of compactly supported wavelets, *Commun. on Pure and Appl. Math.*' 45:485-560, 1992.

[16] X. Tang and W. A. Pearlman, *Three-dimensional wavelet–based compression of hyperspectral images*, in "Hyperspectral data compression", Eds. G. Motta, F. Rizzo, and J. A. Storer, Kluwer Acad. Publ., 2006, 273-308.

[17] G. Motta, F. Rizzo, and J. A. Storer eds "Hyperspectral data compression", Kluwer Acad. Publ., 2006, 273-308.

[18] M. Antonini and M. Barlaud and P. Mathieu and I. Daubechies, Image Coding using Wavelet Transform, IEEE Trans. on Image Processing, vol. 1, no. 2 (1992), 205–220.

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

[20] R. Rao, P. Yip, *Discrete Cosine Transform: Algorithms, Advantages, Applications*, Academic Press, 1990.

[21] 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.

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

[23] A. V. Oppenheim and R. W. Schafer. *Discrete-time signal processing.* Englewood Cliffs, New York, Prentice Hall, 1989.

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

[25] JPEG 2000 Standard, http://www.jpeg.org/jpeg2000/.

[26] 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.

[27] 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.

[28] F. G. Meyer, *Image Compression With Adaptive Local Cosines: A Comparative Study*, IEEE Trans. on Image Proc., vol. 11, no. 6, June 2002, 616–629.

[29] 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., vol. 6, pp. 243-250, 1996.

[30] S. Mallat *A wavelet tour on signal processing,* Acad. Press, 1999.

[31] G. Matviyenko, *Optimized local trigonometric bases*, Applied and Computational Harmonic Analysis, 3, 301-323, 1996.

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

[33] W. Sweldends *The lifting scheme: A custom design construction of biorthogonal wavelets*, Appl. Comput. Harm. Anal. **3(2)**, (1996), 186-200.

[34] 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.

[35] 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.

[36] Z. Xiong, O. Guleryuz and M. T. Orchard, A DCT-based embedded image coder, IEEE Signal Processing Letters., vol. 3, pp. 289-290, Nov. 1996.

[37] 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.

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

[39] 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, Vol. 10, pp. 1374-1387, Dec. 2000.

[40] Specim camera, http://www.specim.fi/.