International Journal of Wavelets, Multiresolution
and Information Processing
Vol. 11, No. 5 (2013) 1350032 (25 pages)

DOI: 10.1142/S021969131350032X





# LCT-WAVELET BASED ALGORITHMS FOR DATA COMPRESSION

AMIR Z. AVERBUCH*, VALERY A. ZHELUDEV†
and MOSHE GUTTMANN‡

*School of Computer Science*
*Tel Aviv University, Tel Aviv 69978, Israel*
**amir@math.tau.ac.il*
†*zhel@post.tau.ac.il*
‡*guttm@cs.tau.ac.il*

DAN D. KOSLOFF

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

Received 16 August 2012
Revised 31 May 2013
Accepted 1 June 2013
Published 29 August 2013

We present an algorithm that compresses two-dimensional data, which are piece-wise smooth in one direction and have oscillatory events in the other direction. Fine texture, seismic, hyper-spectral and fingerprints have this mixed structure. The transform part of the compression process is an algorithm that combines the application of the wavelet transform in one direction with the local cosine transform (LCT) in the other direction. This is why it is called hybrid compression. The quantization and the entropy coding parts in the compression process were taken from SPIHT codec but it can also be taken from any multiresolution based codec such as EZW. 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 of a fine texture, a 2D LCT with coefficients reordering is applied. These algorithms outperform algorithms that are solely based on the the application of 2D wavelet transforms to each direction with either SPIHT or EZW coding including JPEG2000 compression standard. The proposed 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 Butterworth wavelet transforms library that outperforms the 9/7 biorthogonal wavelet transform.

*Keywords*: Wavelet transform; local cosine transform (LCT); compression; hybrid compression; SPIHT; hyper-spectral; seismic; fingerprints.

AMS Subject Classification: 42C40, 68P30, 94A08

## 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 them 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 analytical capabilities that take place after decompression. The compression of these datasets should preserve fine details. Since wavelet transforms have a successful history in achieving high compression ratios for still images, these techniques were ported to handle seismic compression.[14,19,32] 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[2,23] due to the oscillatory patterns that are present in seismic data.

These oscillatory patterns can be properly handled by cosine transforms. The local cosine transform (LCT),[2,23,33] which uses the lapped DCT[a]-IV transform (LCT)[13] with adaptive partition, was applied to compress 2D seismic sections. The LCT catches well oscillatory patterns and sparsifies it as the wavelet transform sparsifies smooth data — see Refs. 6–8. 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,[30] SPIHT[28] and JPEG2000 standard.[18]

It was observed in Ref. 34 that the cosine transforms coefficients can be rearranged in a way where 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).[17,20] The DCT-II–based algorithm coupled with the SPIHT coder for compressing the segmented seismic cube is presented in Ref. 35. An embedded tarp filtering combined with the classification of the reordered DCT-II coefficients (TCE) is applied to the image compression in Ref. 29. Evaluation of the image compression quality is given in Ref. 9.

Seismic data has a different structure in its vertical and horizontal directions. While the horizontal structure is piece-wise smooth, the vertical traces comprise of oscillatory patterns. To some extent, the same can be said on hyper-spectral data where each spatial pixel is represented by a vector (also called multipixel) of the intensities in all the available wavebands (typically $\approx 200$). 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 a different transform to a different direction.

[a]Discrete Cosine Transform.

One realization of this approach is presented in Ref. 16 where 3D hyper-spectral data cubes were compressed by the application of a 2D wavelet transform in the spatial direction and the Karhunen–Loéve transform in the spectral direction. The transforms coefficients were coded by JPEG2000. Another scheme for a 3D hyper-spectral data compression was presented in Ref. 25 where different types of wavelet transforms were applied to the spatial and spectral directions 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).
(2) Storage of these 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 data rectangles result from data acquisition schemes in hyper-spectral imaging and in seismic exploration. Thus, the proposed algorithms can be directly used for the onboard compression. For on-the-ground compression, 3D methods produce better results but our scheme can be easily extended to 3D setting.

We exploit the fact that wavelets provide a sparse representation for 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 a mix of LCT and wavelet coefficients. In order to be processed properly by WOCS, the mixed 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 a spatial meaning by partitioning the data in the vertical direction. The joint coefficients array is organized in a tree-like form by reordering the LCT coefficients. Therefore, this scheme 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 the transform coefficients and then the WOCS is applied. This method is abbreviated by RLCTA.

We compare between the performances of HCA and RLCTA where both use the SPIHT codec, and the performance of JPEG2000. In addition, we also compare with the performance of 2D wavelet-based algorithms that use SPIHT codec. In most of the experiments, which operate on hyper-spectral and seismic data, HCA and RLCTA that uses SPIHT outperforms JPEG2000 and wavelet-based algorithms that use SPIHT. In addition, our experiments demonstrated that the biorthogonal wavelet transforms, which are based on infinite impulse response (IIR) Butterworth filters,[3,4] provide better performance in comparison with the transforms that are based on the popular 9/7 transforms,[1] which are also utilized in the JPEG2000 standard. This is true for the 2D wavelet transforms as well as for the hybrid transforms.

Compression of fingerprints images is another successful application of HCA and RLCTA. The FBI uses the compression standard[11] that is based on the 9/7 2D wavelet transform. This transform, which uses finite impulse response (FIR) filters of lengths 7 and 9, is included also in JPEG2000 compression standard. The HCA and RLCTA produce higher PSNR results and retain better the structure of the fingerprints in comparison to the outputs from the application of 2D wavelet transforms and JPEG2000 to fingerpints.

The HCA performance on multimedia type images, which are relatively smooth such as "Lena" and "Fabrics", was close (sometimes even inferior) to the 2D wavelet-based algorithms performance. On the other hand, 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. HCA and RLCTA retain the texture and the oscillations even in a very low bit rate.

To summarize, the contributions of the paper are threefold:

(1) We exploit the fact that wavelets provide a sparse representation for such a mixed-structure data in the horizontal direction while the LCT handles well oscillatory patterns. This provides a new way to enhance wavelet based compression.
(2) We demonstrate that the biorthogonal wavelet transforms, which are based on infinite impulse response (IIR) Butterworth filters, provide better performance in comparison with the transforms that are based on the popular 9/7 transforms, which are also utilized in the JPEG2000 standard.
(3) The proposed methodology outperforms the existing state-of-the art compression schemes including JPEG2000 applied to fine texture, seismic, hyper-spectral and fingerprints images that are characterized by having sharp oscillations. The proposed method does not have special advantages when it is applied to regular rich multimedia images.

The paper is organized as follows. Section 2 presents some known facts about LCT, wavelet transforms and SPIHT coding. Section 3 describes the hybrid algorithm. Section 4 presents the experimental compression results 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)[26] of the signal $\mathbf{f}$ and its inverse are defined as $\hat{f}^{IV}(k) = \sum_{n=0}^{N-1} f_n \cos[\frac{\pi}{N}(k+\frac{1}{2})(n+\frac{1}{2})]$, $f_n = \frac{2}{N}\sum_{k=0}^{N-1} \hat{f}^{IV}(k) \times \cos[\frac{\pi}{N}(k+\frac{1}{2})(n+\frac{1}{2})]$. The basis signals $\{\cos[\frac{\pi}{N}(k+\frac{1}{2})(n+\frac{1}{2})]\}, n = 0, \ldots, N-1$, 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,[13] which are the windowed lapped DCT-IV

transforms. These bases were successfully exploited in Refs. 2, 5, 23, 22, 33 for image compression in general and seismic data compression in particular.

Assume we have a signal $\mathbf{S} \triangleq \{s_k\}_{k=0}^{N-1}$ 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) or the local Fourier basis of a signal, is to apply overlapped bells to adjacent sub-intervals. Then, the overlapped parts are folded back to 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 Refs. 2, 5, 13, 21. It is important that LCT produces no discrepancies between adjacent intervals. The choice of a bell is discussed in Refs. 5, 22. The bell we chose is given in Sec. 4. There are fast algorithms to implement the LCT.

To achieve a sparse representation of the oscillatory kernel a multiscale adaptive local cosine transform rather than a one-level local cosine transform is used as described in Ref. 6. Multiscale adaptive use of the local cosine transform enables us to achieve impressive compact coding description of oscillatory data. It can be used in sparsification of operator kernels.



### 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 other well known facts that are needed later.

The multiscale wavelet transform of a signal is implemented via iterated multirate filtering by a pair of $L$ (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. 1.

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 coefficients 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 $\psi^j(k), j = 1, \ldots, 4$, $\Phi^4(k)$ and their frequency responses are displayed in Fig. 2.

#### 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 $\psi_{8m+4}^3$, which is centered around the sample $8m + 4$. The high-frequency coefficients $h_{2m}^2$ and $h_{2m+1}^2$ from the second scale are related to the waveforms $\psi_{8m+2}^2$ and $\psi_{8m+6}^2$, 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

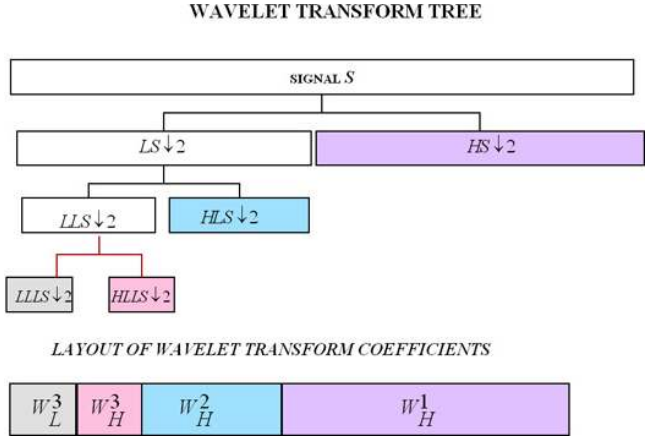


Fig. 1. Three-scale wavelet transform and the layout of the transform coefficients (color online).

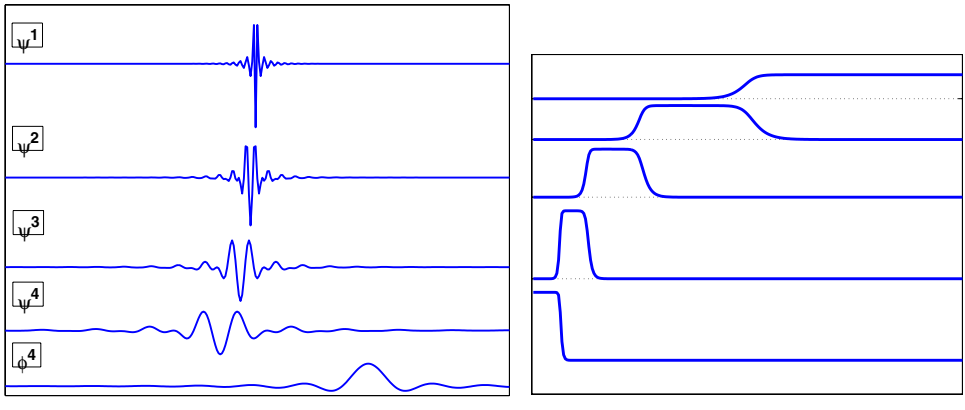


Fig. 2. Left: The high-frequency wavelets $\psi^j(k), j = 1, \ldots, 4$, and the low-frequency wavelet $\Phi^4(k)$. Right: Their frequency responses (color online).

scale coefficients $h^2_{2m}$ and $h^2_{2m+1}$ are the "offsprings" of the coarsest scale coefficient $h^3_m$. In turn, their "offsprings" are the fine scale coefficients $h^1_{4m}$, $h^1_{4m+1}$, $h^1_{4m+2}$ and $h^1_{2m+3}$. Thus, the coarsest scale coefficient $h^3_m$ is the root of the tree

$$\begin{array}{ccc} & h^2_{2m} \nearrow & h^1_{4m} \\ & & \searrow h^2_{4m+1} \\ h^3_m & & \\ & h^2_{2m+1} \nearrow & h^1_{4m+2} \\ & & \searrow h^2_{4m+3} \end{array} . \tag{2.1}$$

The wavelet transform of a two-dimensional array $\mathbf{T} = \{t_{n,m}\}$ of size $N \times M$ is implemented as a tensor product of 1D wavelet transform. First, a pair of filters $L$ and $H$ is applied to the columns of $\mathbf{T}$ and the results are downsampled. The coefficients arrays $\mathbf{W}^1_L$ and $\mathbf{W}^1_H$ 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 sub-arrays $\mathbf{W}_{LL}^2$, $\mathbf{W}_{LH}^2$, $\mathbf{W}_{HL}^2$ 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. 3.

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

$$lh_{n,m}^J \swarrow \begin{matrix} lh_{2n,2m}^{J-1} \\ lh_{2n,2m+1}^{J-1} \\ lh_{2n+1,2m}^{J-1} \\ lh_{2n+1,2m+1}^{J-1} \end{matrix} \searrow, \quad hl_{n,m}^J \swarrow \begin{matrix} hl_{2n,2m}^{J-1} \\ hl_{2n,2m+1}^{J-1} \\ hl_{2n+1,2m}^{J-1} \\ hl_{2n+1,2m+1}^{J-1} \end{matrix} \searrow, \quad hh_{n,m}^J \swarrow \begin{matrix} hh_{2n,2m}^{J-1}, \\ hh_{2n,2m+1}^{J-1} \\ hh_{2n+1,2m}^{J-1} \\ hh_{2n+1,2m+1}^{J-1} \end{matrix} \searrow . \tag{2.2}$$

Similar relations exist between the coefficients from scales $J-1$ and $J-2$, and so on. These relations for $J=3$ are illustrated in Fig. 4.

Thus, each coefficient $lh_{n,m}^J \in \mathbf{W}_{LH}^J$, $hl_{n,m}^J \in \mathbf{W}_{HL}^J$ or $hh_{n,m}^J \in \mathbf{W}_{HH}^J$ from the coarse scale can be treate 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.[30] This codec takes advantage of the self-similarity between wavelet coefficients across the decomposed scales and their decay toward high frequency scales. One of the most efficient algorithms, which is based on the zerotree concept, is the SPIHT algorithm.[28] This algorithm combines adaptive quantization of wavelet coefficients with coding. The produced bitstream is further compressed

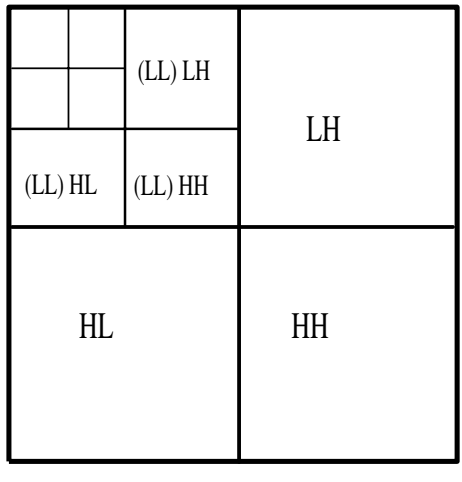


Fig. 3. Three-scale layout coefficients of a 2D wavelet transform.

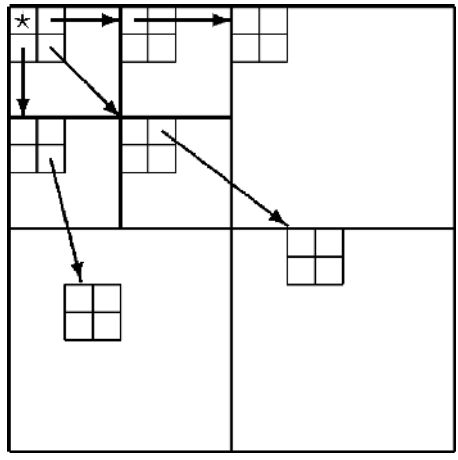

Fig. 4. Relationship among wavelet coefficients from different scales.

losslessly by the application of an adaptive arithmetic coding.[27] 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. To fit these codecs, the array of the mixed transform coefficients is organized in a wavelet-like way. It requires to reorder 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:

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

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

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

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

Assume we are given an $N \times M$ data array $\mathbf{T}$, where $N = 2^k Q$, $M = 2^J R$. We define the partition $P$ of the interval $I \triangleq [0, 1, \ldots, N-1]$ 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 $\mathbf{c}^i \triangleq \{c_n^i\}_{n=0}^{2^k-1}$. Each of them can be a bit-wise partitioning $\mathbf{c}^i = \bigcup_{\beta=0}^{k} \mathbf{b}_\beta^i$ as in Eq. (3.2) 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 $\beta$ 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:

$$\mathbf{c} \to \mathbf{b} \triangleq \bigcup_{\beta=0}^{k} \mathbf{b}_\beta, \quad \mathbf{b}_\beta \triangleq \bigcup_{i=1}^{Q} \mathbf{b}_\beta^i. \tag{3.3}$$

Thus, we get $\mathbf{b}_0 \triangleq (c_0^1, c_0^2, \ldots, c_0^Q)^T$, $\mathbf{b}_1 \triangleq (c_1^1, c_1^2, \ldots, c_1^Q)^T$, $\mathbf{b}_2 \triangleq (c_2^1, c_3^1; c_2^2, c_3^2; \ldots; c_2^Q, c_3^Q)^T$, and so on. This rearrangement is illustrated in Fig. 5. 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

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

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

We perform this LCT coefficients reordering for all the columns of the array $\mathbf{C}$. Thus, we get

$$\mathbf{C} \to \mathbf{B} = \bigcup_{\beta=0}^{k} \mathbf{B}_\beta, \quad \mathbf{B}_\beta \triangleq \bigcup_{i=1}^{Q} \mathbf{B}_\beta^i. \tag{3.5}$$

Then, each row in the array $\mathbf{B}$ is decomposed into scale $J$ by the application of the wavelet transform. This produces the hybrid LCT-wavelet coefficients array denoted as $\mathbf{CW}$. The structure of $\mathbf{CW}$ is similar to the the structure of a 2D wavelet coefficients array, where the transform on the columns decomposed to scale $k-1$, while the transform on the rows was decomposed till scale $J$. A three scales decomposition structure of the $\mathbf{CW}$ array is illustrated in Fig. 6.

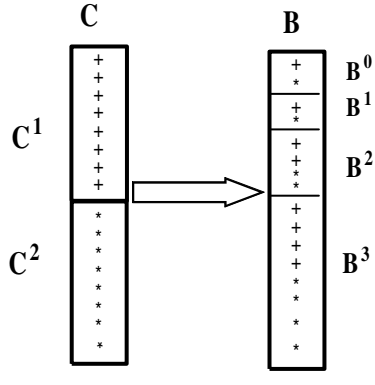


Fig. 5. Reordering scheme of the LCT coefficients.

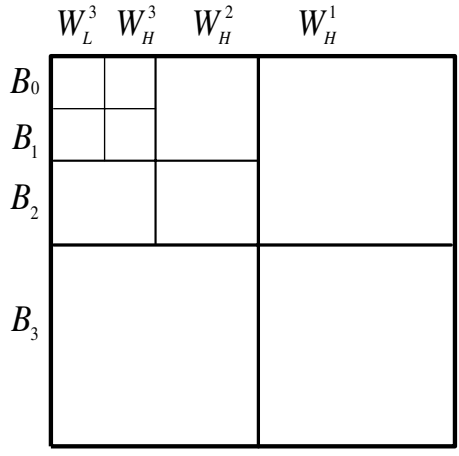


Fig. 6. Three-scales of the LCT-wavelet coefficients layout.

The array **CW** is the input to SPIHT or JPEG2000 coders that implement the quantization followed by the implementation of the entropy coding.

If the data arrays have oscillating structures in both vertical and horizontal directions then the LCT application to both directions followed by reordering of the transform coefficients in both directions is recommended. We call this transform the reordered LCT denoted by (RLCTA).

## 4. Experimental Results

The hybrid and the RLCTA algorithms were applied to compress different data types: seismic, hyper-spectral, fingerprints and multimedia images.

We compared between the performances of these algorithms where the quantization and entropy coding was borrowed from the SPIHT codec. They were compared with 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 in all the transforms. An additional experimental goal was to compare between the performance of the Butterworth wavelet transforms[3,4] and that of the 9/7 biorthogonal wavelet transform.[12,1]

The choice of a bell in the LCT construction (see Sec. 2.1) has some effect on the performance of the LCT and, consequently, on the hybrid algorithm. A library of bells was introduced in Ref. 22. A comparative study of their effects on the performance of an LCT-based image compression algorithm was given in Ref. 24. 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 $b(x) = \sin \frac{\pi}{2}(x + 1/2)$.[5]

**All the transforms in the paper have the following notation:**

**Jpeg2000 standard** is denoted by **JP2k**.

**2D wavelet transforms using 9/7 filters** is denoted by **W9/7**.

**2D wavelet transforms of Butterworth wavelet transform with M vanishing moments** is denoted by **WButt/M**.

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

**2D RLCTA: RLCTA/Q/P** denotes the 2D LCT transform, which partitions the image into $Q \times P$ rectangles where $Q$ is the partition in one direction and $P$ is the partition in the other direction. The transform coefficients were reordered to fit the SPIHT encoding.

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

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

in addition to a visual inspection. The SPIHT codec was enhanced by incorporating into it an adaptive arithmetic coder,[27] which is a popular entropy coder. The bold 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*

In the experiments, we used a stacked common mid point (CMP) data section. We display the original section of size $512 \times 512$ and a fragment of size $200 \times 200$ from it in Fig. 7.

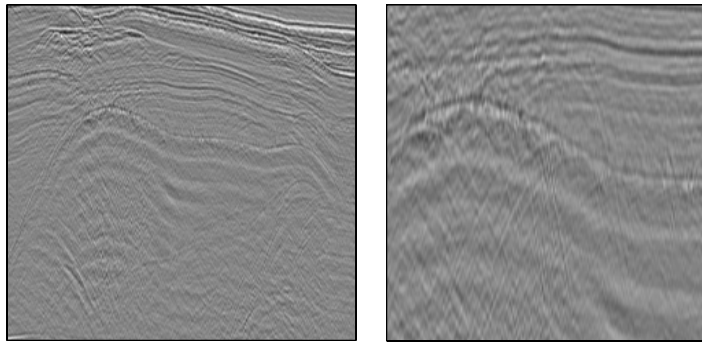

Fig. 7. The original stacked CMP section. Left: The whole section. Right: A fragment from the left image. The $X$-axis corresponds to the horizontal direction along the earth surface while the $Y$-axis corresponds to the vertical (depth) direction.

Table 1. The PSNR values after decompression of the stacked CMP seismic section.

| 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** |

Five types of transforms were used: **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 1.

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

Figure 8 displays fragments from the reconstruction of the CMP section in Fig. 7 (right) after the application of the **JP2k** compression algorithm where the compression ratio is 1/4 bit per pixel (left image in Fig. 8) and from the application of the **RLCTA/8/8**–SPIHT compression (right image in Fig. 8). The **RLCTA/8/8**–SPIHT compression retains the main structure of the data unlike the outcome from the application of the **JP2k** algorithm.

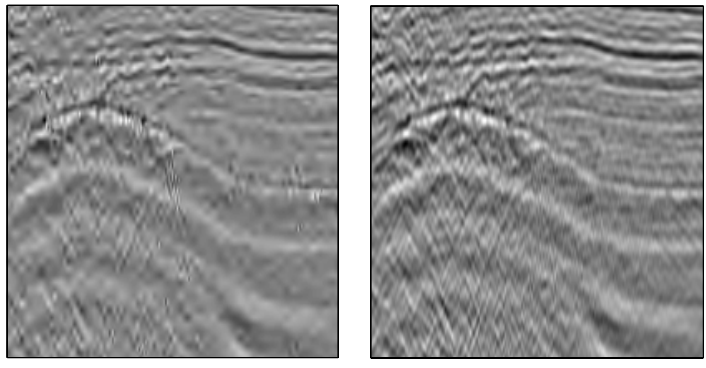

Fig. 8. A fragment from the reconstructed CMP section in Fig. 7 (right). Left: Output from the application of the **JP2k** algorithm. Right: The output from the application of the **RLCTA/8/8**–SPIHT algorithm. The transforms coefficients were encoded by 1/4 bit per pixel. The $X$-axis corresponds to the horizontal direction along the earth surface while the $Y$-axis corresponds to the vertical (depth) direction.

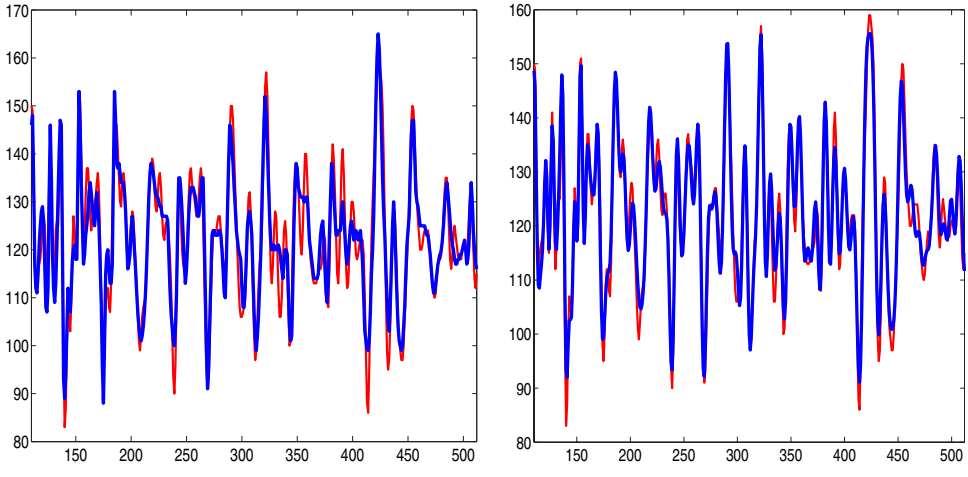


Fig. 9. The differences between the original line #110 from the vertical trace in the CMP section (Fig. 7 (right)) and the reconstructed line after the applications of different compression algorithms. Dotted line: Original. Solid line: Restored. Left: Restored after the application of **JP2k** algorithm. Right: Restored after the application of the **HButt/10/8**–SPIHT algorithm. Both were compressed to 1/2 bit per pixel (color online).

Figure 9 displays line #110 from the vertical trace from the original section versus the same line from the reconstructed section from 1/2 bit per pixel compression by the applications of **JP2k** and **HButt/10/8**–SPIHT based algorithms.

We observe that the reconstructed trace in the right image in Fig. 9 is very close to the original trace. This is not the case for the trace in the left image.

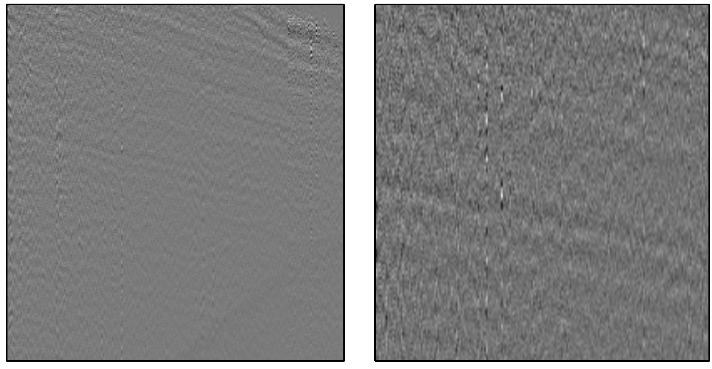

Fig. 10. The original marine shot gather (MSG) section. Left: The whole section. Right: A fragment from the section on the left. The $X$-axis corresponds to the horizontal direction along the earth surface while the $Y$-axis corresponds to the vertical (depth) direction.



#### 4.1.2. *Marine shot gather data section*

For another seismic experiments, we used a marine shot gather (MSG) data section of size $512\times512$. 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 $512\times512$ and fragment of size $200\times200$ from it in Fig. 10.

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 2.

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. The **HButt/10/8** transform outperforms **H9/7/10** and **JP2k**. Figure 11 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. The **HButt/10/8** transform retains the structure of the data much better than **JP2k**.

Figure 12 displays the line #270 from the vertical trace from the original section taken from Fig. 10 (right) versus the same line taken from the reconstructed section after the application of **JP2k** and **HButt/10/8**+SPIHT algorithms.

Table 2. The PSNR values after decompression of the MSG seismic section (Fig. 10 (right)).

| 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 |

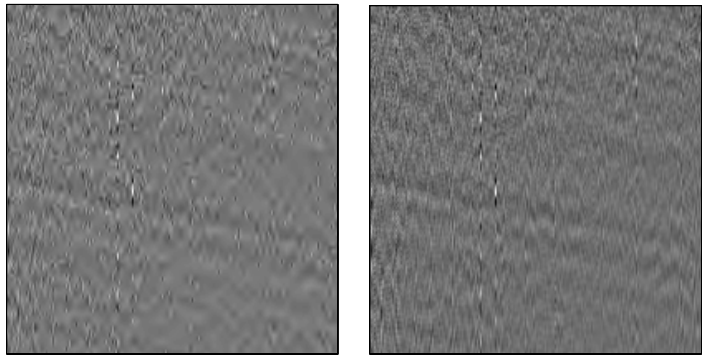

Fig. 11. A fragment from the reconstructed MSG section (Fig. 10 (right)). Left: Reconstruction from the application of **JP2k**, PSNR = 34.63 dB. Right: The reconstruction from the application of the Hybrid **HButt/10/8**+SPIHT, PSNR = 35.23 dB. The compression ratio is 1 bit per pixel. The $X$-axis corresponds to the horizontal direction along the earth surface while the $Y$-axis corresponds to the vertical (depth) direction.

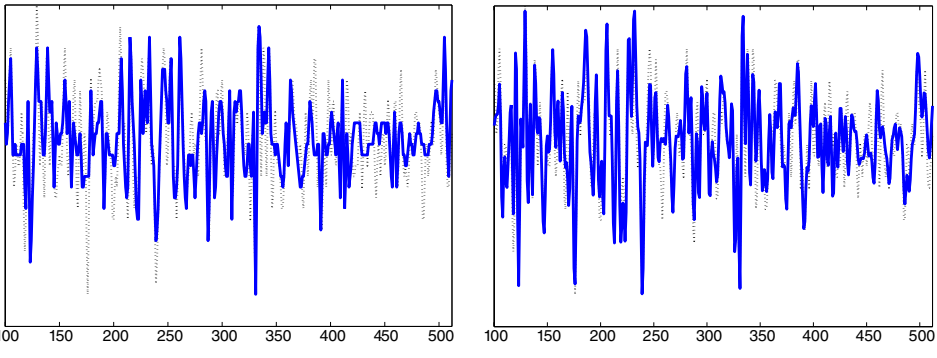


Fig. 12. Line #270 from the vertical trace in the MSG section. Dotted lines: Original. Solid line: Restored. Left: After the application the **JP2k** algorithm. Right: The same after the application of the **HButt/10/8**+SPIHT algorithm. Both are compressed to 1 bit per pixel (color online).

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 ($\approx$ 200)

wavebands from the ground surface. This type of camera can also be placed in a satellite. We used the SPECIM camera.[31] In this camera, the spectrum of intensities for all the wavebands is assigned at once to each spatial pixel that is called multipixel. 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 a 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 an 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 13 displays an urban ground scene at one of the $X/Z$ multipixel planes. This image was captured by the Specim[31] camera placed on a airplane flying 10,000 feet above sea level. The resolution is 1.5 meter/pixel, 3000 × 300 pixels per waveband with 180 wavebands.

Five transforms types 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

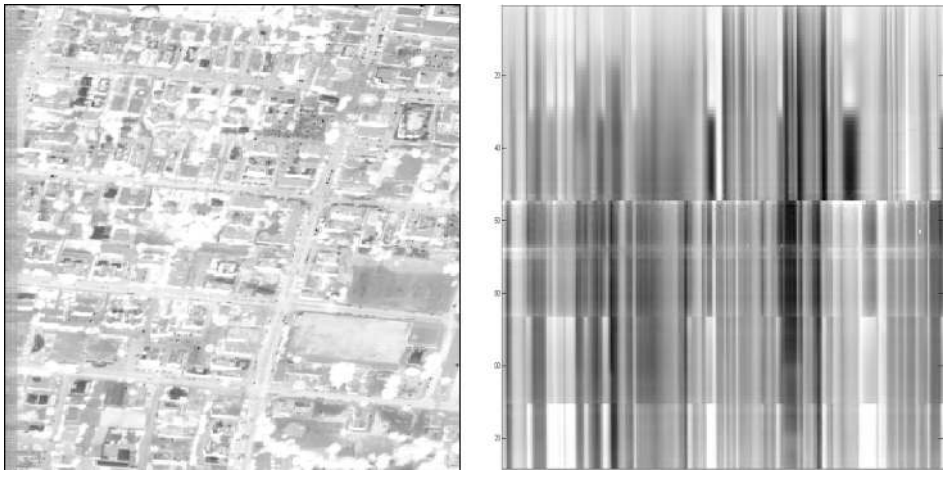

Fig. 13. An hyper-spectral image. Left: One horizontal plane from the data cube. Right: A multipixel plane. Collection of all multipixels that belong to one line from the horizontal plane. The $X$-axis corresponds to the direction along a horizontal line of pixels in the left image while the $Y$-axis indicates the wavebands.

Table 3. PSNR values for the hyper-spectral compression of $X/Z$ plane #200 (plane 200 from the 3000 planes of the whole hyper-spectral cube) from the urban scene Fig. 13 (right).

| 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 |

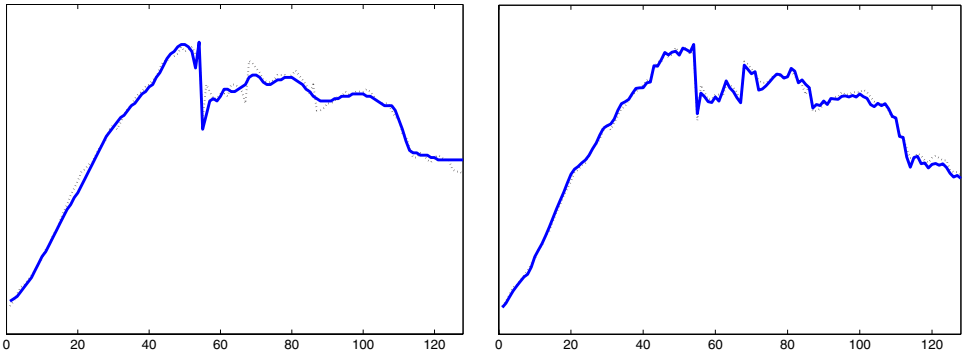


Fig. 14. One multipixel from $X/Z$ plane #200. Dotted lines: Original. Solid line: Restored. Left: After the application of **JP2k**. Right: After the application of **HButt/10/8**+SPIHT encoding/decoding. The compression ratio for both is 0.25 bits per pixel (color online).

**JP2k**. Table 3 presents the comparison results on the $X/Z$ plane #200. The performances are given in PSNR values.

We observe that the hybrid transforms with SPIHT produce PSNR values, which are higher than the results from the applications of the **JP2k** algorithm, which, in turn, outperforms the wavelet transforms. Figure 14 displays a multipixel from plane #200 from 3000 planes in Fig. 13 (left), which was reconstructed after the application of **JP2k** (left image) and **HButt/4/8** (right image) transforms followed by SPIHT encoding/decoding where the compression ratio is 0.25 bit per pixel.

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

### 4.3. *Fingerprints*

The 2D wavelet, hybrid and LCT transforms followed by SPIHT coding were applied to compress a fingerprint image of size $512 \times 512$, shown in Fig. 15. 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.

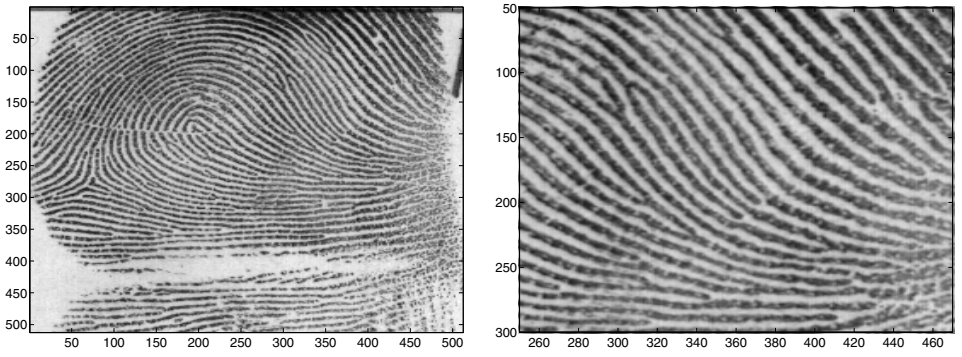

Fig. 15. The original fingerprint image. Left: The whole image. Right: A fragment from the left image.

Table 4. PSNR values for fingerprint compression.

| 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** |

The performances of the SPIHT based algorithms were compared with the performance of JPEG2000 compression standard. The LCT transforms partition the image into 16 horizontal and vertical rectangles. The achieved PSNR values are given in Table 4.

We can see from Table 4 that **RLCTA/16/16**+SPIHT outperforms all the other algorithms in all bitrates. This is due to better handling of the oscillatory events in comparison to wavelet transforms based algorithms.

Figure 16 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 that in the left fragment.

Figure 17 displays the column line #300 from the original section versus the same 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.

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

### 4.4. *Compression of multimedia images*

The hybrid compression algorithms also proved to be efficient for multimedia images. Their superior performances are more evident for images that have

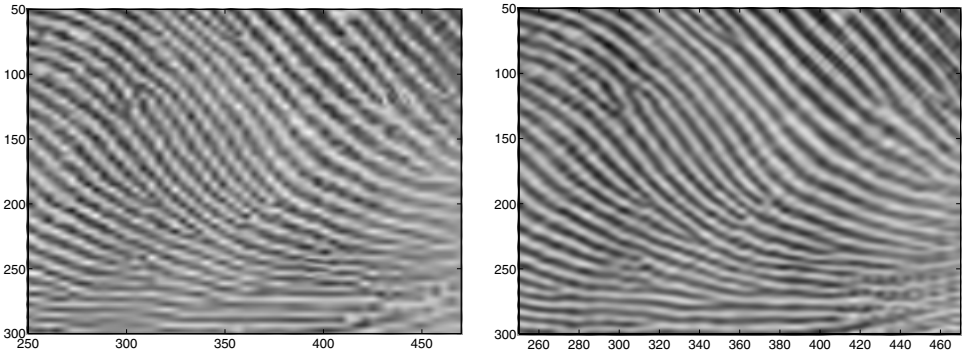


Fig. 16. A fragment from the reconstructed fingerprint image. Left: After the application of **JP2k**, PSNR = 23.54. Right: After the application of **RLCTA/16/16**+SPIHT, PSNR = 24.55. The compression ratio is 1/8 bit per pixel.

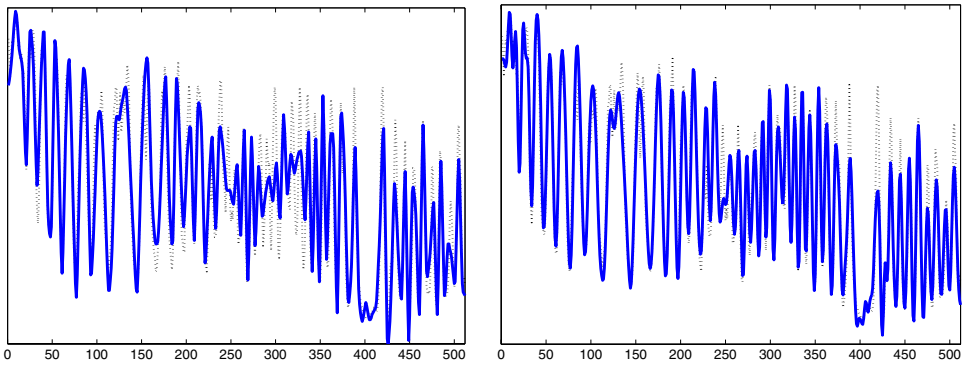


Fig. 17. Column line #300 from the fingerprint image. Dotted line: Original. Solid line: Restored. Left: The two lines after the application of **JP2k**. Right: The two lines after the application of **RLCTA/16/16**+SPIHT (right). The compression ratio is 1/8 bit per pixel (color online).

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 textured areas. **Barbara image** is of size $512 \times 512$ and its fragment is of size $280 \times 280$. They are displayed in Fig. 18. Each pixel in this image has 8 bits.

This image comprises areas with oscillatory textures. We compared between the performances of 6 transforms where the hybrid and RLCTA transforms partition the vertical direction into $Q = 16$ horizontal rectangles each of height 32. The achieved PSNR values are given in Table 5.

**RLCTA/16/16**+SPIHT, which produced the best results except the result for 1 bit per pixel, retains the fine texture of the image much better than **JP2k**. This

Fig. 18. The original "Barbara" image. Left: The whole image. Right: A fragment from the left image.

Table 5. The PSNR values after the compression/decompression of "Barbara".

| 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 |

fact is illustrated in Fig. 19, which displays fragments of the reconstructed "Barbara" image after the applications of **JP2k** (left) and **RLCTA/16/16**+SPIHT (right) algorithms. The compression ratio is 1/8 bit per pixel.

Figure 20 displays the column line #246 from the original section versus this column line 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.

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

**Elaine image** is of size $512 \times 512$ and its fragment of size $236 \times 256$ are displayed in Fig. 21. Each pixel has 8 bits.

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

The highest PSNR values in all bitrates except for 1/8 bit per pixel were produced by **RLCTA/8/8**+SPIHT, while the 2D wavelet transform with Butterworth wavelet **WButt/4**+SPIHT was better with 1/8 bit per pixel. At bitrates 1/8 and 1/4 bit per pixel, reconstruction from **RLCTA/8/8**+SPIHT and hybrid algorithms

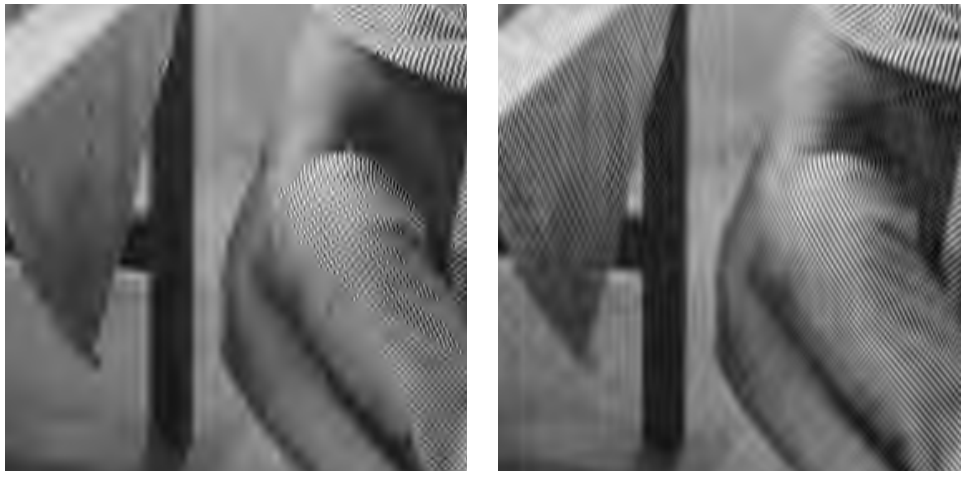

Fig. 19. A fragment from the reconstructed "Barbara" image after the applications of: Left: **JP2k**. Right: **RLCTA/16/16**+SPIHT. The compression ratio is 1/8 bit per pixel.

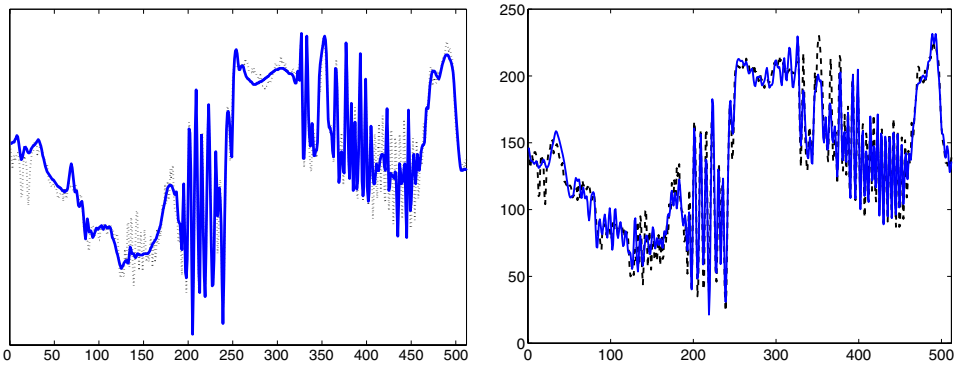


Fig. 20. Column line #246 from Barbara image. Dotted line: Original. Solid line: Restored. Left: Restored after the application of **JP2k**. Right: Restored after the application of **RLCTA/16/16**+SPIHT (right). The compression ratio is 1/4 bit per pixel (color online).

produced a better visual quality of the images in comparison to **JP2k**, which over smoothed the images. This fact is illustrated in Fig. 22, which displays fragments of "Elaine" image that were reconstructed after the application of **JP2k** and of **RLCTA/8/8**+SPIHT. The compression ratio is 1/8 bit per pixel.

Over smoothing by **JP2k** is clearly seen in Fig. 23, which displays the 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.

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

Fig. 21. The original "Elaine" image. Left: The whole image. Right: A fragment from the right image.

Table 6. PSNR values after compression/decompression of "Elaine".

| bit/pixel | JP2k | W9/7 | WButt/4 | H9/7/16 | HButt/10/16 | RLCTA/8/8 |
|---|---|---|---|---|---|---|
| 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** |

Fig. 22. 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.

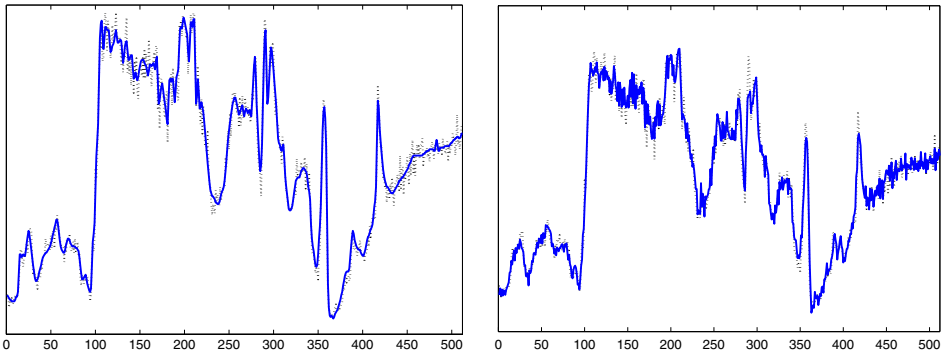


Fig. 23. Column line #280 from the Elaine image. Dotted line: Original. Solid line: Restored. Left: Restored after the application of **JP2k**. Right: Restored after the application of **RLCTA/8/8**+SPIHT (right). The compression ratio is 1/4 bit per pixel (color online).



## 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 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 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 based filters. 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 comprises a fine texture as in the "Elaine" image.

The algorithm retains fine oscillating events even at a low bitrate. This 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 coefficients reordering. 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. M. Antonini, M. Barlaud, P. Mathieu and I. Daubechies, Image coding using wavelet transform, *IEEE Trans. Image Processing* **1**(2) (1992) 205–220.

2. A. Z. Averbuch, F. Meyer, J.-O. Stromberg, R. Coifman and A. Vassiliou, Efficient compression for seismic data, *IEEE Trans. Image Processing* **10**(12) (2001) 1801–1814.
3. 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.
4. A. Averbuch and 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.) (Hindawi Publishing Corp., NY, 2007), pp. 1–56.
5. A. Averbuch, G. Aharoni, R. Coifman and M. Israeli, Local cosine transform — A method for the reduction of blocking effects in JPEG, *J. Mathematical Imaging and Vision*, *Special Issue on Wavelets* **3** (1993) 7–38.
6. A. Averbuch, L. Braverman, R. Coifman, M. Israeli and A. Sidi, Efficient computation of oscillatory integrals via adaptive multiscale local Fourier bases, *Applied and Computational Harmonic Analysis* **9**(1) (2000) 19–53.
7. A. Averbuch, L. Braverman, M. Israeli and R. Coifman, On efficient computation of multidimensional oscillatory integrals with local Fourier bases, *J. Nonlinear Analysis* **47** (2001) 3491–3502.
8. A. Averbuch, E. Braverman, R. Coifman and M. Israeli, Fast evaluation of 2-D and 3-D oscillatory integrals with local Fourier bases, *Int. J. Pure and Applied Mathematics* **3**(1) (2002) 1–28.
9. R. Bensalma and M. C. Larabi, A perceptual metric for stereoscopic image quality assessment based on the binocular energy, *Multidimensional System Signal Processing*, (2012), DOI: 10.1007/s11045-012-0178-3.
10. C. M. Brislawn, Classification of nonexpansive symmetric extension transforms for multirate filter banks, *Applied and Computational Harmonic Analysis* **3**(4) (1996) 337–357.
11. C. M. Brislawn, The FBI fingerprint image compression specification, *Wavelet Image and Video Compression*, P. Topivala (ed.), (Springer, New York, 2002), pp. 271–288.
12. A. Cohen, I. Daubechies and J.-C. Feauveau, Biorthogonal bases of compactly supported wavelets, *Comm. on Pure and Appl. Math.* **45** (1992) 485–560.
13. R. R. Coifman and Y. Meyer, Remarques sur l'analyse de Fourier a fenêtre (C. R. Acad. Sci., 1991), 259–261.
14. P. L. Donoho, R. A. Ergas and J. D. Villasenor, High-performance seismic trace compression, in *65th Ann. Int. Mtg., Soc. Expl. Geophys.*, *Expanded Abstracts* (1995), pp. 160–163.
15. I. Daubechies, *Ten Lectures on Wavelets* (SIAM, 1992).
16. Q. Du and J. E. Fowler, Hyperspectral image compression using JPEG2000 and principal component analysis, *IEEE Geoscience and Remote Sensing Letters* **4**(4) (2007) 201–205.
17. 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 Trans. on Consumer Electronics* **44**(3) (1998) 500–508.
18. JPEG 2000 Standard, http://www.jpeg.org/jpeg2000/.
19. M. F. Khéne and S. H. Abdul-Jauwad, Efficient seismic compression using the lifting scheme, in *70th Ann. Int. Mtg., Soc. Expl. Geophys.*, *Expanded Abstracts* (2000), pp. 2052–2054.
20. H. S. Malvar, Fast progressive image coding without wavelets, in *Proc. IEEE Data Compression Conference*, J. A. Storer and M. Cohn (eds.) (Snowbird, UT, March 2000), pp. 243–252.

21. S. Mallat, *A Wavelet Tour of Signal Processing: The Sparse Way*, 3rd edn. (Elsevier, 2009).
22. G. Matviyenko, Optimized local trigonometric bases, *Applied and Computational Harmonic Analysis* **3** (1996) 301–323.
23. F. G. Meyer, Fast compression of seismic data with local trigonometric bases, *Proc. SPIE 3813*, *Wavelet Applications in Signal and Image Processing VII*, eds. A. Aldroubi, A. F. Laine and M. A. Unser (1999), pp. 648–658.
24. F. G. Meyer, Image compression with adaptive local cosines: A comparative study, *IEEE Trans. on Image Proc.* **11**(6) (2002) 616–629.
25. 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* **3**(1) (2006) 125–129.
26. R. Rao and P. Yip, *Discrete Cosine Transform*: *Algorithms*, *Advantages*, *Applications* (Academic Press, 1990).
27. A. Said, Introduction to Arithmetic Coding — Theory and Practice, HP Lab, Published as a chapter in Lossless Compression Handbook by Khalid Sayood (Academic Press, 2004).
28. A. Said and W. W. Pearlman, A new, fast and efficient image codec based on set partitioning in hierarchical trees, *IEEE Trans. Circ. and Syst. for Video Tech.* **6** (1996) 243–250.
29. V. P. Shah, J. E. Fowler and N. H. Younan, Tarp filtering of block-transform coefficients for embedded image coding, in *Proc. IEEE Int. Conf. Acoustics*, *Speech*, *and Signal Processing*, Toulouse, France, May 14–16, 2006, Vol. 2 (IEEE, 2006), pp. 21–24.
30. J. M. Shapiro, Embedded image coding using zerotree of wavelet coefficients, *IEEE Trans. Sign. Proc.* **41** (1993) 3445–3462.
31. Specim camera, http://www.specim.fi/.
32. A. Vassiliou and M. V. Wickerhauser, Comparison of wavelet image coding schemes for seismic data compression, in *67th Ann. Int. Mtg., Soc. Expl. Geophys.*, *Expanded Abstracts* (1997), 1334–1337.
33. Y. Wang and R.-S. Wu, Seismic data compression by an adaptive local cosine/sine transform and its effect on migration, *Geophysical Prospecting* **48** (2000) 1009–1031.
34. Z. Xiong, O. Guleryuz and M. T. Orchard, A DCT-based embedded image coder, *IEEE Signal Processing Letters* **3** (1996) 289–290.
35. V. A. Zheludev, D. D. Kosloff and E. Y. Ragoza, Compression of segmented 3D seismic data, *Int. J. Wavelets*, *Multiresolution and Information Processing* **2**(3) (2004) 269–281.