ISSN 1392 – 124X INFORMATION TECHNOLOGY AND CONTROL, 2012, Vol. 41, No. 3

# A New Le Gall Wavelet-Based Approach to Progressive Encoding and Transmission of Image Blocks

**Deividas Kančelkis, Jonas Valantinas**

*Department of Applied Mathematics, Kaunas University of Technology*
*Studentų St 50-325c, LT-51368 Kaunas, Lithuania*
*e-mail: deividas.kancelkis@stud.ktu.lt, jonas.valantinas@ktu.lt*

crossref http://dx.doi.org/10.5755/j01.itc.41.3.860

**Abstract.** In this paper, a modified version of the discrete reversible (integer-to-integer) Le Gall wavelet transform (DLGT), distinguishing itself by apparently improved space localization properties and visibly extended potential capabilities, is proposed. The key point of the proposal – ensuring full decorrelation of Le Gall wavelet coefficients across the lower scales. Based on the latter circumstance, a novel exceptionally fast procedure for computing the integer DLGT spectra of the selected image blocks (regions of interest - ROI) is presented. It is shown that the new developments can be efficiently applied to progressive encoding and transmission of image blocks. Progressive encoding and transmission of image blocks is achieved by first transmitting a "rough" estimate of the original image, then sending further details related to one or another image block (ROI). To translate the idea into action, the zero-tree-based encoder SPIHT (Set Partitioning in Hierarchical Trees) with an improved quad-tree analysis scheme is employed.

**Keywords**: digital images; discrete wavelet transforms; Le Gall wavelets; Lifting schemes; image processing; progressive image encoding; ROI.

## 1. Introduction

Progressive encoding and subsequent transmission of digital images refers to image compression techniques that allow both the original image reconstruction without loss of any detail and the construction of image approximations (estimates) with the accuracy level depending on the amount of data available. Lossless compression is highly important for images obtained at a great cost, such as space or medical images. In this case, even negligible loss of data may destroy some details needed during further processing, or add artefacts that lead to misinterpretation. In various problem-oriented applications fast extraction of image estimates with a priori prescribed levels of accuracy may also be necessary. Surely, the speed of this operation depends on the amount of data that should be retrieved from the storage for image reconstruction. On the other hand, in the case of non-progressive compression, neither an image estimate nor the original image is available until the whole image data are retrieved.

Nowadays, most modern image compression techniques employ a discrete wavelet transform, usually followed by quantization and entropy coding [1-6]. Especially practicable are image coders that allow progressive encoding with an embedded bit stream, such as the embedded zero-tree wavelet (EZW) image coder, suggested by Shapiro [7]. With embedded bit streams, the wavelet coefficients are encoded in bit planes, with the most significant bit planes being transmitted first. In that way, the decoder can stop decoding at any point in the bit stream, and it will reconstruct an image with required level of accuracy. There have been many variants of zero-tree-based progressive image coders since Shapiro introduced his algorithm in 1993. The SPIHT (Set Partitioning in Hierarchical Trees) algorithm, proposed by Said and Pearlman, deserves special attention in this class of coders because it provides the highest image quality, lossless compression, progressive image transmission and so forth [8]. The SPIHT algorithm appears to be very useful for applications where the user can quickly inspect the image and decide if it is good enough to be saved, or need a refinement.

Some other interesting ideas and developments in the wavelet-based image compression field are presented in [9-14].

The locally progressive image coding idea is not a new one. The possibility of defining regions of interest in an image is a significant feature of the latest image compression standard JPEG 2000 [2]. These regions of interest are coded with better quality than the rest of the image. This is done by scaling the wavelet coefficients so that the bits associated with the regions of interest are placed in higher bit planes. So, the regions of interest are decoded before the part of the image that is not of interest.

In this paper, the locally progressive image encoding and transmission is achieved by first transmitting a “rough” estimate of the original image to the user, then sending further details related to one or another selected block (ROI) of the image. The idea (proposed approach) explores improved space localization properties of the new version of the discrete reversible (integer-to-integer) Le Gall wavelet transform (DLGT) and the newly developed exceptionally fast procedure for finding the DLGT spectra of the selected (at the user’s request) image blocks (ROI). The latter procedure leans upon the assumption that the modified DLGT spectrum of the image under processing is known and plays a key role in ensuring reasonably high overall performance of the approach.

The rest of the paper is organized as follows. Section 2 introduces the DLGT transform and describes procedures for finding the DLGT spectrum of a digital image (signal). Section 3 passes a comment on factors ensuring improved space localization properties of the modified DLGT and presents a novel fast procedure (algorithm) for finding the integer DLGT spectra of the selected image blocks (ROI). Experimental results and some commentary on the energy compaction property of the new version of DLGT are presented in Section 4.

## 2. Implementing the discrete reversible Le Gall wavelet transform

Any discrete wavelet transform (DWT) represents an iterative procedure. Each iteration (step) of the DWT applies the scaling function to the data input (digital signal). If the original signal $X$ has $N$ ($N = 2^n, n \in \mathrm{N}$) values, the scaling function will be applied in the wavelet transform iteration to calculate $N/2$ averaged (smoothed) values. In the ordered wavelet transform the smoothed values are stored in the upper half of the $N$ element input vector. The wavelet function (in each step of the wavelet transform) is also applied to the input data. If the original signal has $N$ values, the wavelet function will be applied to calculate $N/2$ differences (reflecting change in the data). In the ordered wavelet transform, the wavelet (differenced) values are stored in the lower half of the $N$ element input vector. On the subsequent iteration, both mentioned functions (scaling and wavelet) are applied repeatedly to the ordered set of smoothed values calculated during the preceding iteration. After a finite number of iterations ($n$ steps) the DWT spectrum $Y$ of the digital signal $X$ is found. The vector $Y$ comprises the only smoothed value (obtained in the $n$-th iteration) and the ordered set of differenced values (obtained in the $n-1$ preceding iterations).

The discrete reversible (integer-to-integer) Le Gall wavelet transform (DLGT) has five scaling and three wavelet function coefficients. The scaling function coefficients are - $h_0 = -1/8$ , $h_1 = 1/4$ , $h_2 = 3/4$ , $h_3 = 1/4$ and $h_4 = -1/8$ , while the wavelet function coefficient values are - $g_0 = -1/2$ , $g_1 = 1$ , and $g_2 = -1/2$ . The scaling and wavelet functions are calculated by taking the (integer) scalar product of the coefficients and five or three input data values (Figure 1).

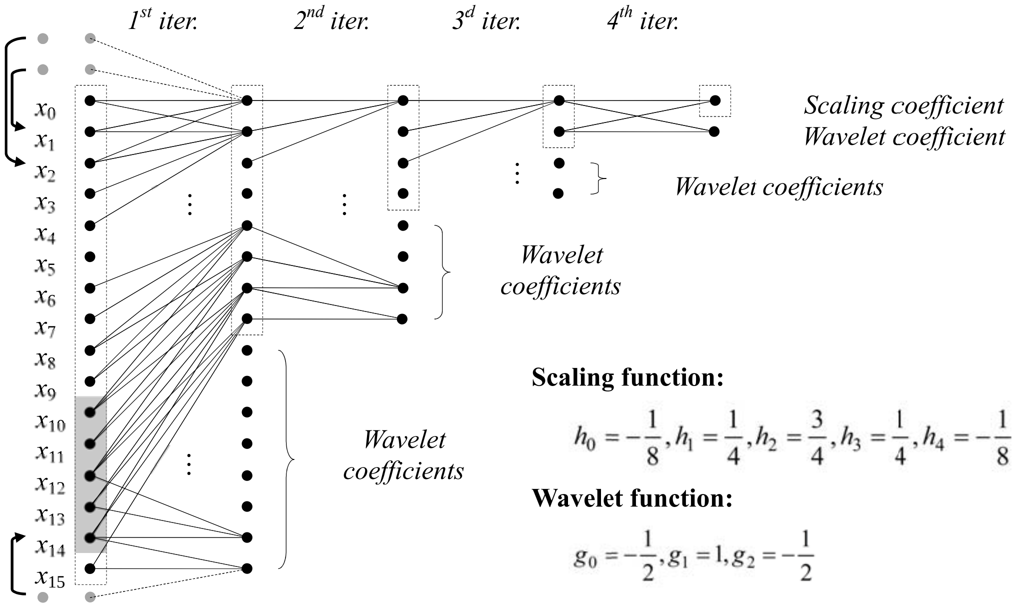



**Figure 1**. Finding the DLGT spectrum of a digital signal ( $N = 16$ )

In practice, to compute the integer DLGT spectrum $Y$ of the digital input signal $X$ of size $N = 2^n$ ( $n \in \mathrm{N}$ ), an efficient iterative procedure (Lifting Scheme [15, 16]) can be applied.

Let us assume that $S^{(i)} = (s_0^{(i)}\, s_1^{(i)} \ldots s_{2^{n-i}-1}^{(i)})^T$ and $D^{(i)} = (d_0^{(i)}\, d_1^{(i)} \ldots d_{2^{n-i}-1}^{(i)})^T$ stand for the result of application of the Le Gall scaling and wavelet functions to the data vector $S^{(i-1)}$, respectively ( $i \in \{1,2,\ldots,n\}$ ; $S^{(0)} = X$ , i.e. $s_k^{(0)} = x_k$ , for all $k = 0,1,\ldots,N-1$). Then:

$$d_k^{(i)} = s_{2k+1}^{(i-1)} - \left\lfloor \frac{1}{2}\left(s_{2k}^{(i-1)} + s_{2k+2}^{(i-1)}\right) \right\rfloor,$$

$$s_k^{(i)} = s_{2k}^{(i-1)} + \left\lfloor \frac{1}{4}\left(d_{k-1}^{(i)} + d_k^{(i)}\right) + \frac{1}{2} \right\rfloor,$$

for all $k = 0,1,\ldots,2^{n-i}-1$ ; here $s_{2^{n-i+1}}^{(i-1)} := s_{2^{n-i+1}-2}^{(i-1)}$ , $d_{-1}^{(i)} := d_0^{(i)}$ and $i \in \{1,2,\ldots,n\}$. Thus, the integer DLGT spectrum $Y$ of the digital signal $X$ is obtained in $n$ iterations and takes the form:

$$Y = (s_0^{(n)}\, d_0^{(n)}\, d_0^{(n-1)}\, d_1^{(n-1)}\, d_0^{(n-2)}\, d_1^{(n-2)}\, d_2^{(n-2)}\, d_3^{(n-2)} \ldots$$
$$\ldots\, d_0^{(1)}\, d_1^{(1)} \ldots d_{N/2-1}^{(1)})^T.$$

The inverse integer-to-integer DLGT is specified this way:

$$s_{2k}^{(i-1)} = s_k^{(i)} - \left\lfloor \frac{1}{4}\left(d_{k-1}^{(i)} + d_k^{(i)}\right) + \frac{1}{2} \right\rfloor,$$

$$s_{2k+1}^{(i-1)} = d_k^{(i)} + \left\lfloor \frac{1}{2}\left(s_{2k}^{(i-1)} + s_{2k+2}^{(i-1)}\right) \right\rfloor,$$

for all $k = 0,1,\ldots,2^{n-i}-1$ and $i \in \{1,2,\ldots,n\}$.

The "edge" problem, which takes place at both ends of the data vector $S^{(i-1)}$ ( $i \in \{1,2,\ldots,n\}$ ) and determines the partially localized nature of the discrete Le Gall wavelet transform, here (Lifting Scheme) is solved by treating the data vector as if it is mirrored at the ends (Figure 1).

In order to compute the integer DLGT spectrum of a two-dimensional digital image $X$ of size $N_1 \times N_2$ $(N_i = 2^{n_i},\ \ n_i \in \mathrm{N},\ i = 1,2)$ , the one-dimensional discrete reversible Le Gall wavelet transform should be applied $N_1 + N_2$ times, i.e. $N_2$ times along the first spatial axis and then $N_1$ times along the second spatial axis.

## 3. A novel approach to progressive encoding and transmission of image blocks

In the subsections below, we present a new approach to progressive encoding and transmission of grey-level image blocks. The essence of the approach – additional bit streams are used to improve quality not of the whole image under reconstruction but that of selected smaller image blocks (ROI). Implementation of the approach is based on the improved space localization properties of the modified integer DLGT and the newly developed fast procedure for finding (modified) integer DLGT spectra of the selected ROI.

### 3.1. Full decorrelation of Le Gall wavelet coefficients at lower scales

The characteristic feature of DWT is that individual basis functions are localized in space (compact support). Full localization in space (or, what is the same, full decorrelation of wavelet coefficients) means that numerical values of wavelet coefficients, computed on the same iteration (at the same scale!), are specified exclusively by image (signal) blocks that cover the whole image (signal) and do not overlap. Unfortunately, only the simplest Haar (Daubechies D2) wavelet transform, most frequently used in educational environment, is fully localized in space [17]. Higher order wavelet transforms (DLGT, Daubechies D4, CDF 9/7, etc. [11, 16]) are partially localized in space since image (signal) blocks which specify numerical values of any adjacent wavelet coefficients overlap (Figure 1; the overlapping part is coloured in grey). This gives some limitations to practical application of higher order DWT, especially in such areas where block-processing of digital data is preferable (e.g. image feature extraction, locally progressive image encoding, localization of defects in textural images.

In the paper, full decorrelation of Le Gall integer wavelet coefficients across the lower scales is achieved by partitioning the original data vector (signal) $X$ of size $N$ ( $N = 2^n, n \in \mathrm{N}$ ) into a finite set of non-overlapping blocks (ROI) of a priori prescribed size $2^p$ ( $1 \le p < n$ ) and by transferring the before-mentioned "edge" problem (Section 2) to those blocks. That is to say, when solving the "edge" problem, not only the (intermediate) data vector $S^{(i)}$ ( $i \in \{0,1,\ldots,p-1\}$ ) but also its component blocks of size $2^{p-i}$ are mirrored at the ends. This way, improved (full) localization in space is ensured for all Le Gall integer wavelet coefficients across the $i$ -th ( $i \in \{p, p+1,\ldots,n\}$ ) scale.

Following this perception, we have developed a new iterative procedure (Lifting Scheme) for computing the modified integer DLGT spectrum $Y$ of $X$ , namely (for all $k = 0,1,\ldots,2^{n-i}-1$ and $i \in \{1,2,\ldots,n\}$ ) :

$$d_k^{(i)} = \begin{cases} s_{2k+1}^{(i-1)} - s_{2k}^{(i-1)}, & k \in \{l_i - 1,\, 2l_i - 1, \ldots, q_i l_i - 1\}, \\ s_{2k+1}^{(i-1)} - \left\lfloor \frac{1}{2}\left(s_{2k}^{(i-1)} + s_{2k+2}^{(i-1)}\right) \right\rfloor, & \text{otherwise}, \end{cases}$$

$$s_k^{(i)} = \begin{cases} s_{2k}^{(i-1)} + \left\lfloor \frac{1}{2} d_k^{(i)} + \frac{1}{2} \right\rfloor, \\ \quad \text{for all } k \in \{0, l_i, 2l_i, \ldots, (q_i - 1) l_i\}; \\ s_{2k}^{(i-1)} + \left\lfloor \frac{1}{4}\left(d_{k-1}^{(i)} + d_k^{(i)}\right) + \frac{1}{2} \right\rfloor, \text{ otherwise}; \end{cases}$$

here: $l_i = 2^{p-i}$, $q_i = 2^{n-p}$, for $i = 1, 2, \ldots, p$, and $l_i = 1$, $q_i = 2^{n-i}$, for $i = p+1, p+2, \ldots, n$.

The new Lifting Scheme for the inverse modified DLGT has also been developed, and is presented below, i.e. (for all $k = 0, 1, \ldots, 2^{n-i} - 1$ and $i \in \{n, n-1, \ldots, 2, 1\}$):

$$s_{2k}^{(i-1)} = \begin{cases} s_k^{(i)} - \left\lfloor \frac{1}{2} d_k^{(i)} + \frac{1}{2} \right\rfloor, \\ \quad \text{for all } k \in \{0, l_i, 2l_i, \ldots, (q_i - 1) l_i\}, \\ s_k^{(i)} - \left\lfloor \frac{1}{4}\left(d_{k-1}^{(i)} + d_k^{(i)}\right) + \frac{1}{2} \right\rfloor, \text{ otherwise}, \end{cases}$$

$$s_{2k+1}^{(i-1)} = \begin{cases} d_k^{(i)} + s_{2k}^{(i-1)}, k \in \{l_i - 1, 2l_i - 1, \ldots, q_i l_i - 1\}, \\ d_k^{(i)} + \left\lfloor \frac{1}{2}\left(s_{2k}^{(i-1)} + s_{2k+2}^{(i-1)}\right) \right\rfloor, \text{otherwise}. \end{cases}$$

Thus, in $n$ iterations, the original data vector (signal) $X = S^{(0)}$ is restored.

The modified integer DLGT spectrum of a two-dimensional digital image $X$ of size $N_1 \times N_2$ ($N_i = 2^{n_i}$, $n_i \in \mathrm{N}$, $i = 1, 2$) is obtained by repeated use of the one-dimensional modified integer DLGT, i.e. $N_2$ times along the first spatial axis and then $N_1$ times along the second spatial axis.

### 3.2. Fast evaluation of the modified DLGT spectra for image blocks

To show extended potential capabilities of the modified integer-to-integer DLGT (Section 3.1) and to proclaim viability of the locally progressive image coding idea, we here propose a novel fast procedure (algorithm) for evaluating the (modified) integer DLGT spectra of the selected blocks (ROI) of the digital image. The proposed algorithm refers to the assumption that the modified integer DLGT spectrum of the original image is known.

To begin with, we briefly survey the situation concerning the one-dimensional case. Let

$$Y = (s_0^{(n)}\, d_0^{(n)}\, d_0^{(n-1)}\, d_1^{(n-1)}\, d_0^{(n-2)}\, d_1^{(n-2)}\, d_2^{(n-2)}\, d_3^{(n-2)} \ldots \ldots\ d_0^{(1)}\, d_1^{(1)} \ldots d_{N/2-1}^{(1)})^T$$

be the modified integer DLGT spectrum of the data vector (signal) $X = (x_0\, x_1\, x_2\, \ldots\, x_{N-2}\, x_{N-1})^T$. Each wavelet coefficient $d_j^{(i)}$ ($i \in \{p, p+1, \ldots, n\}$, $j \in \{0, 1, \ldots, 2^{n-i} - 1\}$), evidently, can be put into one-to-one correspondence with the signal block $X_j^{(i)} = (x_{2^i \cdot j}\, x_{2^i \cdot j+1} \ldots x_{2^i (j+1)-1})^T$, i.e. numerical value of $d_j^{(i)}$ is specified uniquely by $X_j^{(i)}$ (full localization manifests itself on the $i$-th scale).

To find the (modified) integer DLGT spectrum $Y_j^{(i)}$ of $X_j^{(i)}$, the following two steps should be carried out:

1. The very first spectral coefficient (smoothed value $s_j^{(i)}$) in $Y_j^{(i)}$ is determined by:

$$s_j^{(i)} = s_0^{(n)} + \sum_{r=1}^{n-i} \left( \frac{1-(-1)^{j_{r-1}}}{2} \cdot d_{j_r}^{(i+r)} - \left\lfloor \frac{1}{2} d_{j_r}^{(i+r)} + \frac{1}{2} \right\rfloor \right);$$

   here: $j_0 = j$, $j_r = \lfloor j_{r-1}/2 \rfloor$, for all $r = 1, 2, \ldots, n-i$;

2. The rest spectral coefficients (differenced values) in $Y_j^{(i)}$ are extracted from the modified integer DLGT spectrum $Y$ of $X$, i.e. they are identified with the ordered set of wavelet coefficients

$$\{d_j^{(i)},\, d_{2j}^{(i-1)},\, d_{2j+1}^{(i-1)},\, d_{4j}^{(i-2)},\, d_{4j+1}^{(i-2)},\, d_{4j+2}^{(i-2)},\, d_{4j+3}^{(i-2)},\, d_{8j}^{(i-3)}, \ldots, d_{8j+7}^{(i-3)}, \ldots,\, d_{2^{i-1}\cdot j}^{(1)},\, d_{2^{i-1}\cdot j+1}^{(1)}, \ldots,\, d_{2^{i-1}(j+1)-1}^{(1)}\}.$$

Thus, the discrete (modified) integer DLGT spectrum $Y_j^{(i)}$ of the signal block (ROI) $X_j^{(i)}$ of size not less than $2^p$ ($p \in \{1, 2, \ldots, n-1\}$) is computed and takes the form:

$$Y_j^{(i)} = (s_j^{(i)}\, d_j^{(i)}\, d_{2j}^{(i-1)}\, d_{2j+1}^{(i-1)}\, d_{4j}^{(i-2)} \ldots \ldots d_{2^{i-1}\cdot j}^{(1)}\, d_{2^{i-1}\cdot j+1}^{(1)} \ldots d_{2^{i-1}(j+1)-1}^{(1)})^T.$$

Now, let $X = [X(m_1, m_2)]$ be a two-dimensional grey-level image of size $N \times N$ ($N = 2^n$, $n \in \mathrm{N}$) and let $Y = [Y(k_1, k_2)]$ be its two-dimensional modified (full decorrelation of wavelet coefficients across the scales not higher than $p$, $1 < p < n$) integer DLGT spectrum.

Consider a wavelet coefficient $Y(k_1, k_2)$, $k_1, k_2 \in \{1, 2, \ldots, 2^{n-p+1} - 1\}$. It is evident that indices $k_1$ and $k_2$ can be presented in the form: $k_1 = 2^{n-i_1} + j_1$, $k_2 = 2^{n-i_2} + j_2$, $i_1, i_2 \in \{1, 2, \ldots, n\}$ $j_1 \in \{0, 1, \ldots, 2^{n-i_1} - 1\}$, $j_2 \in \{0, 1, \ldots, 2^{n-i_2} - 1\}$.

One can easily ascertain that $Y(k_1, k_2)$ is associated with the image block (ROI) $X^{(k_1,k_2)}[X(\tilde{m}_1, \tilde{m}_2)]$, where $(\tilde{m}_1, \tilde{m}_2) \in V_{k_1} \times V_{k_2}$ and $V_{k_r} = \{j_r 2^{i_r}, j_r 2^{i_r} + 1, \ldots, (j_r + 1) 2^{i_r} - 1\}$, $r = 1, 2$. Let us denote the (modified) integer DLGT spectrum of the image block $X^{(k_1,k_2)}$ by $Y^{(k_1,k_2)} = [Y^{(k_1,k_2)}(u_1, u_2)]$, where $u_r \in \{0, 1, \ldots, 2^{i_r} - 1\}$, $r = 1, 2$.

A newly developed fast procedure (algorithm) for computing the two-dimensional (modified) integer DLGT spectrum $Y^{(k_1,k_2)}$ of the image block (ROI) $X^{(k_1,k_2)}$ is presented below.

**Algorithm.**

**1.** The following sets are formed:

$S_V = \{\alpha_0, \alpha_1, \ldots, \alpha_{n-i_1}\}$, where: $\alpha_0 = k_1$;

$\alpha_s = \lfloor \alpha_{s-1}/2 \rfloor$, $s = 1, 2, \ldots, n - i_1$;

$S_H = \{\beta_0, \beta_1, \ldots, \beta_{n-i_2}\}$, where: $\beta_0 = k_2$;

$\beta_t = \lfloor \beta_{t-1}/2 \rfloor$, $t = 1, 2, \ldots, n - i_2$;

$\gamma_1 = \{\gamma_1(0), \gamma_1(1), \ldots, \gamma_1(n - i_1 - 1)\}$, where:

$\gamma_1(0) = j_1$; $\gamma_1(s) = \lfloor \gamma_1(s-1)/2 \rfloor$, $s = 1, 2, \ldots, n - i_1 - 1$;

$\gamma_2 = \{\gamma_2(0), \gamma_2(1), \ldots, \gamma_2(n - i_2 - 1)\}$, where:

$\gamma_2(0) = j_2$, $\gamma_2(t) = \lfloor \gamma_2(t-1)/2 \rfloor$, $t = 1, 2, \ldots, n - i_2 - 1$;

$\Im_{k_r} = \{k_r\} \cup \left\{ \bigcup_{q=1}^{i_r - 1} \Im_{k_r}(q) \right\}$, where:

$\Im_{k_r}(q) = \{2^q k_r, 2^q k_r + 1, \ldots, 2^q (k_r + 1) - 1\}$, $r = 1, 2$.

**2.** An array of intermediate data $[\hat{A}^{(k_1,k_2)}(l_1, l_2)]$ ($l_1 \in \{0\} \cup S_V \cup \Im_{k_1}$, $l_2 \in \{0\} \cup \Im_{k_2}$) is found, namely:

$$\hat{A}^{(k_1,k_2)}(l_1, 0) = Y(l_1, 0) + \sum_{t=1}^{n-i_2} \left( \frac{1 - (-1)^{\gamma_2(t-1)}}{2} Y(l_1, \beta_t) - \left\lfloor \frac{1}{2} Y(l_1, \beta_t) + \frac{1}{2} \right\rfloor \right),$$

for all $l_1 \in \{0\} \cup S_V \cup \Im_{k_1}$;

$$\hat{A}^{(k_1,k_2)}(l_1, l_2) = Y(l_1, l_2),$$

for all $l_1 \in \{0\} \cup S_V \cup \Im_{k_1}$ and $l_2 \in \Im_{k_2}$.

**3.** The inverse (modified) integer DLGT is applied to the $l_1$-th ($l_1 \in \{0\} \cup S_V$) row of the data array $[\hat{A}^{(k_1,k_2)}(l_1, l_2)]$. The resulting vector is designated as $\left( A^{(k_1,k_2)}(l_1, 0) \ldots A^{(k_1,k_2)}(l_1, l_2) \ldots \right)$.

**4.** A vector $B^{(k_1,k_2)} = \left( B^{(k_1,k_2)}(0,0) \ldots B^{(k_1,k_2)}(0, l_2) \ldots \right)$, $l_2 \in \{0\} \cup \Im_{k_2}$, is formed, namely:

$$B^{(k_1,k_2)}(0, l_2) = A^{(k_1,k_2)}(0, l_2) + \sum_{s=1}^{n-i_1} \left( \frac{1 - (-1)^{\gamma_1(s-1)}}{2} A^{(k_1,k_2)}(\alpha_s, l_2) - \left\lfloor \frac{1}{2} A^{(k_1,k_2)}(\alpha_s, l_2) + \frac{1}{2} \right\rfloor \right).$$

**5.** The (modified) integer DLGT is applied to $B^{(k_1,k_2)}$. The resulting DLGT spectrum is $\left( \hat{B}^{(k_1,k_2)}(0,0) \ldots \hat{B}^{(k_1,k_2)}(0, l_2) \ldots \right)$.

**6.** Finally, the integer Le Gall wavelet coefficients of the image block $X^{(k_1,k_2)}$ are found in accordance with:

$Y^{(k_1,k_2)}(0,0) = \hat{B}^{(k_1,k_2)}(0,0)$;

$Y^{(k_1,k_2)}(u,0) = \hat{A}^{(k_1,k_2)}(l_1, 0)$, $u = 1, 2, \ldots, 2^{i_1} - 1$;

here $l_1$ is the $u$-th element of the ordered set $\Im_{k_1}$ (numbering of elements starts with 1);

$Y^{(k_1,k_2)}(0,v) = \hat{B}^{(k_1,k_2)}(0, l_2)$, $v = 1, 2, \ldots, 2^{i_2} - 1$;

here $l_2$ is the $v$-th element of the ordered set $\Im_{k_2}$ (numbering of elements starts with 1);

$Y^{(k_1,k_2)}(u,v) = \hat{A}^{(k_1,k_2)}(l_1, l_2)$, $u = 1, 2, \ldots, 2^{i_1} - 1$, $v = 1, 2, \ldots, 2^{i_2} - 1$; $l_1$ and $l_2$ are the $u$-th and the $v$-th elements of the ordered sets $\Im_{k_1}$ and $\Im_{k_2}$, respectively (numbering of elements in $\Im_{k_1}$ and $\Im_{k_2}$ starts with 1).

**7.** The end. The two-dimensional (modified) integer DLGT spectrum $Y^{(k_1,k_2)}$ of the image block (ROI) $X^{(k_1,k_2)}$ is computed.

Some comments on the efficiency of the newly developed algorithm for computing the (modified) integer DLGT spectra of the selected image blocks are presented in Section 4.

## 4. Experimental analysis and discussion

To motivate the proposed progressive image block encoding and transmitting idea, a couple of test images have been processed (Figure 2).

Computer simulation was performed on a PC with CPU Intel® Core i7 CPU@3.4GHz, 8 GB RAM, Windows7 x64, Programming language Java.

First and foremost, the efficiency of the newly developed algorithm for computing the (modified) integer DLGT spectra of the selected image blocks (Subsection 3.2), in comparison with direct evaluation of DLGT spectra for the same blocks (Lifting Scheme; Section 2), was analysed.

The achievable impressive speed gains, expressed in terms of $\rho = \tau_d / \tau_{pr}$ ($\tau_d$ specifies the time needed for direct evaluation of DLGT spectra for image blocks and $\tau_{pr}$ - that needed by the proposed procedure), are presented in Table 1.

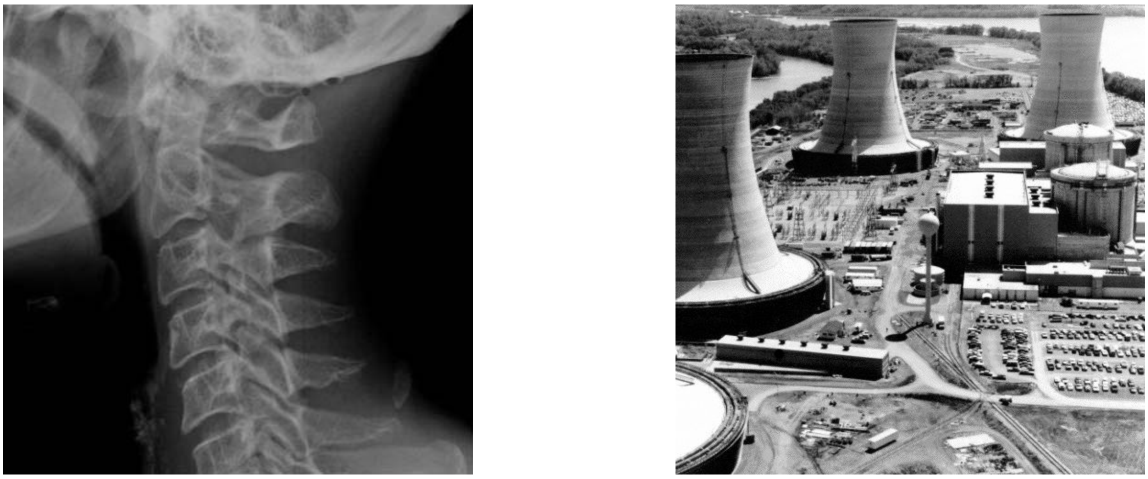
**Table 1.** Comparison of two approaches to finding DLGT spectra for ROI of size $2^p \times 2^p$

| $p$ / $N \times N$ | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|
| 512x512 | 165.6 | 334.5 | 526.1 | - | - |
| 1024x1024 | 154.9 | 313.3 | 505.9 | 678.5 | - |
| 2048x2048 | 148.9 | 302.8 | 489.6 | 648.2 | 629.0 |

Undoubtedly, fast passage from the modified integer DLGT spectrum of the image under processing to DLGT spectra of the selected image blocks (ROI) opens the door to many new real-time applications of the modified version of integer DLGT, say, image feature extraction in the Le Gall wavelet domain, locally progressive image encoding, localization of defects in textural images, etc. One interesting application of the modified DLGT to locally progressive encoding of regions of interest in a digital signal is described in [6].

In the context of progressive encoding and transmission of image blocks (ROI), the above results (achievable speed gains in computing the integer DLGT spectra for the ROI selected at the user's request) are highly valuable.

To simplify description of the obtained results, we here introduce the following notations: the CPU time required to perform the (modified) integer Le Gall wavelet transform (DLGT) of the digital image (or that of ROI) is denoted by $\tau_1$, to perform wavelet-based (SPIHT) encoding of the image (or that of ROI) – by $\tau_2$, to transmit data across a low communications channel – by $\tau_3$, to perform both the inverse SPIHT and the inverse integer DLGT, i.e. to reconstruct the image, – by $\tau_4$.



(a) (b)

**Figure 2**. Test grey-level images: (a) *Neck.bmp* $1024 \times 1024$; (b) *Power.bmp* $512 \times 512$

Three cases were analysed and compared: first of all, a non-compressed image was sent across the low communications channel to the user (Figure 3, a); secondly, lossless encoding (SPIHT algorithm with an improved quad-tree analysis scheme [9]) in the Le Gall wavelet domain was applied to the image, and compressed image was sent to the user (Figure 3, b); thirdly, the image was processed (lossy encoding by SPIHT), a "rough" estimate of the original image was sent to the user, the selected image blocks (ROI) were processed by applying both the (modified) integer DLGT and the SPIHT encoder, and then a high quality estimate of the selected ROI were sent again to the user's PC (Figure 3, c).

We have assumed that the capacity (data rate) of the low communications channel equals 100 KBps.

Obviously, transmitting of non-processed image *Neck.bmp* of size $1024 \times 1024$ (Figure 2, a) across the channel (Figure 3, a) requires $\tau = \tau_3 = 10.49$ s.

Computing the modified ($p = 7$) integer DLGT spectrum of the same image, lossless encoding (SPIHT algorithm; $T = 1$), sending across the channel and reconstruction (Figure 3, b) requires $\tau = \tau_1 + \tau_2 + \tau_3 + \tau_4 = 0.038 + 405.46 + 2.85 + 11.97 =$ $= 420.318$ s. Evidently, application of the SPIHT algorithm, with threshold value $T = 1$, is unacceptable (for the given capacity of the channel).

The developed approach (Figure 3, c) requires: as before, $\tau_1 = 0.038$ s for computing the modified integer DLGT spectrum of the image *Neck.bmp*; $\tau_2' = 0.041$ s for SPIHT encoding ($T = 32$; 0.5 KB of compressed data $Z$; $\beta = 2097$); $\tau_3' = 0.005$ s for transmitting of $Z$ to the user's PC across the channel, and $\tau_4' = 0.036$ s for reconstructing the "rough" estimate of the image (Figure 4, a).

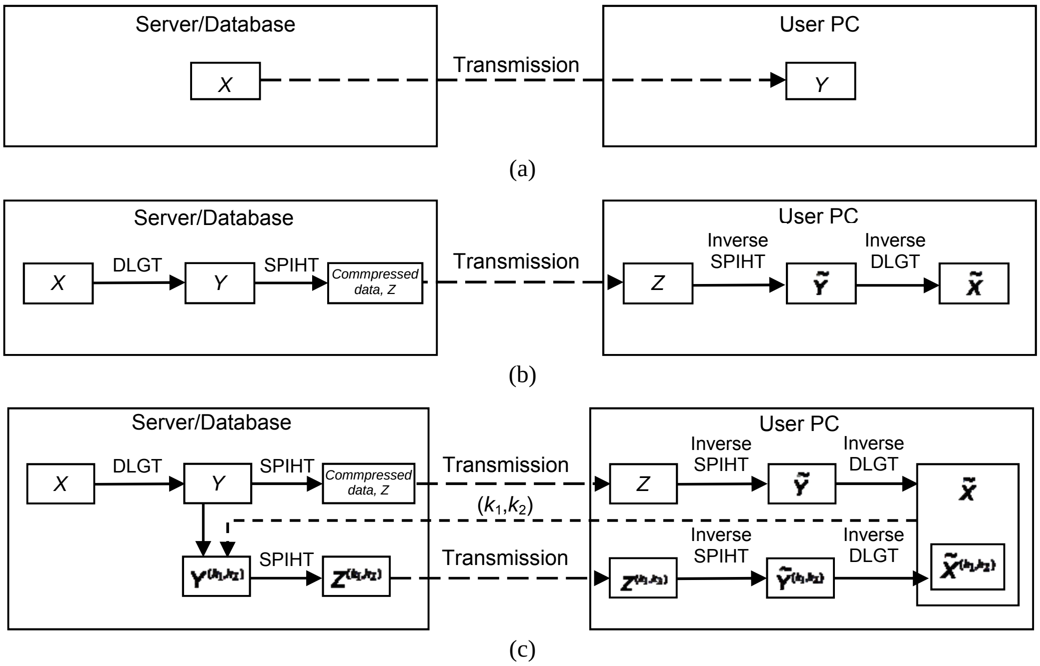



**Figure 3.** The digital image processing and transmission: (a) the image is sent across the low communications channel to the user; (b) lossless encoding (SPIHT) is applied to the image, whereupon compressed image is sent to the user; (c) the proposed locally progressive digital image encoding and transmitting approach

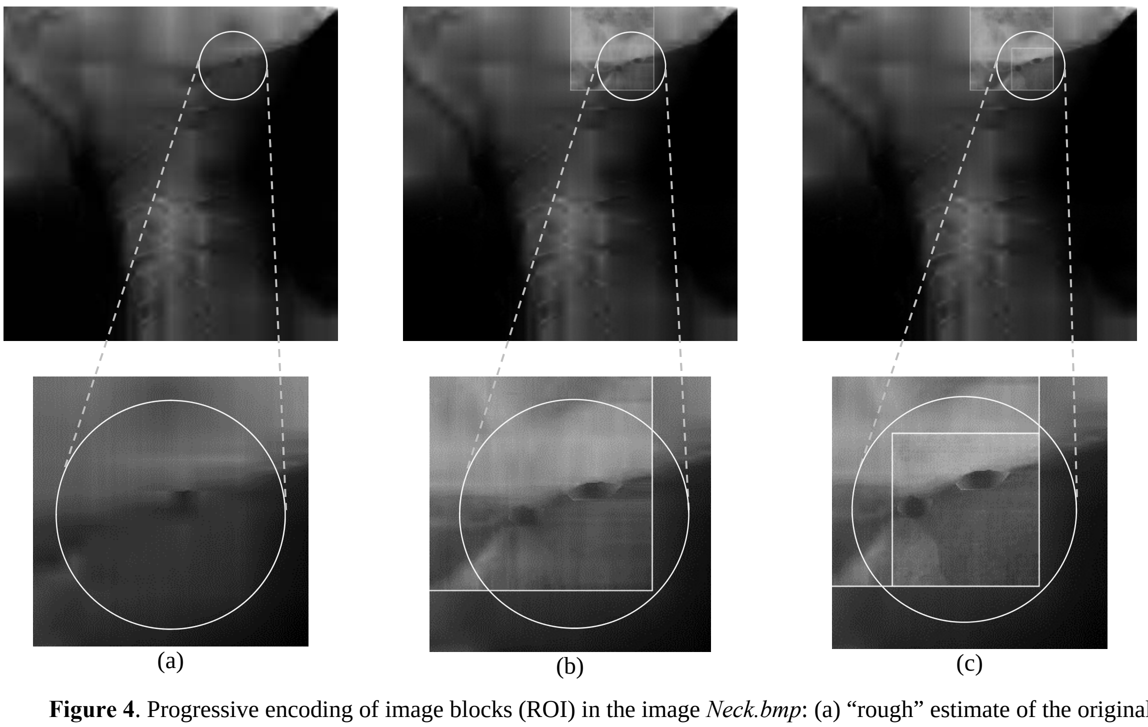



**Figure 4**. Progressive encoding of image blocks (ROI) in the image *Neck.bmp*: (a) "rough" estimate of the original image; (b) additional bits are added to the ROI $X^{(4,6)}$ ($T = 8$, 1.38 KB, $\tau' = 0.02$); (c) third quarter of the ROI $X^{(4,6)}$, i.e. image block $X^{(9,13)}$, in greater detail ($T = 2$, 3 KB, $\tau'' = 0.08$)

Consequently, the total image processing time equals $\tau' = \tau_1 + \tau_2' + \tau_3' + \tau_4' = 0.12$ s.

In Figure 4 (b, c), some additional blocks (ROI) of the image *Neck.bmp*, namely, $X^{(4,6)}$ and $X^{(9,13)}$, are processed and analysed. Compression details of the selected ROI are also indicated.

Thus, block-processing of the image *Neck.bmp*, in accordance with the third approach (Figure 3, c), takes (in total) $\tau + \tau' + \tau'' = 0.12 + 0.02 + 0.08 = 0.22$ s, and appears to be nearly 50 times faster than in the first approach (Figure 3, a).

The proposed locally progressive image encoding idea (Figure 3, c) has also been applied to visual inspection of several image blocks (ROI) in the image *Power.bmp* $512 \times 512$ (Figure 2, b). The obtained image processing speed gains are really hopeful: to generate a "rough" estimate (Figure 5, a; $T = 64$, 4.41 KB; $\beta = 59.44$ ) of the image requires $\tau = \tau_1 + \tau_2 + \tau_3 + \tau_4 = 0.007 + 0.27 + 0.044 + 0.012 =$ $= 0.333$ s; to prepare (for coherent analysis) the high quality image blocks $X^{(6,5)}$, $X^{(6,6)}$, $X^{(6,7)}$ and $X^{(7,7)}$ ($T = 2$; 42.14 KB), requires (in total) 1.488 s. So, the very first high quality image block (ROI) $X^{(6,5)}$ is made available to the user in 0.697 s, i.e. 3.76 times faster than in the first approach (Figure 3, a; sending of the original image *Power.bmp* to the user's PC requires 2.62 s). Processing and visual inspection of coherent image blocks (ROI) $X^{(6,6)}$, $X^{(6,7)}$ and $X^{(7,7)}$ runs on the real-time scale.

Since in each case (Figure 4, b, c; Figure 5, b) the user has the necessary ROI available, we come to the conclusion that the proposed Le Gall wavelet-based progressive image encoding and transmitting idea (Figure 3, c) may lead (especially, when applied to medical or space images characterized by large amounts of data) to auspicious and fast-track results.

(a)

(b)

**Figure 5**. Fast visual inspection of image blocks (ROI) in the image *Power.bmp* $512 \times 512$ : (a) "rough" estimate of the image; (b) "rough" estimate of the image with image blocks (ROI) $X^{(6,5)}$, $X^{(6,6)}$, $X^{(6,7)}$ and $X^{(7,7)}$ in greater detail

## 5. Conclusions

In the paper, a novel fast procedure for the computation of the (modified) discrete integer Le Gall wavelet (DLGT) spectra for the selected blocks (regions of interest - ROI) in the digital image is presented. The procedure employs improved space localization properties of a newly developed version (modification) of DLGT and refers to the fact that the modified DLGT spectrum of the original image is known.

It is shown that the developed procedure can be efficiently applied to implementing of a locally progressive digital image encoding and transmitting idea (approach). The essence of the approach – additional bit streams are used to add new details not to the whole image under processing but to the selected (at the request of the user!) blocks (ROI) in the image. The ultimate implementation of the approach is supported by the use of SPIHT encoder with an enhanced quad-tree analysis scheme.

Evidently, the developed approach (progressive encoding and transmission of image blocks) can also be put into practice using other zero-tree-based image compression algorithms, such as EZW (Embedded Zero-tree Wavelet), EBCOT (Embedded Block Coding with Optimized Truncation), etc. Which algorithm performs better? It goes without saying, the faster image encoding algorithm the better overall performance of the proposed approach.

Comparison of the developed locally progressive digital image coding technique with the new image compression standard JPEG 2000 [2], which also provides the possibility for defining ROI in an image, is scarcely pointful (both approaches have essential differences). In JPEG 2000, selected image blocks (ROI) are coded with better quality than the rest of the image, and this is done (just after the pre-processing step) by scaling up the wavelet coefficients so that the bits associated with the ROI are placed in higher bit planes, i.e. at the front of bit streams. Moreover, in JPEG 2000, the visual inspection of coherent ROI in the image (an interactive dialogue with the user) is limited.

We unreservedly believe that the proposed idea will find various applications in implementing efficient and up-to-date digital data processing technologies. In particular, the proposed locally progressive digital image coding idea can be employed when large amounts of requested graphical data are sent across a low communications channel (say, the Internet).

## References


[1] **I. Daubechies.** *Ten Lectures on Wavelets*, SIAM, Philadelphia, 1992. http://dx.doi.org/10.1137/1.9781611970104.

[2] **C. Christopoulos, A. Skodras, T. Ebrahimi.** The JPEG2000 Still Image Coding System, An Overview, *IEEE Trans. on Consumer Electronics*, *Vol.* 46, *No*. 4, 2000, 1103-1127. http://dx.doi.org/10.1109/30.920468.

[3] **M. Vetterli.** Wavelets, approximation and compression, *IEEE Signal Processing Mag*., *Vol*. 18, *No*. 5, 2001, 59-73. http://dx.doi.org/10.1109/79.952805.

[4] **R. Sudhakar, R. Karthiga, S. Jayaraman.** Image Compression Using Coding of Wavelet Coefficients: A Survey, *International Journal on Graphics, Vision and Image Processing*, *Vol*. 5, *No*. 6, 2005, 25-38.

[5] **L. Kaur, R. C. Chauhan, C. Saxena.** Wavelet based compression of medical ultrasound images using vector quantization. *Journal of Medical Engineering&Technology*, *Vol.* 30, *No*. 3, 2006, 128-133. http://dx.doi.org/10.1080/03091900500235705.

[6] **J. Valantinas, D. Kančelkis.** A new wavelet-based approach to progressive encoding of regions of interest in a digital signal. *Information Technology and Control*, *Vol*. 38, *No*. 3, 2009, 195–202.

[7] **J. M. Shapiro.** Embedded image coding using zero-trees of wavelet coefficients, *IEEE Trans. Signal Processing*, *Vol*. 41, 1993, 3445-3462. http://dx.doi.org/10.1109/78.258085.

[8] **A. Said, W. A. Pearlman.** A new, fast, and efficient image codec based on set partitioning in hierarchical trees, *IEEE Trans. Circuits Syst. Video Technology*, *Vol*. 6, 1996, 243-250. http://dx.doi.org/10.1109/76.499834.

[9] **J. Valantinas, D. Kančelkis.** Speeding-up image encoding times in the SPIHT algorithm. *Information Technology and Control*, *Vol*. 40, *No*. 1, 2011, 7–11. http://dx.doi.org/10.5755/j01.itc.40.1.187.

[10] **R. S. H. Istepanian, A. A. Petrosian.** Optimal Zonal Wavelet-Based ECG Data Compression for Mobile Telecardiology System, *IEEE Trans. Inform. Technol. Biomed.*, *Vol*. 4, 2000, 200-211. http://dx.doi.org/10.1109/4233.870030.

[11] **R. M. Rao, A. S. Bopadikar.** *Wavelet Transforms: Introduction to Theory and Applications*, Pearson Education, Asia, 2002.

[12] **H. F. Ates, M. T. Orchard**. An Adaptive Edge Model in the Wavelet Domain for Wavelet Image Coding, *Signal Processing: Image Comm.*, *Elsevier*, *Vol*. 20, *No*. 2, 2005, 169-185.

[13] **B. D. Vijendra, N. R. Alamelu.** Wavelet Based Medical Image Compression Using ROI EZW, *International Journal of Recent Trends in Engineering*, *Vol*. 1, *No*. 3, 2009, 97-100.

[14] **J. Valantinas, D. Kančelkis.** Improving compression time in zero-tree based image coding procedures, *Proceedings of the* 2*nd* *International Conference on Image Processing Theory Tools and Applications* (*IPTA* 2010), July 7-10, 2010, Paris, France, 118-121.

[15] **W. Sweldens.** The lifting scheme: A construction of second generation wavelets, *SIAM J Math Anal*, *Vol*. 29, *No*.2, 1997, 511-546. http://dx.doi.org/10.1137/S0036141095289051.

[16] **A. Jense, A. la Cour-Harbo.** Ripples in Mathematics: the Discrete Wavelet Transform, Springer, 2001.

[17] **C. Phang, P. Phang.** Modified Fast and Exact Algorithm for Fast Haar Transform, *International Journal of Computer Science and Engineering, Vol*. 2, *No*. 2, 2008, 55-58.


Received June 2011.