Compression and Contents Protection of Images, Texts and Graphics for Distance Learning Applications

ROUMEN KOUNTCHEV[1]; VLADIMIR TODOROV [2], ROUMIANA KOUNTCHEVA [3]
[1] Radiocommunications Dept., Technical University of Sofia,
8 Kl. Ohridski St., BG-1000 Sofia, BULGARIA, rkountch@tu-sofia.bg
[2] T&K Engineering, Mladost 3, POB 12, Sofia 1712, BULGARIA

Abstract: In the paper is presented a new method for efficient compression of still images, based on the Inverse Difference Pyramid (IDP) decomposition. A special version of the method is developed for the lossless compression of graphics and texts. The method is aimed at secure and flexible transfer of information via Internet. It permits the insertion of multiple digital watermarks in the image phase domain together with efficient data hiding. The method offers flexible approach for image contents protection and data hiding.

Key-words: Lossless and lossy image compression, Inverse Difference Pyramid decomposition, Layered image transfer, Digital watermarking, Complex Hadamard Transform.

1. Introduction

One of the most powerful communications tools nowadays is the Internet. It ensures fast transfer of information of any kind: images, texts, graphics, sound, etc. Together with this, the transferred information is easily accessible for too many people. Special requirements appear when medical data, containing confidential information is provided via Internet. In this case, the main goals are: to compress the information efficiently, so that to provide it as soon as possible and without visible quality deterioration and to protect this information from unauthorized access, editing or use. The widely used tools for image processing when Internet is concerned are JPEG and JPEG2000 [1, 2]. These famous standards very efficiently compress grayscale and color images, but they are not so good when texts and graphics are processed. Such information is usually processed with software, based on the JBIG and JPEG2000 (lossless version) standards [3]. They offer good compression, when bi-level images are processed, but they are not so efficient for grayscale text and graphic images. Additional difficulties appear when the processed information comprises pictures and texts together. The new method presented here, solves these problems to a high degree. This method is based on the Inverse Difference Pyramid (IDP) decomposition and performs efficient lossy and lossless image compression. The paper is arranged as follows. In Section 2 is presented the IDP method and its features, which permit the insertion of resistant and fragile watermarks in the compressed data; in Section 3 is presented the basic idea for lossless image compression; in section 4 is presented the algorithm for digital image watermarking with IDP, in section 5 are presented the experimental results, in Section 6, the conclusion, are pointed the method advantages, the application areas and its future development.

2. Inverse Difference Pyramid Decomposition

The IDP decomposition method [4] is presented with the following steps:

Step 1: The matrix [B(i,k)] of the digital halftone image is divided in K sub-images, [L(i,k)] each with size $2^n x 2^n$ elements, where n=3,4 or 5 depending on the original image size.

Step 2: The elements of the matrix $[L_{k_p}(i,k)]$, where $k_p=1,2,..,4^pK$, are defined for the level p=0,1,..,n-1 of the IDP pyramid of P levels, in correspondence with:

$$L_{k_p}(i,k) = \begin{cases} B_{k_0}(i,k) & \text{for} \quad p=0; \\ E_{k_{p-1}}(i,k) & \text{for} \quad p=1,2,...,n-1, \end{cases} \quad (1)$$

$k_p=1,2,..,4^pK$, $i, k = 0,1,2,.., 2^{n-p}-1$.

Here $B_{k_0}(i,k)$ is the (i,k) pixel from the sub-image with number k_0=1,2,..,K in the zero level (p=0) of the pyramid, which corresponds with the pixel B(i,k) from the original image; and $E_{k_{p-1}}(i,k)$ is respectively the pixel (i,k) from the difference image with number k_p in the pyramid level p, for p=1,2,…,n-1.

Step 3: The matrix of the sub-image $[L_{k_p}(i,k)]$ is transformed, using “truncated” two-dimensional orthogonal transform of some kind, like DCT, CHT (Complex Hadamard Transform), Haar, etc., selected in advance. The coefficients of the corresponding transform matrix are calculated in accordance with the expression:

$$s_{k_p}(u,v)=\begin{cases} s_{k_p}(u_r,v_r) & \text{for } m_p(u,v)=1;\\ 0 & \text{for } m_p(u,v)=0, \end{cases} \qquad (2)$$

$u,v = 0,1,..,2^{n-p}-1$ and $p = 0,1,..,n-1$,

where

$$s_{k_p}(u_r,v_r)=\frac{1}{4^{n-p}}\sum_{i=0}^{2^{n-p}-1}\sum_{k=0}^{2^{n-p}-1} L_{k_p}(i,k)\,t_p(i,k,u_r,v_r)$$

for $r = 1,2,...,R_p$; $m_p(u,v)$ are the elements of the binary matrix-mask $[M_p]$ with size $2^{n-p}x2^{n-p}$ for the level p, which defines the position of the "retained" coefficients, $s_{k_p}(u_r,v_r)$;

$R_p=\sum_{i=0}^{2^{n-p}-1}\sum_{k=0}^{2^{n-p}-1} m_p(i,k)$ - the number of the "retained" spectrum coefficients in the level p, selected in advance so that to be within the area defined by $1\le R_p<4^{n-p}$; $t_p(i,k,u_r,v_r)$ - the pixel (i,k) from the basic image (the kernel of the selected orthogonal transform) with spatial frequency (u_r,v_r) in the pyramid level p.

The elements $m_p(u,v)$ in (2) are defined in accordance with the following four operations:

- Calculation of the modules of the spectrum coefficients of every sub-image in the pyramid level p:

$$|s_{k_p}(u,v)|=\frac{1}{4^{n-p}}\left|\sum_{i=0}^{2^{n-p}-1}\sum_{k=0}^{2^{n-p}-1} L_{k_p}(i,j)\,t_p(i,k,u,v)\right|$$

for $u,v = 0,1, 2,..,2^{n-p}$.

- Calculation of the modules of the coefficients of the mean transform for the level p:

$$\overline{|s_p(u,v)|}=\frac{1}{4^p K}\sum_{k_p=1}^{4^p K}|s_{k_p}(u,v)|$$

- Arrangement of the "mean" modules in uniformly decreasing order:

$$\overline{|s_p(u_1,v_1)|}\ge\overline{|s_p(u_2,v_2)|}\ge\ldots\ge\overline{|s_p(u_{R_p},v_{R_p})|}; \qquad (3)$$

- Definition of the elements $m_p(u,v)$, performed in accordance with (3) as follows:

$$m_p(u,v)=\begin{cases} 1 & \text{for } u=u_r \text{ and } v=v_r \text{ for } r=1,2,..R_p;\\ 0 & - \quad \text{in all other cases.} \end{cases}$$

Step 4: The approximating model $\tilde{L}_{k_p}(i,k)$ for the sub-image k_p in the pyramid level p is defined, using the inverse orthogonal transform:

$$\tilde{L}_{k_0}(i,k)=\tilde{B}_{k_0}(i,k)=\sum_{u_r}\sum_{v_r} s_{k_0}(u,v)\,t_0^{in}(i,k,u,v)$$

for p=0 and $i,k =0,1,..,2^n-1$,

$$\tilde{L}_{k_p}(i,k)=\tilde{E}_{k_p}(i,k)=\sum_{u_r}\sum_{v_r} s_{k_p}(u,v)\,t_p^{in}(i,k,u,v)$$

for $p=1,..,P-1$ and $i,k =0,1,..,2^{n-p}-1$.

Here $t_p^{in}(i,k,u,v)$ is the kernel of the selected inverse orthogonal transform for the level p.

Step 5: The elements of the difference image k_p in the pyramid level p are defined:

$$E_{k_p}(i,k)=\begin{cases} B_{k_0}(i,k)-\tilde{B}_{k_0}(i,k) & \text{for } p=0;\\ L_{k_{p-1}}(i,k)-\tilde{L}_{k_{p-1}}(i,k) & \text{for } p=1,2,..,n-1. \end{cases}$$

when $i,j = 0,1,..,2^{n-p}-1$.

Step 6: The coefficients $s_{k_p}(u_r,v_r)$ from all sub-images in the pyramid level p are arranged as a two-dimensional massif R_p in accordance with their spatial frequency (u_r,v_r) for $r=1,2,..,R_p$.

Step 7: Each massif of same spectrum coefficients is converted into one-dimensional data sequence, using recursive Hilbert scan. All data sequences from same IDP level are arranged as one common data sequence.

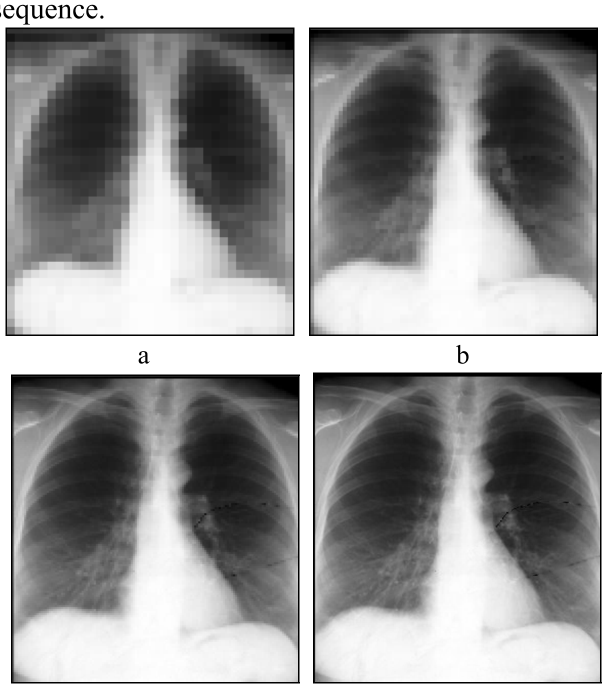


Fig.1. Presentation of the multi-level IDP decomposition for the test image "Thorax" (512x512 pixels, 8bpp).

The processing continues with lossless coding of the obtained data.

In Fig.1 are shown the images, obtained from 4 consecutive IDP levels for the test image "Thorax". This approach permits to stop the image decomposition in the level, for which the desired image quality is obtained [4]. For this image kind (grayscale medical images) the efficiency of the IDP compression is similar with that of the JPEG standard.

The software, based on the described method, was developed for grayscale (8bpp) and color (24 bpp) images. It could be implemented for images in any other format (for example12 bpp, grayscale).

For some applications the archived images should be without any change in their quality, i.e. they require lossless compression. For such applications the IDP decomposition consists of two levels only (the highest ones). In this case the lower pyramid level is with sub-image size 8x8 pixels, and the higher one – with size 4x4 pixels. In the higher level, should be calculated all the 16 transform coefficients.

The IDP decomposition offers the ability to insert resistant or fragile watermark in the compressed image data. The resistant watermark is inserted in the image spectrum domain. It is suitable for cases, when lossy compression is used and the watermark data is added in the compressed image data. The IDP decomposition permits the insertion of different watermark in every pyramid level. This watermark is resistant against widely used attacks as cropping, editing, etc [5]. The resistant watermark could be used for access control as well. For this is used a "hiding", visual watermark, which overlaps the original image and hides its contents. The watermark is removed with a password or another tool for access control. The fragile watermark is used, when image contents identification is needed. For this, the IDP method permits the watermark to be inserted as an additional level in the compressed data or as additional information in the amplitude of the last pyramid level. The watermark is very sensitive for any kind of image contents change and is used as a proof for unauthorized editing, resulting from usual pirates' attacks. The IDP decomposition permits the insertion of fragile and resistant watermarks together.

3. Lossless Data Compression with Histogram-Adaptive Run-Length Coding

The data sequence obtained in result of the IDP decomposition is processed with Histogram-Adaptive Run-Length Coding (HARLC) [6]. This part of the processing is based on the specific feature of the IDP decomposition, that the values of most of the coefficients in the higher level are "zero". The basic steps of the algorithm for lossless coding are as follows:

Step 1. The processing starts with the calculation of the histogram $h(\Delta S)$ of the coefficients values in the data sequence $\{\Delta S\}$;

Step 2. The histogram is analysed and are defined the parts, where $h(\Delta S)=0$.

Step 3. The sequences of same values (most frequently equal to zero) in the sequences $\{\Delta S\}$ are presented with some of the not-used histogram values in accordance with the following rules:

- Sequences of zeros, with length smaller or equal than that of the longest sequence of not used values in the histogram, is replaced by a number, equal to the sum of the start value of the sequence of not used values and the length of the compressed sequence;
- Sequences of zeros in the transformed data, longer than the sequence of not used values in the histogram, but shorter than 2^{mn}, are represented with 2m words. The first of these words contains the start value of the longest sequence of not used values, the next (m-1) words contain zeros, and the remaining words – the length of the coded sequence;
- Sequences of same values, different from zero, are represented in similar way, adding one more word, containing this value.
- Very long sequences (if there are any) are represented with additional word containing the sequence length;

Step 4. At the end of the processing the code words are coded with modified Huffman code.

In result of the processing the image data is compressed and the compressed images are converted in new format (tk).

This compression is very efficient for processing of graphic images and texts. It gives very good results for compression and storage of signals like ECG, etc.

Specific advantage of the IDP decomposition is that it permits to recognise text and pictures in compound images and to process them in most suitable way: the pictures - with multi-layer IDP, and texts and graphics – with two-level IDP. The recognition is based on the image contents analysis. For the areas, where texts and graphics were identified, is applied lossless compression and for the remaining part of the image – lossy compression with multi-layer IDP.

4. Image Watermarking with IDP Decomposition

4.1. Watermark insertion

The IDP decomposition permits the insertion of different watermark in every pyramid layer [5, 7]. The watermark insertion for one pyramid layer is described below with the following steps:

Step 1. The input image is divided in sub-blocks with size NxN elements, where $N=2^n$.

Step 2. Then are calculated the coefficients s(u,v) from the transform of each sub-block, using the two-dimensional Complex Hadamard Transform (CHT):

$$s(u,v)=\frac{1}{N^2}\sum_{i=0}^{N-1}\sum_{k=0}^{N-1}L(i,k)e^{-j\frac{\pi}{2}(ui+vk)}t(u,i)t(v,k)$$

for u,v = 0,1,..,N-1, where

$$j=\sqrt{-1},\quad t(p,q)=\begin{cases}1 & \text{for } n=2;\\ (-1)^{\sum_{r=3}^{n}\left\lfloor\frac{p}{2^{r-1}}\right\rfloor\left\lfloor\frac{q}{2^{r-1}}\right\rfloor} & \text{for } n=3,4,..,\end{cases}$$

p,q = 0,1,.., 2^n-1, n = $\lg_2$N.

Here t(p,q) is a sign function, and $\left\lfloor\frac{a}{b}\right\rfloor$ - operator, defining the integer part, obtained in result of the division of the numbers a and b.

Step 3. The modules and the phases of the obtained spectrum coefficients are calculated:

$$s(u,v)=M(u,v)e^{j\varphi(u,v)}=s_R(u,v)-js_I(u,v)$$

in correspondence with the relations

$$M(u,v)=\sqrt{[s_R(u,v)]^2+[s_I(u,v)]^2}\quad\text{and}$$

$$\varphi(u,v)=-\text{arctg}\left[\frac{s_I(u,v)}{s_R(u,v)}\right],$$

where

$$s_R(u,v)=\sum_{i=0}^{N-1}\sum_{k=0}^{N-1}L(i,k)t(u,i)t(v,k)\cos[\tfrac{\pi}{2}(ui+vk)]$$

$$s_I(u,v)=\sum_{i=0}^{N-1}\sum_{k=0}^{N-1}L(i,k)t(u,i)t(v,k)\sin[\tfrac{\pi}{2}(ui+vk)]$$

Step 4. Here are chosen the couples s(u,v) and s*(u,v) of complex-conjugated coefficients whose phases are inverse $\varphi(u,v)=-\varphi^*(u,v)$, and the modules answer the requirement:

$$|M(u,v)|=|M^*(u,v)|>\eta=\begin{cases}\alpha M_{max} & \text{if } \alpha M_{max}>\delta,\\ \delta & \text{- in other cases,}\end{cases}$$

Here η is a threshold, whose value is a part α<1 of the module M_{max} belonging to the largest coefficient (or a group of coefficients) in the transform of the processed sub-block. The minimum value for η is limited with the requirement it to be higher than some small positive constant, δ.

Step 5. One consecutive bit w_r of the digital watermark is inserted in the phases of the coefficients s(u,v) and s*(u,v), already chosen in step 3 in correspondence with the relation:

$$\varphi_w(u,v)=-\varphi_w^*(u,v)=\begin{cases}\varphi(u,v)+\Delta & \text{if } w_r(p)=1;\\ \varphi(u,v)-\Delta & \text{if } w_r(p)=0\end{cases}$$

for r = 1,2,..,R, where φ_w(u,v) and φ_w^*(u,v) are the phases of the watermarked coefficients s_w(u,v) and s_w^*(u,v), and R is the number of the binary elements w_r(p) of the digital watermark, p. The parameter Δ is an angle, which defines the "depth" of the watermark and together with α influences its "transparency" and resistance against attacks.

Step 6. The pixels L_w(i,k) of the watermarked sub-block are calculated, using two-dimensional Inverse Complex Hadamard Transform (ICHT) for the already obtained sub-block transform:

$$L_w(i,k)=\sum_{u=0}^{N-1}\sum_{v=0}^{N-1}s'(u,v)e^{j\frac{\pi}{2}(ui+vk)}t(u,i)t(v,k)$$

for i, k = 0,1,.., N-1, where

$$s'(u,v)=\begin{cases}M(u,v)e^{j\varphi_w(u,v)} & \text{if } M(u,v)\sin\varphi_w(u,v)\neq 0;\\ M(u,v)e^{j\varphi(u,v)} & \text{– in other cases.}\end{cases}$$

Here M(u,v), φ(u,v) and φ_w(u,v) are defined in correspondence with steps 2 and 5.

The sequence w_r is obtained in result of the XOR operation for every bit of the inserted watermark and corresponding bit, belonging to a pseudorandom sequence, which is a secret or public key used for the watermark encryption. The autocorrelation function of the sequence w_r should be similar with a delta-pulse. This requirement ensures high reliability, when the watermark is detected or extracted from the watermarked image.

4.2. Watermark extraction

Steps 1- 4 of the algorithm for watermark insertion are applied consecutively on the tested image aiming at the detection of all spectrum coefficients s(u,v), suitable for watermarking.

Step 5. The presence of an eventually inserted watermark p of D digits is checked. For this is calculated the coefficient $C_{m,p}$, which shows the correlation between the watermark in the pyramid level p and the watermark m, used for the watermarking of the spectrum coefficients s_w(u,v) for all sub-blocks, selected in **Step 4** of the watermark insertion. The cross-correlation coefficient is defined with the relation:

$$C_{m,p}=\sum_{r=1}^{R}[\varphi_r+\Delta_r(m)]\Delta_r(p)=A_p+B_{m,p}$$

for m, p = 1,2,..,D, where $[\varphi_r+\Delta_r(m)]=\varphi_{rw}(m)$ is the phase of the watermarked coefficient s_{rw}(u,v),

$$A_p=\sum_{r=1}^{R}\varphi_r\Delta_r(p),\quad B_{m,p}=\sum_{r=1}^{R}\Delta_r(m)\Delta_r(p)$$

φ_r – is the absolute value of the phase before the watermarking, and

$$\Delta_r(p)=(-1)^{w_r(p)}\Delta=\begin{cases}+\Delta & \text{if } w_r(p)=1;\\ -\Delta & \text{if } w_r(p)=0.\end{cases}$$

Here $w_r(p)$ are the binary elements from the pseudorandom sequence of R bits, which describes the inserted watermark for level p.

For big values of R and when $\varphi_r \approx$ const, from Step 5 of the watermark extraction procedure follows the relation $A_p = \varphi_0 \sum_{r=1}^{R} \Delta_r(p) \approx 0$. In case that the selected coefficients are not watermarked, $C_{0,p} \equiv A_p \approx 0$.

For the watermarked spectrum coefficients

$$C_{m,p} \approx \begin{cases} \sum_{r=1}^{R} [\Delta_r(m)]^2 = R\Delta^2 & \text{if } m = p; \\ \sum_{r=1}^{R} \Delta_r(m)\Delta_r(p) \approx 0 & \text{if } m \neq p. \end{cases}$$

Step 6. The decision for the detection of the watermark p_0 is taken in accordance with the rule:

$$\text{Watermark}_{p_o} = \begin{cases} \text{Yes} & \text{if} \quad (C_{m,p_0}/R\Delta^2) \geq \theta; \\ \text{No} & - \quad \text{in other cases,} \end{cases}$$

for m, p_0 = 1,2,..D, where θ is a pre-defined threshold, whose value is in the range $0 < \theta < 1$.

5. Experimental Results

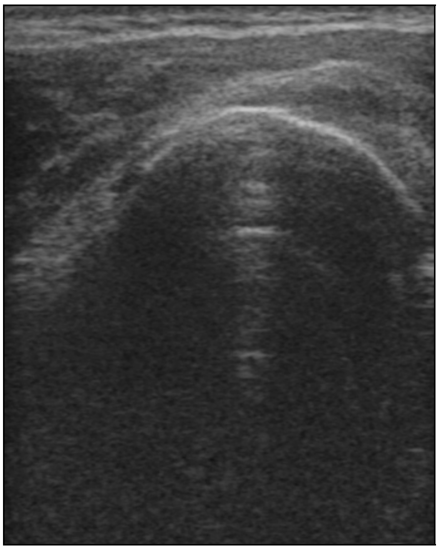
5.1. IDP Decomposition and compression

The software implementation (TKView) of the presented IDP decomposition and of the method for lossless data coding was developed in C++ and Visual C, Windows environment. The experiments were performed with large number of medical images: roentgen, ultrasound, mammographic, ECG, etc. After the processing the original bmp images were transformed in the new (tk) format.

Unlike JPEG and JPEG2000, the IDP method offers specific options for better presentation of some kinds of medical information, for example, ultrasound images. As it was pointed above, in the IDP levels are used different sets of coefficients. This approach permits to suppress certain image patterns, corresponding with the neglected spectrum coefficients.

In Fig.2 is presented the result of the IDP lossy compression of the ultrasound image “Axial”. The existing “speckle” noise is visually reduced with proper selection of the participating spectrum coefficients in the IDP level.

The ability to perform the filtration with the proper selection of the combinations of two-dimensional coefficients permits to suppress speckles or other similar noises of different size and shape. In the case when the original image contains bigger speckles, it is possible to use pyramid with three or more levels in order to increase the size of the filtered patterns and to suppress bigger forms. Usually this decreases the compression ratio significantly.



a. Original test image “Axial”

b. Test image “Axial” restored after compression 41 with speckle filtration

Fig.2. Test image “Axial” (344x430 pixels, 24 bpp)

The IDP decomposition is very efficient when lossless compression of graphic images is needed, for example for ECG and text images. Results for IDP (tk) and JPEG2000 (lossless) are given in Table 1. The last two columns contain the size of the compressed images.

Table 1

Image	CR TK	CR JPG2000	Size TK B	Size JPG2000 B
ECG	29,84	9,73	11 447	35 104
ECG1	16,55	6,99	17 727	41 988

Some of the test ECG images are presented in Fig.3 below.

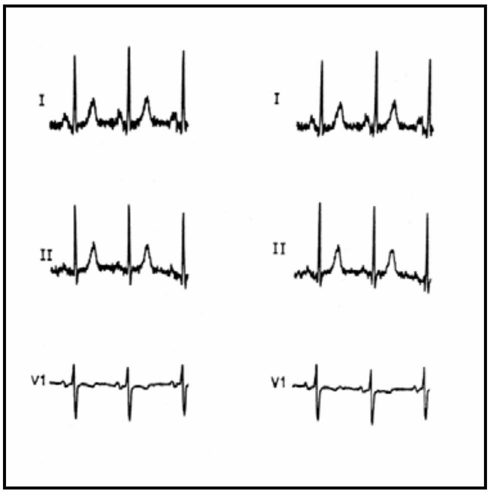
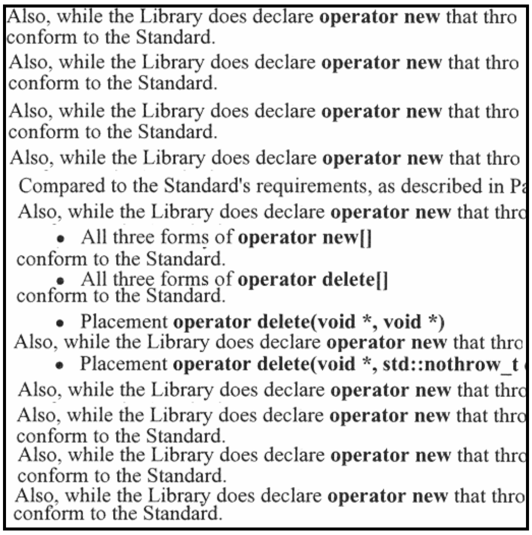


Fig.3. Test image ECG (581x588 pixels, 8bpp)

The method efficiency for processing and lossless compression of text images is illustrated with Fig.4. For this test image the compression ratio, obtained with lossless IDP compression (TKView) is 6. In Fig.4.a is presented the same image, processed with LuraTech Algovision (lossless JPEG2000). The distortions in the restored image are visible. The same product offers compression ratio 1,9 for this image when lossless compression is used.

Similar results were obtained for many kinds of graphic images as well (texts, signatures, contours, fingerprints, etc.).



Also, while the Library does declare **operator new** that thro
conform to the Standard.
Also, while the Library does declare **operator new** that thro
conform to the Standard.
Also, while the Library does declare **operator new** that thro
conform to the Standard.
Also, while the Library does declare **operator new** that thro
Compared to the Standard's requirements, as described in Pa
Also, while the Library does declare **operator new** that thro
- All three forms of **operator new[]**

conform to the Standard.
- All three forms of **operator delete[]**

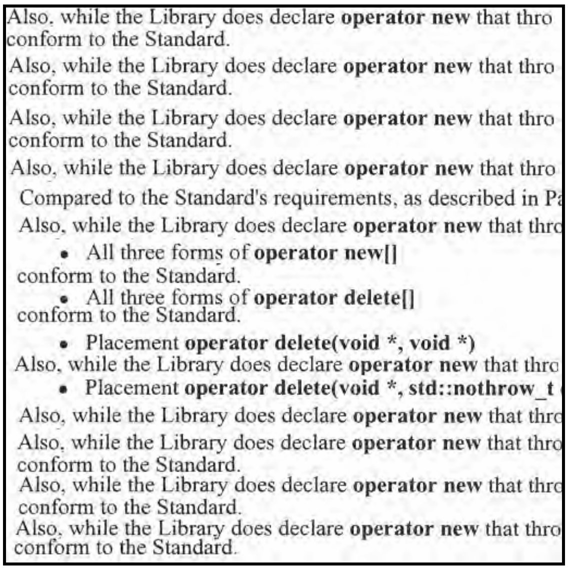
conform to the Standard.
- Placement **operator delete(void *, void *)**

Also, while the Library does declare **operator new** that thro
- Placement **operator delete(void *, std::nothrow_t**

Also, while the Library does declare **operator new** that thro
Also, while the Library does declare **operator new** that thro
conform to the Standard.
Also, while the Library does declare **operator new** that thro
conform to the Standard.
Also, while the Library does declare **operator new** that thro
conform to the Standard.

Fig.4.a. Original test image "Text"
(512x512 pixels, 8bpp)

The IDP decomposition permits to set regions of interest for some parts of the processed picture, which to be compressed with lossy or lossless version, based on image histogram analysis [6]. The ROIs should be framed by a square or rectangular form, which better corresponds with the IDP decomposition. Example test image is presented in Fig. 5 below.



Also, while the Library does declare **operator new** that thro
conform to the Standard.
Also, while the Library does declare **operator new** that thro
conform to the Standard.
Also, while the Library does declare **operator new** that thro
conform to the Standard.
Also, while the Library does declare **operator new** that thro
Compared to the Standard's requirements, as described in Pa
Also, while the Library does declare **operator new** that thro
- All three forms of **operator new[]**

conform to the Standard.
- All three forms of **operator delete[]**

conform to the Standard.
- Placement **operator delete(void *, void *)**

Also, while the Library does declare **operator new** that thro
- Placement **operator delete(void *, std::nothrow_t**

Also, while the Library does declare **operator new** that thro
Also, while the Library does declare **operator new** that thro
conform to the Standard.
Also, while the Library does declare **operator new** that thro
conform to the Standard.
Also, while the Library does declare **operator new** that thro
conform to the Standard.

Fig.4.b. Test image "Text", restored after compression with compression ratio 6 (JPEG2000).

CIRCUITS GRAND PUBLIC

La première "info appliance" sur

Le circuit Geode de NS intègre l'ensemble des fonctions de traitement du signal qui seront nécessaires aux systèmes communicants du futur. Première cible : des boîtiers-décodeurs multimédias à moins de 200$.

A peine sorti de l'aventure Cyrix, National Semiconductor n'aura pas mis longtemps avant de rediriger son savoir-faire en processeurs x86 sur une cible que l'Américain espère plus rentable et moins concurrentielle que le PC. NS ne jure plus désormais que par les "information appliances", ces systèmes numériques

Fig.5. Test image with 2 ROIs

The test image in Fig. 5 has 2 ROIs: the human face and the text. In this case the processing of the ROI, containing the human face will be performed with the lossy compression algorithm and the remaining part (the text) – with the lossless one.

5.2. Digital watermarking

A test image was watermarked using the suggested algorithm. In Fig 6 is shown the original grey-scale image of 256×256 pixels. A block size of 16×16 pixels was used for the implementation of the watermarking algorithm. The watermarked image where a total amount of 184 bits were embedded is shown in Fig. 7.

Fig.6. Original image "Lena"

Fig.7. Watermarked image at $\Delta = 12$

Table 2

Image name	Amount of embedded bits
Lena 256x256	184
Baboon 256x256	182
Camera 256x256	105
Peppers 256x256	179

Table 2 shows the amount of the embedded bits for four test images. The experiments proved that the watermark was embedded only in non-homogenous areas of the image. Despite of the inserted watermark no visible changes were noticed.

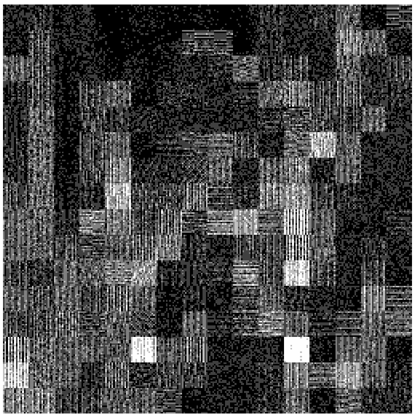


Fig.8. Absolute difference image at scale factor of 64

In Fig. 8 is shown the absolute difference between the original and the watermarked image "Lena" scaled by 64. As expected, the biggest difference occurs around edges. The watermarked image was compressed using a JPEG encoder.

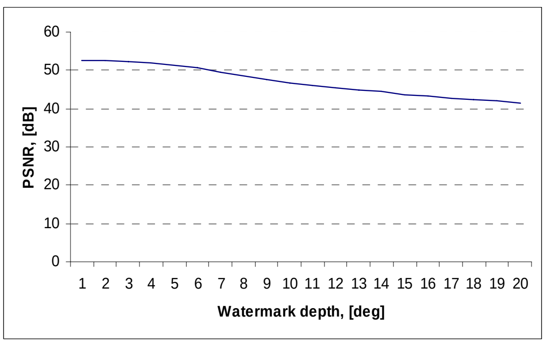


Fig.9. PSNR results at different Δ

Some of the experimental results are presented in Figures 9 and 10: Fig. 9 shows the resulting PSNR for different values of the parameter Δ; the bit error rate for different values of Δ and JPEG quality factors is shown in Fig. 10.

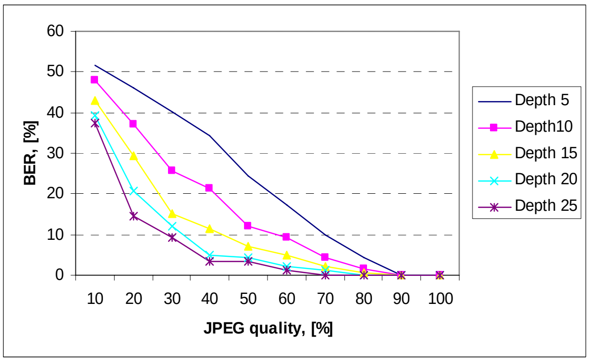


Fig. 10. BER versus JPEG quality

Examples for digital image watermarking based on the IDP decomposition are shown in Fig.11. In the figure is shown a text image with hiding watermark, which does not permit to understand the image contents.

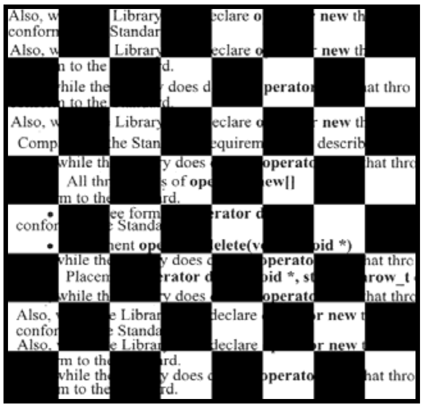


Fig.11. Watermarked test image

For some applications is used a "hiding" watermark, which should hide the most important part of the image (ROI) only. In result, the ROI should not be seen until the watermark is removed. Example is presented in Figures 12 (the watermark) and 13 (the watermarked test image).

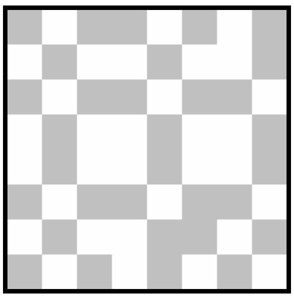


Fig.12. Example watermark image

The watermark in Fig.13 was inserted for example only – there is not real medical ROI in this place. For real applications the watermark should be darker, in order to hide the image contents better.

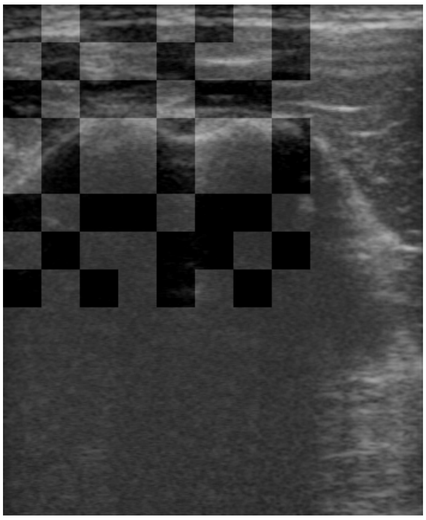


Fig.13. Ultrasound image with inserted hiding watermark

5. Conclusion

The ***main advantages*** of the IDP decomposition method are:

- The method offers efficient image compression, suitable for creation of large data bases of patients' medical information;
- The method permits the recognition of texts and pictures in compound images and their compression with most efficient algorithms, setting ROIs in the processed image;
- The method permits the processed images to be transferred layer by layer, with increasing quality. In result, the image transfer could be stopped any time, when required image quality is obtained;
- The method offers wide abilities for image filtration and noise suppression in ultrasound images;
- The method permits the insertion of fragile and resistant watermarks in the processed visual information, on the basis of which to be developed new tools for data access management;
- The method provides new tool for lossless compression of graphic images (ECG, etc.);
- The method offers efficient compression of biometric information and hiding this information as watermark, in order to use it for some kind of secure access control.

The method will be ***further developed*** in following directions:

- To permit the setting of flexible regions of interest (ROI) in the image, which to be processed with IDP multi-level decomposition or with lossless HARL compression;
- To perform automatic detection of texts and graphics in compound images and setting the corresponding ROIs;
- To develop the ability to process images stored in different format;
- To permit to compress the image in accordance with the required quality of the restored image and to stop automatically the IDP decomposition when this quality is obtained.

Acknowledgement:
This paper was supported by the Bulgarian Ministry of Education and Science (Contract № I 1304).

References

[1] T. Acharya, P. Tsai. *JPEG 2000 standard for image compression: concepts, algorithms and VLSI architectures.* J. Wiley & sons. NJ, 2005.

[2] M. Antonini, M. Barlaud, P. Mathieu, I. Daubechies. Image Coding using Wavelet Transform. *IEEE Trans. on Image Processing*, Vol. 1, No.2, April 1992, pp. 205-220.

[3] Y. Shi, H. Sun. *Image and Video Compression for Multimedia Engineering.* CRC Press, 2000.

[4] R. Kountchev, M, Milanova, C. Ford, R. Kountcheva. *Multi-layer Image Transmission with Inverse Pyramidal Decomposition*. In "Computational Intelligence for Modelling and Predictions", S. Halgamuge, L. Wang (Eds.), Vol. 2, Chapter No 13, pp. 179–196, Springer-Verlag Berlin Heidelberg, 2005.

[5] R. Kountchev, M. Milanova, C. Ford, S. Rubin. *Multimedia Watermarking with Complex Hadamard Transform in the Inverse Pyramid Decomposition*. Proc. of the 2003 IEEE Intern. Conference on Information Reuse and Integration (IRI'03), Las Vegas, NV, USA, October 27-29, 2003, pp. 305-310.

[6] R. Kountchev, M. Milanova, C. Ford, Vl. Todorov, R. Kountcheva. *Detection and Lossless Compression of Texts and Graphics in Compound Images.* International Symposium on Innovations in Intelligent Sysems and Applications (INISTA, 2005), June 15-18 2005, Istanbul, Turkey, pp.1-4.

[7] R. Kountchev, V. Mirchev. Digital Watermarking Using Complex Hadamard Transform and Phase Modulation. Intern. Scientific Conf. on Information, Communication and Energy Systems and Technologies (ICEST'05), Nis, Serbia and Montenegro, Proc. Vol. 1, June 2005, pp. 331-333.