Image Compression with Recursive IDP Algorithm

Roumen Kountchev[1]

Abstract – **A recursive algorithm for compression of halftone and color images, in correspondence with the method for Inverse Difference Pyramid decomposition (IDP), based on two-dimensional orthogonal transforms, is described in the paper. The basic steps of the algorithm for coding and decoding are defined, and the corresponding block diagram is synthesized. Some results of the model of the method behavior and comparison with the standard JPEG are given also.**

Keywords – **image compression, inverse difference pyramid decomposition, truncated orthogonal transform.**

I. Introduction

The basic requirements towards image compression algorithms are the big compression ratio, the high quality of the restored image and the low computational complexity. The algorithm for still images compression, based on the Inverse Pyramid Decomposition (IDP) method [1], presented here, suits these requirements to a high degree. The method efficiency is evaluated comparing it with the wavelet decomposition using a bank of 3/5-tap digital filters, used in the JPEG 2000 standard [2]. Possible applications of the offered algorithm are evaluated on the basis of the comparison results.

II. IDP Coding

The algorithm for recursive IDP coding of halftone digital images with orthogonal transforms is presented with the block diagram in Fig. 1. The orthogonal transform, used here, is the two-dimensional Discrete Cosine Transform (DCT) [1]. The coding of the original digital image is done performing the following steps:

Step 1: The Matrix $[B(i,j)]$ of the digital halftone image is divided in K sub-images, $[L(i,j)]$ each with size $2^n \times 2^n$ elements, where n is in the interval $n = 3,4,5$.

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

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

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

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

Step 3: The matrix of the sub-image $[L_{k_p}(i,j)]$ is transformed, using the so called "truncated" two-dimensional orthogonal transform of the kind, selected in advance, like DCT, WHT, Haar, etc. The coefficients of the corresponding matrix-transform are defined with the expression:

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

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

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

for r=1,2,...,R_p; $m_p(u,v)$ are the elements of the binary matrix-mask $[M_p]$ with size $2^{n-p} \times 2^{n-p}$ for the level p, which defines the position of the "retained" coefficients $s_{k_p}(u_r,v_r)$; $R_p = \sum_{i=0}^{2^{n-p}-1} \sum_{j=0}^{2^{n-p}-1} m_p(i,j)$ – the number of the "retained" spectrum coefficients in the level p, selected in advance in the interval $1 \leqslant R_p < 4^{n-p}$; $t_p(i,j,u_r,v_r)$ - the pixel (i,j) from the basic image (the kernel of the selected orthogonal transform) with spatial frequency (u_r,v_r) in the level p.

The elements $m_p(u,v)$ in the Eq. (2) are defined with the following four steps:

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

$$|s_{k_p}(u,v)| = \frac{1}{4^{n-p}} \left| \sum_{i=0}^{2^{n-p}-1} \sum_{j=0}^{2^{n-p}-1} L_{k_p}(i,j) t_p(i,j,u,v) \right| \quad (3)$$

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

3.2. Calculation of the modules of the coefficients of the mean transform for the level p:

$$|\bar{s}_{k_p}(u,v)| = \frac{1}{4^pK} \sum_{k_p=1}^{4^pK} |s_{k_p}(u,v)| \quad (4)$$

3.3. Arrangement of the "mean" modules in uniformly decreasing order:

$$|\bar{s}(u_1,v_1)| \geqslant |\bar{s}(u_2,v_2)| \geqslant ... \geqslant |\bar{s}(u_{R_p},v_{R_p})|; \quad (5)$$

3.4. Definition of the elements m_p(u,v) in accordance with Eq. (5) as follows:

$$m_p(u,v) = \begin{cases} 1 \text{ for } u{=}u_r \text{ and } v{=}v_r \text{ for } r{=}1,2,..,R_p; \\ 0 - \text{in all other cases.} \end{cases} \quad (6)$$

Step 4: The quantizated value of every retained spectrum coefficient is defined:

[1]Roumen K. Kountchev is with the Faculty of Communications and Communications technologies, Technical University of Sofia, Kliment Ohridsky 8, 1000 Sofia, Bulgaria; E-mail: rkountch@vmei.acad.bg

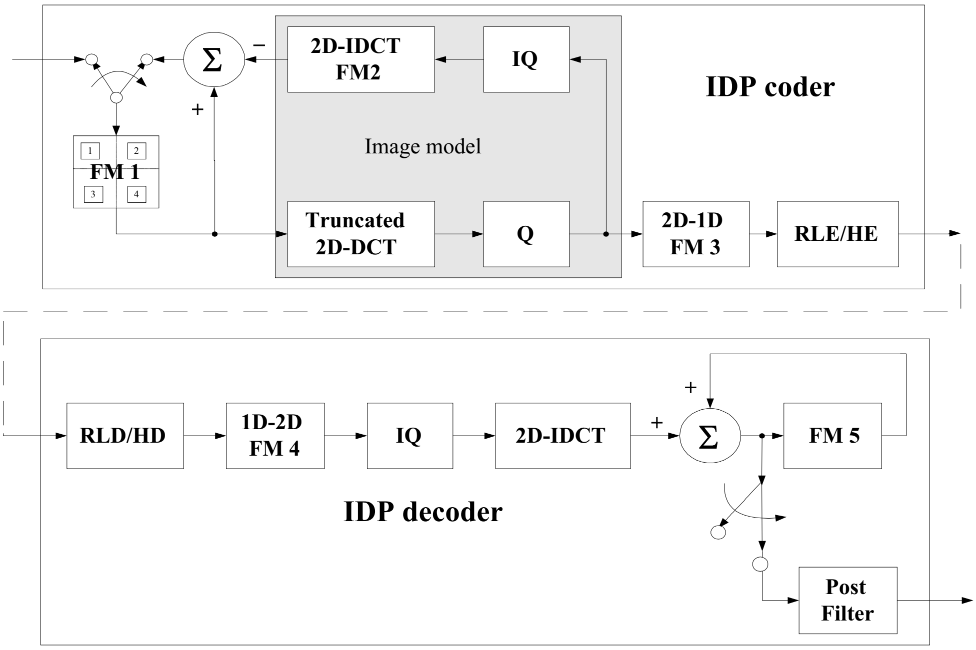


Fig. 1. Block-diagram of recursive IDP coder

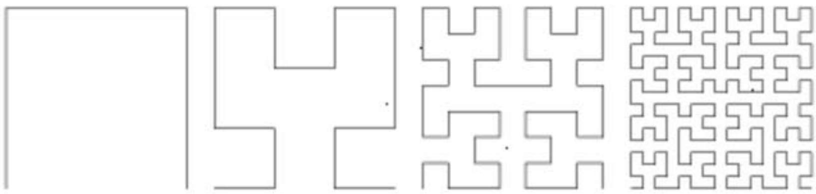


Fig. 2. Recursive Hilbert scan for $n = 2, 3, 4$

$$s^q_{k_p}(u_r, v_r) = [s_{k_p}(u_r, v_r) / \Delta_p(u_r, v_r)]_{\text{integer}} \quad (7)$$

for r=1,2,.., R_p, where $\Delta_p(u_r, v_r)$ is an element of the quantizating matrix $[Q_p]$ for the level p, selected in advance on the basis of experimental research on the influence of the quantization error on the restored image quality, $[_*]_{integer}$ – operator, defining the integer part of the number in the brackets.

Step 5: The dequantizated value of every quantizated spectrum coefficient is defined:

$$s'_{k_p}(u_r, v_r) = s_{k_p}(u_r, v_r).\Delta_p(u_r, v_r) \quad (8)$$

Steps 6: The approximating model $\tilde{L}_{k_p}(i,j)$ for the sub-image k_p in pyramid level p is defined, using the inverse orthogonal transform:

$$\tilde{L}_{k_0}(i,j) = \tilde{B}_{k_0}(i,j) = \sum_{u_r}\sum_{v_r} s'_{k_0}(u,v) t_0^{\text{in}}(i,j,u,v) \quad (9)$$

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

$$\tilde{L}_{k_p}(i,j) = \tilde{E}_{k_p}(i,j) = \sum_{u_r}\sum_{v_r} s'_{k_p}(u,v) t_p^{\text{in}}(i,\, j,\, u, v) \quad (10)$$

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

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

Step 7: The elements of the difference image k_p in the pyramid level p are defined:

$$E_{k_p}(i,j) = \begin{cases} B_{k_0}(i,j) - \tilde{B}_{k_0}(i,j) \text{ for } p = 0; \\ L_{k_{p-1}}(i,j) - \tilde{L}_{k_{p-1}}(i,j) \text{ for } p = 1, 2, .., P-1. \end{cases} \quad (11)$$

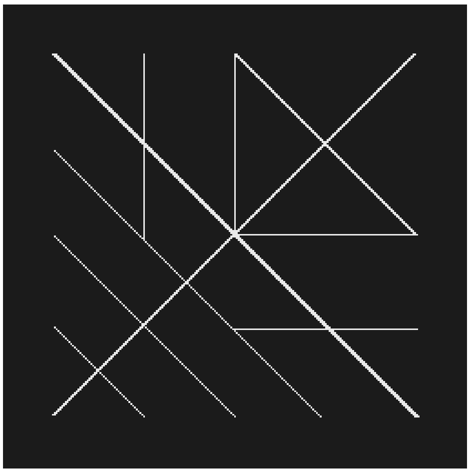
when $i, j = 0, 1, .., 2^{n-p} - 1$.

Step 8: The coefficients $s^q_{k_p}(u_r, v_r)$ from all sub-images in the pyramid level p are arranged in R_p two-dimensional massifs in accordance with their spatial frequency (u_r, v_r) for r=1,2,..,R_p.

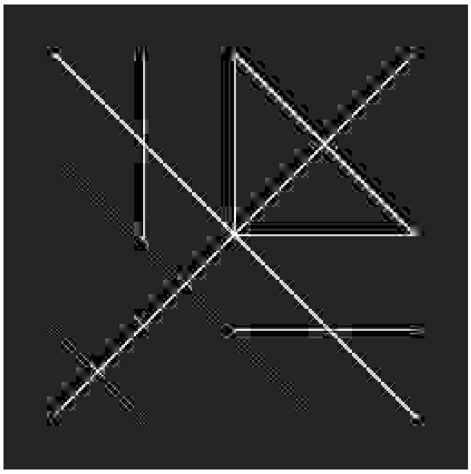
Step 9: Every two-dimensional massif of spectrum coefficients is converted into one-dimensional, using recursive Hilbert scan [3], shown in Fig. 2. Thus obtained one-dimensional massifs from every IDP level are arranged as one common sequence. At the beginning of this sequence is inserted the header, which contains information about the elements of the mask $[M_p]$, the number of the selected matrix $[Q_p]$, the values of R_p and , the kind of the orthogonal transform for every pyramid level, etc.

10.1. Adaptive coding of the lengths of the series of equal symbols (RLE);

10.2. Adaptive coding with modified Huffman code (HE). In result is obtained a compressed data file of the image data, which could be transferred via the communications channel or saved in a PC memory, depending on the IDP algorithm.

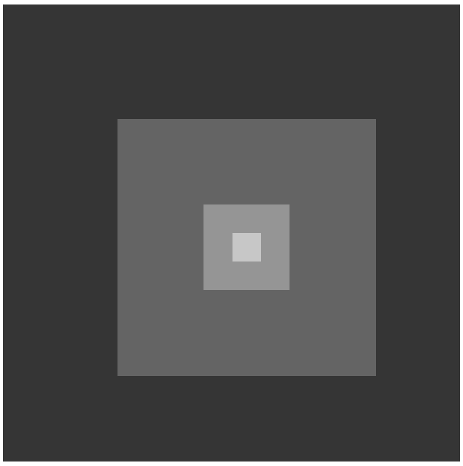


IDP compression 38,8:1 with PSNR = f

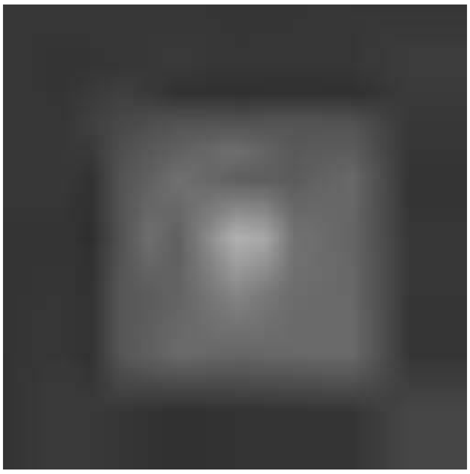


JPEG2000 compression 38,5:1

Fig. 3. Test image “Crosses”, 256x256, 8 bpp

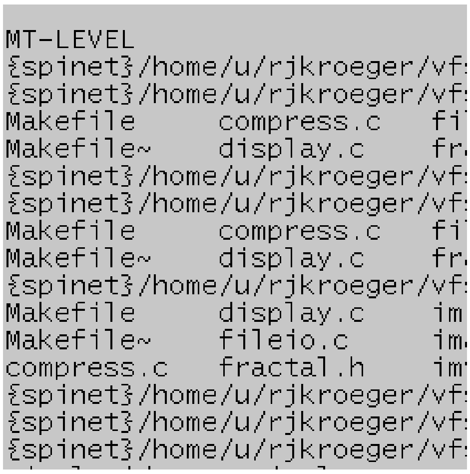


IDPcompression 468:1 with PSNR = f

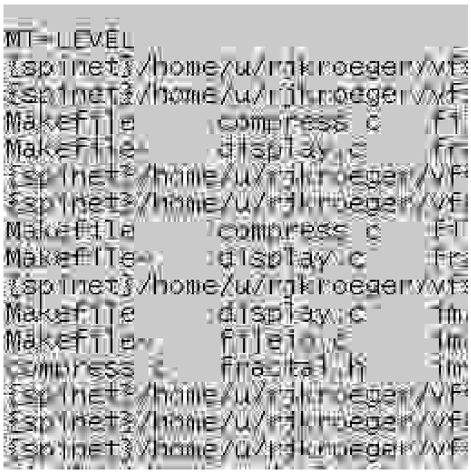


JPEG2000 compression 468:1

Fig. 4. Test image “Squares”, 256x256, 8 bpp



IDP compression 13.2:1 with PSNR = f



JPEG2000compression 13,2:1

Fig. 5. Test image “Text”, 256x256, 8 bpp

III. IDP Decoding

The decoding of the compressed image data in accordance with the general recursive IDP algorithm is performed following the steps below (Fig. 1):

Step 1: The Huffman codes and the lengths of the series of equal symbols are decoded (RLD+HD decoding);

Step 2: The coefficients $s^q_{k_p}(u_r, v_r)$ are dequantizated, in accordance with Eq. (8);

Step 3: The approximation model of the sub-image $\tilde{L}_{k_p}(i,j)$ is calculated using the inverse orthogonal transform, Eqs. (9)-(10);

Step 4: The elements $B'(i,j)$ of the restored image are calculated:

$$B'_k(i,j) = \tilde{B}_{k_0}(i,j) + \sum_{p=1}^{P-1} \tilde{E}_{k_{p-1}}(i,j) \tag{12}$$

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

Step 5: The decoded image is post-filtrated adaptively meaning the values of the pixels from the both sides of the borders of the sub-images $[L(i,j)]$ with size $2^n \times 2^n$. The border pixels are positioned in bands (respectively rows and columns) with width equal to two pixels.

IV. Coding of Color Images

The coding of color images (written in format 4:4:4), based on the described algorithm was performed, applying it on the matrix of every primary color component: R,G,B. In order to obtain higher compression ratio, the R,G,B components of every pixel (i,j) were transformed in Y,Cr,Cb and the 4:4:4 format was converted into 4:2:0, in correspondence with the ITU Recommendation 601-R:

$$\begin{bmatrix} Y(i,j) \\ Cb(i,j) \\ Cr(i,j) \end{bmatrix} = \begin{bmatrix} 0,2990 & 0,5870 & 0,1140 \\ -0,1687 & -0,3313 & 0,5000 \\ 0,5000 & -0,4187 & -0,0813 \end{bmatrix} \begin{bmatrix} R(i,j) \\ G(i,j) \\ B(i,j) \end{bmatrix}$$

$$\begin{bmatrix} R(i,j) \\ G(i,j) \\ B(i,j) \end{bmatrix} = \begin{bmatrix} 1,0000 & 0,0000 & 1,4020 \\ 1,0000 & -0,3441 & -0,7141 \\ 1,0000 & 1,7720 & 0,0000 \end{bmatrix} \begin{bmatrix} Y(i,j) \\ Cb(i,j) \\ Cr(i,j) \end{bmatrix} \tag{13}$$

When the image format is changed from 4:4:4 into 4:2:0, the horizontal and the vertical size of the matrices $[Cb]$ and $[Cr]$ becomes two times smaller in respect to the matrix $[Y]$. This is carried out, using the mean value of every four neighboring pixels in $[Cb]$ and $[Cr]$ in correspondence with the relations:

$$\begin{aligned} Cr_0(i,j) &= \frac{1}{4}[Cr(i,j-1)+Cr(i+1,j-1) \\ &\quad +Cr(i,j)+Cr(i+1,j)] \\ Cb_0(i,j) &= \frac{1}{4}[Cb(i,j-1)+Cb(i+1,j-1) \\ &\quad +Cb(i,j)+Cb(i+1,j)], \end{aligned} \tag{14}$$

where $Cr_0(i,j)$ and $Cb_0(i,j)$ are respectively the elements of the matrices $[Cr_0]$ and $[Cb_0]$, whose size is two times smaller than that of $[Cb]$ and $[Cr]$.

Every one of the components Y, Cr_0, Cb_0 is processed, applying the already described general algorithm IDP for coding of halftone images.

The decoding of the compressed color images is performed with the following two operations:

- Double increasing the size of the recovered (decompressed) matrices $[\hat{C}r_0]$ and $[\hat{C}b_0]$, using 2D zero-order interpolation, for the calculation of matrices $[\hat{C}r]$ and $[\hat{C}b]$;
- Inverse transform in correspondence with Eq. (13) for the components $\hat{Y}$, $\hat{C}r$, $\hat{C}b$ of every pixel, for the calculation of the corresponding primary color components, $\hat{R}, \hat{G}, \hat{B}$.

V. Results of the Modelling

Some results of the modeling of the described IDP algorithm and from its comparison with the JPEG 2000 standard are illustrated with the images shown in Figs. 3-5 and the data in Table 1 [1]. The results show that the IDP algorithm has some advantages over JPEG 2000 in the lower computational complexity and gives higher image quality for same compression ratio for text images and drawings. For texture images the algorithm, used in the JPEG 2000 standard, ensures higher compression ratio.

Table 1. Comparison of the computational complexity of decompositions IDP and WP (JPEG 2000)

	Operation			
	Sums per pixel in	Sums per pixel in	Multiplies per pixel in	Multiplies per pixel in
Decomposition	the coder	the decoder	the coder	the decoder
IDP	14,31	7,15	2,18	1,09
WP (JPEG 2000)	31,5	36,75	21	21

VI. Conclusion

A new image compression algorithm based on the IDP decomposition with 2D orthogonal transform, and suitable for processing of halftone and color images has been developed and presented in this work. It was firmly demonstrated, that the algorithm should be preferred in the following two applications:

- For compression of graphics and text images;
- For real-time intra-frame compression of TV images.

References

[1] R. Kountchev, V. Haese-Coat, J. Ronsin. Inverse Pyramidal Decomposition with multiple DCT. Signal Processing: Image Communication, Elsevier, Vol. 17, February 2002, pp. 201-218.

[2] M. Rabbani, R. Joshi. An Overview of the JPEG 2000 Still Image Compression Standard. Signal Processing: Image Communication, Elsevier, Vol. 17, Jan. 2002, pp. 3-48.

[3] J. Quinqueton, M. Berthod. A locally adaptive Peano scanning algorithm.IEEE Trans. Pattern Analysis and Machine Intelligence, No 3, 1981, pp. 403-412.