ISSN 1392 – 124X, ISSN 2335 – 884X (online) INFORMATION TECHNOLOGY AND CONTROL, 2013, Vol. 42, No. 4

Region-of-Interest Coding based on Fovea and Hierarchical Trees

J. C. Galan-Hernandez, V. Alarcon-Aquino, J. M. Ramirez-Cortes[1], O. Starostenko

Department of Computing, Electronics, and Mechatronics
Universidad de las Americas Puebla
Sta. Catarina Martir, Cholula, Puebla. C.P. 72810. Mexico
e-mail: juan.galanhz@udlap.mx, vicente.alarcon@udlap.mx

[1] Department of Electronics
Instituto Nacional de Astrofisica, Optica y Electronica
Tonantzintla, Puebla Mexico

crossref http://dx.doi.org/10.5755/j01.itc.42.4.3076

Abstract. Image and video compression exploits the redundancy of data to create a smaller representation. Lossy compression can be considered to be a type of transform coding where the raw data is transformed to a domain. Such a transform coding stores most of image energy into very few coefficients. In this paper we propose a compression algorithm based on Set Partitioning In Hierarchical Trees (SPIHT) that exploits the Human Visual System (HVS) and its fovea. In order to increase the image quality of the reconstructed image, regions of interest (ROI) are defined around a given point of gaze. The use of a fovea combined with ROI for image compression can help to improve the quality of the perception of the image and preserve different levels of detail around the ROI. In the proposed approach, the image is compressed using the Lifting Wavelet Transform and then quantized at multiple compression ratios around the point of fixation of an observer, taking advantage of the natural aliasing of the HVS around the fovea. Such a compression delivers better image or frame reconstruction when a fixation point of an observer is given.

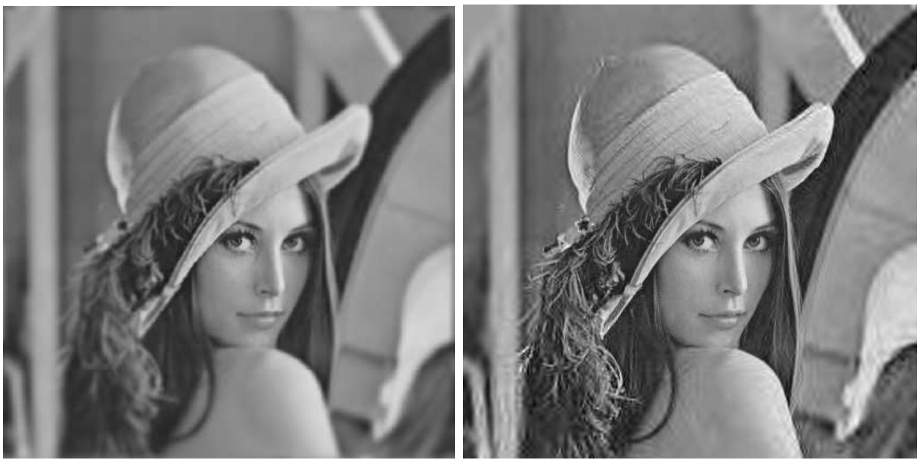
Keywords: Compression techniques; wavelet transforms; regions-of-interest; fovea; hierarchical trees.

1. Introduction

Compression is nowadays widely used for reducing the size of the data files in audio, image and video [20]. A variety of powerful and sophisticated wavelet-based image compression schemes have recently been developed. Such schemes are often lossy compression algorithms; however, lossy compression introduces distortions due to which useful information contained in images can be inevitably lost [16], [11]. Distortions introduced by lossy compression could lead into loss of sensitive data when used in areas such as medical imaging and remote sensing. On the other hand, compression algorithms based on regions with different compression ratios can alleviate such loss by preserving the details over a particular object or area. Given a ratio compression n, such algorithms isolate one or several regions of interest (ROI) from the background. Then, the background is compressed at higher ratios than n while all ROIs are compressed at lower ratios than n. Overall, the image will be compressed at a ratio of n, but the quality of the ROIs will be higher than the background when the image is reconstructed. Standards like MPEG4 and JPEG2000 define an operation mode using ROIs.

Current research has been focused on incorporating the human perception into the coding systems to increase the quality of the coding algorithms [5]. The objective of perceptual coding is to achieve maximum visual quality of the decoded visual data by taking advantage of the human visual system (HVS) characteristics. While the HVS is characterized by a large field of view, one of its characteristics is that the amount of details that the HVS can perceive declines rapidly from the point of gaze [18]. Such an effect is known as fovea. Due to this decreasing of perception, images can be coded including less details in areas that are farther from the point of gaze. Image and video compression algorithms can be fed with the data of the point of gaze in order to achieve a better quality when the visual data are reconstructed. This is known as fovea compression.

Proposals for wavelet-based fovea compression are presented in [3], [15], [6] and [24]. The idea of these approaches is to modify the continuous wavelet



(a) Foveated Wavelet Transform using CWT (b) Quantized Wavelet Coefficients using DWT

Figure 1. Different foveation methods using wavelets

transform that decimate the coefficients using a weight function. Another approach using fovea points over a wavelet is discussed in [8]. Instead of using a fovea operator over the Continuous Wavelet Transform, a quantization operator $q(x)$ is applied to each coefficient of the discrete wavelet transform (DWT). Such a quantization operator is defined by a weight window. Figure 1 depicts the results from both approaches applied to the image Lenna. Figure 1a shows the results of the foveated continuous wavelet transform applied to the image. Figure 1b shows the results of foveation by applying a quantized operator applied to the coefficients of the DWT of the image. It can be seen that the CWT-based fovea approach shows an anti-aliasing behavior in the image (especially in the upper right corner) than the DWT-based fovea algorithm making the image details more diffused overall.

Recent research has also been focused on detection of human eye point of gaze through either hardware such as in [17], [4], [21] or using collaborative computation as in [18]. Because this research solves one of the main problems of fovea compression, investigating new algorithms for fovea compression becomes relevant.

The Set Partitioning In Hierarchical Tree (SPIHT) algorithm does not allow to define ROIs. In [7], [14] and [13], different proposals for ROI compression with SPIHT are presented. In this work, we propose to exploit the visual sensitivity reduction caused by increasing a pixel distance from the fixation point as a ROI isolation criteria. This approach takes advantage of the particular area of the structure of the human retina called fovea. The use of the fovea can potentially increase the quality of the perception of the reconstructed image while maintaining a high data quality over the ROI and isolating data loss outside of a given area around a fixation point. The proposed approach is based on the use of the Lifting Wavelet Transform (LWT) and the SPIHT algorithm applied as a multi-resolution compression method creating foveas over different areas of the image. The LWT has the advantage over the DWT and the CWT of being computationally less complex and yields into a sharper reconstructed images that the ones coded with the CWT. The remainder of this paper is organized as follows. In Section II an overview of the SPIHT algorithm is given. In Section III an overview of fovea based compression is reported. Section IV presents the proposed approach, Section V presents results and Section VI reports conclusions and future work.

2. The SPIHT algorithm

The algorithm called Set Partitioning In Hierarchical Trees (SPIHT) is a compression scheme based on wavelets proposed in [19]. It has important properties such as high compression ratios and progressive transmission. The SPIHT is based on bit-plane encoding and takes advantage of the statistical behavior of the wavelet coefficients. When one level wavelet decomposition is applied to an image, four bands are obtained: an LL band or approximation coefficients band, and three detail coefficients bands called HL, HH, LH. Higher levels of decomposition over the LL sub-band applied recursively yield into more detail coefficients HL_l, LH_l and HH_l where l is the level of decomposition to which the sub-band belongs.

Let $C(\cdot)$ be a set of wavelet coefficients from a wavelet decomposition of the image I with I_R rows and I_C columns and let $K \in N$ be the number of decomposition levels. There is a relation [25] between a coefficient $C(i,j)$ with $C(i,j) \in HL_l \cup LH_l \cup HH_l$ and $1 < l \leq K$ and the set of coefficients $O(i,j)$ known as *offspring* defined by The coefficient $C(i,j)$ is called parent of $O(i,j)$. Applying such property

$$O(i,j) = \{(2i,2j),(2i,2j+1),(2i+1,2j),(2i+1,2j+1)\}. \quad (1)$$

recursively from the highest band with K levels of decomposition to $K-1$ yields into a set $D(i,j)$ known as quadtree with the root $C(i,j)$. $\mathcal{D}(i,j)$ is defined as

$$\mathcal{D}(i,j) = \begin{cases} \mathcal{O}(i,j) \cup \mathcal{D}(k,m) \text{ where } (k,m) \in \mathcal{O}(i,j) & if\ 0 \le 2*i \le I_R \text{ and } 0 \le 2*j \le I_C \\ \mathcal{O}(i,j) & \text{otherwise} \end{cases} \tag{2}$$

where $C(i,j) \in HL_l \cup LH_l \cup HH_l$ and $(k,m) \in \mathcal{O}(i,j)$. Lastly, a set $L(i,j)$, called *grand-descendant*, is defined by

$$\mathcal{L}(i,j) = \mathcal{D}(i,j) \setminus \mathcal{O}(i,j). \tag{3}$$

Given a threshold T if all coefficients from a quadtree D are lower than T, such a quadtree is called a zerotree [22]. Zerotrees are common in wavelet decompositions and SPIHT exploits such property for compression. If a quadtree is a zerotree, SPIHT only outputs a zero instead of sending bits for each coefficient of the zerotree. SPIHT defines three lists: list of insignificant pixels (LIP), list of insignificant sets (LIS) and list of significant pixels (LSP). LIP stores the position of each pixel that are lower than a given threshold, LIS stores the position of each root of all zerotrees for a given threshold and LSP stores the position of all coefficients higher than a given threshold.

The SPIHT algorithm is defined as a five steps algorithm:

1. Initialization. The threshold is initialized as $T = 2^{\lfloor \log_2(\max(|C(i,j)|)) \rfloor}$ with $C(i,j) \in LL \cup HL_l \cup LH_l \cup HH_l$ and $1 \le l \le K$. All pair (i,j) where $C(i,j) \in LL \cup HL_K \cup LH_K \cup HH_K$ is inserted into LIP and all pair (i,j) where $C(i,j) \in HL_K \cup LH_K \cup HH_K$ is inserted into LIS.
2. Significance pass. Check all elements $C(i,j)$ with $(i,j) \in LIS$. If the absolute value of the coefficient $|C(i,j)|$ is higher than a threshold T, the algorithm outputs a 1 followed by the sign of $C(i,j)$. Then, (i,j) is deleted from LIS and stored into LSP. Also, a matrix of thresholds W_Q is updated with $W_Q(i,j) = T$. Then, all the coefficients from the quadtrees that the position of its root (i,j) is inside LIS are compared against the threshold T in order to determine which quadtree is a zerotree. If a quadtree is not a zerotree when at least one of the coefficients $|C(i',j')|$ that belongs to that quadtree is higher than the threshold T. If that coefficient $C(i',j')$ belongs to the set $HL_\delta \cup LH_\delta \cup HH_\delta$ with $1 \le \delta \le K$ each coefficient $C(k,m) \in HL_\phi \cup LH_\phi \cup HH_\phi$ with $1 \le \phi < \delta$ is classified and its position inserted into LIS. After that, all coefficients are checked again. If a coefficient $|C(k,m)|$ is lower than T a 0 is output by the significance pass. If $|C(k,m)|$ is higher than the threshold T the significance pass outputs a 1 and its sign, the matrix of thresholds W_Q is updated with $W_Q(k,m) = T$ and inserted into LSP. All positions (k,m) where $C(k,m) \in HL_\delta \cup LH_\delta \cup HH_\delta$ are also stored inside LIS if $C(k,m) < T$.
3. Refinement pass. Each coefficient $C(i,j)$ with $(i,j) \in LSP$ is evaluated. If $|C(i,j)| \in [W_Q(i,j), W_Q(i,j) + T)$ then the refinement pass outputs a 0. Else if $|C(i,j)| \in [W_Q(i,j) + T, W_Q(i,j) + 2T)$ then the refinement pass outputs a 1 and updates $W_Q(i,j)$ as $W_Q(i,j) = T$.
4. The threshold is updated as $T = T/2$.
5. Return to 2.

The algorithm can be stopped either on an arbitrary value of T or if a bit per pixel (bpp) compression ratio is met for the output.

3. Fovea compression

Foveated images are images which have a non-uniform resolution [3]. Silverstein [23] reported how the human eye experiments a form of aliasing from the fixation point or fovea point to the edges of the image. Such aliasing increases in a logarithmic rate on all directions. This can be seen as concentric cutoff frequencies from the fixation point. Foveated images have been exploited in video and image compression. The use of fovea points yields reduced data dimensionality, which may be exploited within a compression framework. A foveated image can be represented by the following expression [6]

$$I_0(x) = \int I(t) C^{-1}(x) s\left(\frac{t-x}{w(x)}\right) dt \tag{4}$$

where $C(x) = \left\| s\left(\frac{-x}{w(x)}\right) \right\|$, $I(x)$ is a given image and $I_0(x)$ is the foveated image. The function s is called the weighted translation of s by x. Several weighted translation functions have been proposed [15]. For wavelets, foveation can be applied in both the wavelet transform [3], and the wavelet coefficients [9], [10]. Given a foveation operator T with a weight function $w(x) = \alpha|x|$, and a smooth function $g(x)$ with support on $[-\alpha^{-1}, \alpha^{-1}]$, a 2D wavelet transform is defined by

$$\begin{aligned} \theta_{l,m,u,n} &= \langle T\psi_{l,m}, \psi_{u,n} \rangle \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \psi_{l,m}(t) \psi_{u,n}(x) \frac{1}{|x|} g\left(\frac{t-x}{\alpha|x|}\right) dt dx \end{aligned} \tag{5}$$

where $\theta_{l,m,u,n}$ can be viewed as the wavelet transform of the kernel along with the t-axis followed by wavelet transform along the x-axis, and $\{\phi_{l_0,n}\} 0 \le n \le 2^{l_0} \cup \{\psi_{l,n}\} l < l_0, 0 < n < 2^{-j}$ define an orthonormal wavelet basis, $\psi_{l,n}(\cdot)$ and $\varphi_{l_0,n}(\cdot)$ represent scaled and translated versions of the mother wavelet $\psi(\cdot)$ and scaling function $\varphi(\cdot)$ respectively, and the notation $< .,. >$ defines the scalar product between two vectors [3].

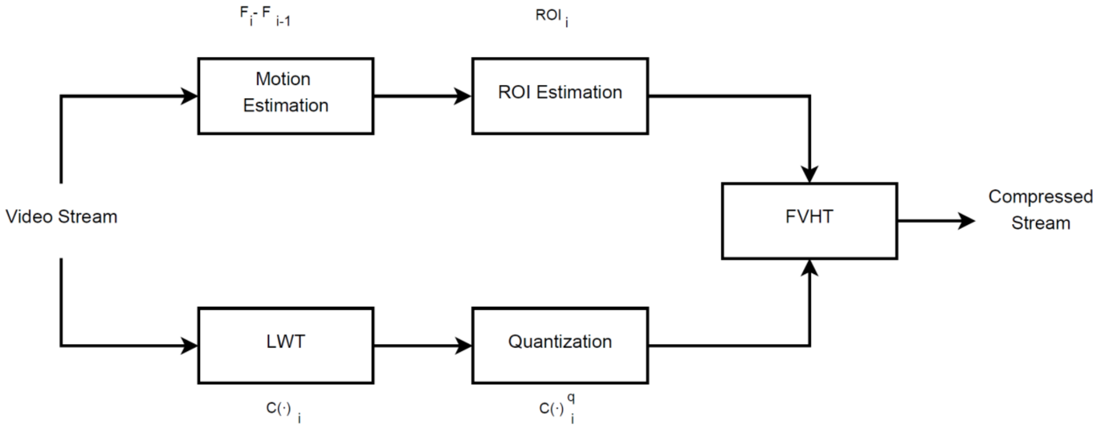


Figure 2. Block diagram of the proposed approach

4. Proposed approach

Image compression on the frequency domain using real valued coefficients is done through coefficient quantization. After quantization, all coefficients become integer valued for further compression using a Variable Length Coding algorithm such Run Length Encoding or Arithmetic Encoding. We propose to use a variable quantization of each coefficient determined by the decomposition band where the coefficient belongs to and its distance from a given fixation point. Such a quantization exploits the fovea effect of the Human Visual System (HVS). The proposed approach uses a fovea window centered at a given fixation point to determine how to quantize each wavelet coefficient. The coefficients are quantized and compressed using a modified SPIHT algorithm. The reconstructed image will have the highest visual quality near the fixation point and it will decrease as the pixels are farther from the fixation point.

Figure 2 shows a block diagram of the proposed Fovea Hierarchical Tree (FVHT) algorithm when applied to a video stream. Given a video stream with frames F_i, the applied blocks are the following:

- *Motion Estimation*: The fovea points are estimated by using the movement of the objects in the video scene. In order to compute the absolute difference a fast algorithm for motion estimation is used. This block outputs the difference between two subsequent video frames F_i and F_{i-1}.
- *ROI Estimation*: The output from the absolute difference block is used as input for ROI Estimation block. Each pixel from the motion estimation block that is different of 0 will be taken as a fovea center. This block outputs an array of fovea points as ROI_i
- *LWT*: This block applies the Lifting Wavelet Transform (LWT) to the input frame and outputs the coefficients as $C(\cdot)_i$.
- *Quantization*: The coefficients generated by the LWT block $C(\cdot)_i$ are transformed to integers using a fixed quantization for compression $C(\cdot)_i^q$
- *FVHT*: The FVHT block outputs a compressed stream of the quantized coefficients $C(\cdot)_i^q$ using the information of the estimated fovea points ROI_i.

If FVHT is applied to an image, the fovea points ROI_i must be input as a parameter to the FVHT instead of using motion estimation. Given a fixation point, the window parameters are computed and a cutoff window is created. This window will be used by the modified SPIHT algorithm in order to evaluate the compression ratio of each coefficient. The proposed approach takes an image or video frame and calculates the fovea cutoff window and the LWT of the image. Then, the coefficients are quantized to integers and the window is applied to get a compressed stream. In [15], when the fovea is applied to an image, the highest quality is reached at the pixel located on the fixation point. The proposed approach allows to define a ROI of variable size around the fixation point that will retain the best quality.

A. Fovea Cutoff Window

Fovea compression is expressed through a fovea cutoff window. Ideal fovea cutoff window is a logarithmic function as reported in [2]. However, such a function preserves only the center area of the size of a pixel of the fovea region. In order to have an arbitrary ROI size for the FVHT algorithm, a decaying window equation is proposed,

$$w(n_{x,y}) = \begin{cases} d\left(\frac{-\|n_{x,y}\|-\alpha}{1-\alpha}\right) & where - 1 \leq \|n_{x,y}\| - \alpha \\ 1 & where - \alpha \leq \|n_{x,y}\| \alpha \\ d\left(\frac{\|n_{x,y}\|-\alpha}{1-\alpha}\right) & where\ \alpha \leq \|n_{x,y}\| \leq 1 \\ 0 & \text{where } n \geq 1 \end{cases} \tag{6}$$

where $||n_{x,y}||$ is the ∞-norm from the point (x, y) to the fovea center F, α is the radius of the ROI and $d(x)$ is a decreasing function. $d(x)$ can be defined according to the needs of the implementation. d(x) must satisfy the following conditions:

1. It is a monotonically decreasing function over the interval $[0, 1]$.
2. It is continuous over the interval $[0, 1]$.
3. when $x = 0$, $d(x) = 1$.
4. when $x = 1$, $d(x) = 0$.

If $d(x)$ is a logarithmic function, then it gives the ideal fovea cutoff. The quality around the ROI is related to how fast the function $d(x)$ decreases. Better image quality around the fixation point is obtained when d decrements its value slowly when x is closer to 0. However, regions that are farther from the fixation point will have lesser quality because most bits will be used to encode the region of interest. A useful function for d is the power law function used in histogram equalization [23]:

$$d(x) = c(1 - x)^{\gamma} \tag{7}$$

where c and γ are positive constants and $x \in [0, 1]$. The parameter $1 - x$ was introduced as a modification of the original formula in order to satisfy the conditions of the proposed window in (6). The parameters c and γ can be used to easily define the decaying speed of the window. Also, a useful and sightly less complex function derived from (7) is the ramp function:

$$d(x) = c(1 - x) \tag{8}$$

Figure 3 shows decaying windows calculated with the proposed power law function. The decaying windows in Figure 3 were calculated using $c = 1$ and $\gamma = 2, \gamma = 40, \gamma = 0.5, \gamma = 0.04$ as parameters. Note that using $c \neq 1$ will create a window that scales the desired compression boundaries. The distance from the point (x, y) to the fovea center F represented by the parameter $||n_{x,y}||$ in (6) is calculated using the ∞-norm normalized to the interval $[0, 1]$ defined by

$$||n_{x,y}|| = \frac{max\{|x - f_x|, |y - f_y|\}}{N} \tag{9}$$

where (x, y) is an arbitrary position of a wavelet coefficient and (f_x, f_y) is the coordinate pair of the fovea centre F and N is the maximum distance between the fovea center F and the farthest corner of the image.

Figure 4 shows a comparison of two windows generated using the standard Euclidean norm and the proposed norm. In Figure 4a, the cutoff window was calculated using the ∞-norm for computing the distance between a coefficient and a given fixation point of $(256, 256)$. In Figure 4b, the euclidean norm was used with the same parameters. It can be seen how the ∞-norm only adds a few more points around the fovea corner but calculating such norm is less complex, making it viable for low computer-power devices.

If v fovea centers are given $F_0, F_1, \ldots, F_v$ where $v \in N$, the distance $\|n_{x,y}\|$ is given by

$$\|n_{x,y}\| = \min\{\|n_{x,y}^0\|, \|n_{x,y}^1\|, \ldots, \|n_{x,y}^v\|\} \tag{10}$$

where $\|n_{x,y}^i\|$ is the distance from (x, y) to the fovea center F_i. The Fovea Hierarchical Tree (FVHT) algorithm is a modified version of SPIHT that compress using fovea regions. The algorithm is fed with the wavelet coefficients, the fovea center F, the decaying function $d(x)$ and the lower and upper bounds $(b, L]$ with b as the lowest bit rate and L as the highest bit rate of the compression. The interval $(b, L]$ defines how the compression bit rate will increase as the pixels move farther from the fovea centroid. The overall compression ratio will be of L bpp. The previously defined decaying function defined in (6) is modified to

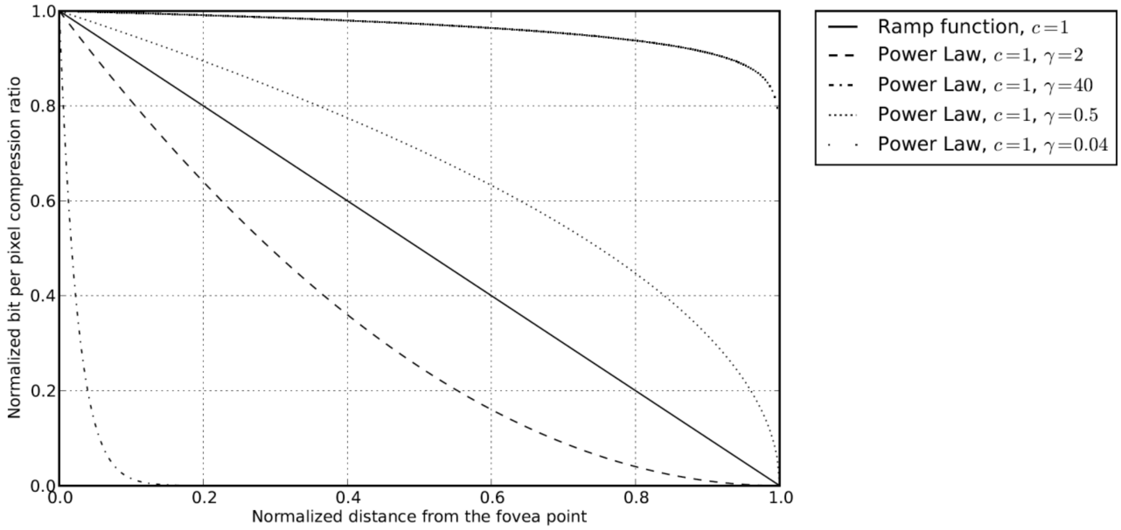
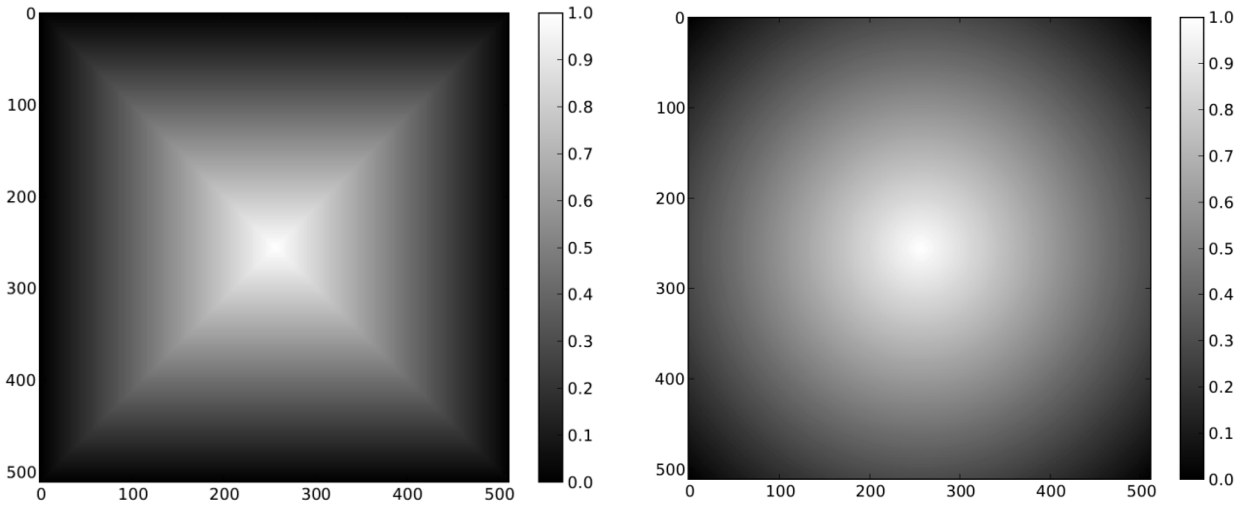


Figure 3. Decaying window comparison using different parameters for the power law function



(a) Fovea Cuttoff Window using the ramp function and ∞-norm

(b) Fovea Cutoff Window using the ramp function and euclidean norm

Figure 4. Different foveation methods using wavelets

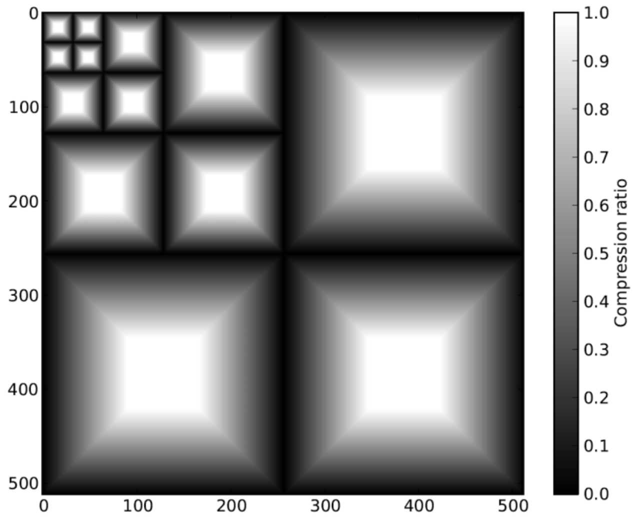


Figure 5. Fovea cutoff window for a 512×512 pixels image with four levels of decomposition, higher value means less compression ratio

$$w(n_{x,y}) = \begin{cases} L & where\ 0 \le n \le \alpha \\ d\left(\frac{\|n_{x,y}^{l}\| - \alpha^{l}}{1 - \alpha^{l}}\right) & where\ \alpha \le n \le 1 \\ b & where\ n \ge 1. \end{cases} \quad (11)$$

In order to apply the fovea cutoff window to each wavelet sub-band, the fovea center is updated accordingly,

$$f^{l} = \left(\frac{x}{2^{l}}, \frac{y}{2^{l}}\right) \quad (12)$$

where l is the decomposition level to which the sub-band belongs. From (11), on each pass of the algorithm, the distance function of each coefficient position for each sub-band with scale l is evaluated as follows:

$$\|n_{x,y}^{l}\| = \frac{2^{l} max\{|x - f_{x}^{l}|,\ |y - f_{y}^{l}|\}}{N} \quad (13)$$

Also, α from (11) is defined for each sub-band as:

$$\alpha^{l} = \frac{\alpha}{2^{l}} \quad (14)$$

The cutoff window for each wavelet decomposition sub-band is expressed using (11), (13) and (14) as follows:

$$w^{l}(n_{x,y}^{l}) = \begin{cases} L & where\ 0 \le n \le \alpha \\ d\left(\frac{\|n_{x,y}^{l}\| - \alpha^{l}}{1 - \alpha^{l}}\right)(L - b) + b & where\ \alpha \le n \le 1 \\ b & where\ n \ge 1. \end{cases} \quad (15)$$

Figure 5 shows a window generated for a wavelet decomposition of an image of 512×512 pixels with four levels of decomposition. It can be seen how the window is scaled across all sub-bands. A value of 1 means that coefficient is compressed with the lowest compression ratio given, while a coefficient with a window value of 0 is compressed with the highest compression ratio given.

The proposed algorithm determines the compression bit rate for each coefficient evaluating the decaying window function on each algorithm pass at every coefficient coordinates. If the current bit rate is lower than $w^l(n^l_{x,y})$, then the coefficient is encoded, otherwise it is discarded. Each quadtree is evaluated besides the distance of its root to its corresponding scaled fovea center to avoid loss of information if an element of the quadtree should still be encoded besides its root distance. The distance is evaluated on each pass in order to not increase the memory usage of the algorithm. The resultant image has a bit-rate of L. The distance evaluation should be done in both the significance pass and the refinement pass. However, on the significance pass, the positions of the coefficients are discarded from the LIP and in the refinement pass they are discarded from the LSP. The list LIS remains the same as in the SPIHT algorithm.

B. FVHT Algorithm

The proposed FVHT algorithm is described as follows:

```
1. Initialization. The threshold is initialized as $T = 2^{\lfloor \log_2(max(|C(i,j)|)) \rfloor}$ with $C(i,j) \in LL \cup HL_l \cup LH_l \cup HH_l$ and $1 \leq l \leq K$. Each $(i,j)$ with $C(i,j) \in LL \cup HL_K \cup LH_K \cup HH_K$ is inserted into $LIP$ and each $(i,j)$ with $C(i,j) \in HL_K \cup LH_K \cup HH_K$ is inserted into $LIS$ as entry type A.
2. Significance pass
  2.1. $\forall(i,j) \in LIP$ do:
    2.1.1. output $S_n(i,j)$
    2.1.2. if $S_n(i,j) = 1$ then:
      • $LIP = LIP \setminus \{(i,j)\}$
      • if $w^l(n^l_{i,j})$ is greater than the current bpp then:
        ⋆ $LSP = LSP \cup \{(i,j)\}$
  2.2. $\forall(i,j) \in LIS$ do:
    2.2.1. if the entry is of type $A$ then:
      2.2.1.1. output $S_n(\mathcal{D}(i,j))$
      2.2.1.2. $\forall(k,m) \in \mathcal{O}(i,j)$ do:
        • output $S_n(k,m)$
        • if $S_n(k,m) = 1$ and $w^l(n^l_{k,m})$ is greater than the current bpp, then add $(k,m)$ to the LSP andoutput the sign of $C(k,m)$
        • if $S_n(k,m) = 0$, then add $(k,m)$ to the end of the LIP
      2.2.1.3. if $\mathcal{L}(i,j) \neq \emptyset$, then move $(i,j)$ to the end of the LIS, as an entry of type $B$; else, remove $(i,j)$ from the LIS
    2.2.2. if entry is of type $B$ then:
      • output $S_n(\mathcal{L}(i,j))$
      • if $S_n(\mathcal{L}(i,j)) = 1$ then:
        ⋆ $LIS = LIS \cup \mathcal{O}(i,j)$ where $(k,l) \in \mathcal{O}(i,j)$ are of type A
        ⋆ $LIS = LIS \setminus \{(i,j)\}$
3. Refinement pass: $\forall(i,j) \in LSP$, except for the coefficients included in the last sorting pass (i.e., with the same $T$), output the $n$-th most significant bit of $|C(i,j)|$ iff $w^l(n^l_{i,j})$ is greater than the current bit per pixel ratio. Else, remove $(i,j)$ from the LSP.
4. Set $T = T/2$.
5. Return to 2.
```

As in SPIHT, the proposed algorithm checks for zero trees on each wavelet sub-band from the highest to the lowest band. At the beginning of the algorithm, all quadtree roots are marked as type A. In each pass, the roots are checked for to be significant. When a quadtree root is significant, its Type changes to B either if it has descendants at two decomposition levels from its position, or if not, it is just removed from the list. This sorting allows the algorithm to guarantee the best quality of the reconstructed image [19].

5. Simulation results

The proposed algorithm was implemented on python and tested using standard non-compressed 512×512 pixel images. The fovea center used was the center pixel (256, 256) with parameters of $\alpha = 0.3$ and the power law function with parameters $c = 1$ and

$\gamma = 1$ (the ramp function) for easy identification of the fovea center and to spot easily the compression artifacts on the tested reconstructed images. The wavelet used was the biorthogonal CDF 9/7 with four levels of decomposition. This wavelet, chosen for the JPEG2000 standard, shows a good PSNR when the image is reconstructed [1]. The proposed algorithm was tested against the SPIHT algorithm. Figure 6 shows the reconstructed wavelet decomposition of the image cameraman. Figure 6a shows the reconstructed wavelet coefficients using standard SPIHT algorithm and Figure 6b shows the reconstructed wavelet coefficients using the proposed FVHT. In Figure 6b it is shown how FVHT gives preference to pixels inside the fovea area specially on the last decomposition sub-band where coefficients from around the center enclosed within a dashed circle were recovered. On the other hand, the same coefficients were not recovered when using the SPIHT algorithm. The coefficients preserved by FVHT enclosed inside the dashed circle in Figure 6b allow a better reconstruction of the fovea area. This is because the FVHT algorithm keeps the coefficients near the fovea center and discards those farther from the fixation point when the data stream is closer to meet the compression ratio. Figure 7 shows the reconstructed image with both SPIHT and FVHT algorithms. Figure 7a shows the reconstructed image with the SPIHT algorithm at 1 bit per pixel (bpp) compression rate. Figure 7b shows the same reconstructed image using the FVHT algorithm at 1 bpp as its higher compression ratio and at 0.06 bpp as its lower compression ratio. Figure 7b shows more artifacts around the edges because that area was compressed at a higher compression ratio than the with the SPIHT algorithm. However, the area of the image closer to the fixation point shows less distortions on the FVHT reconstructed image. The data stream obtained from both images had a final compression ratio of 1 bpp.

The PSNR (Peak Signal to Noise Ratio) is commonly used as a measure of quality of reconstruction of lossy compression. The function is defined by

$$PSNR = 10\log\left(\frac{255^2}{MSE}\right) \tag{16}$$

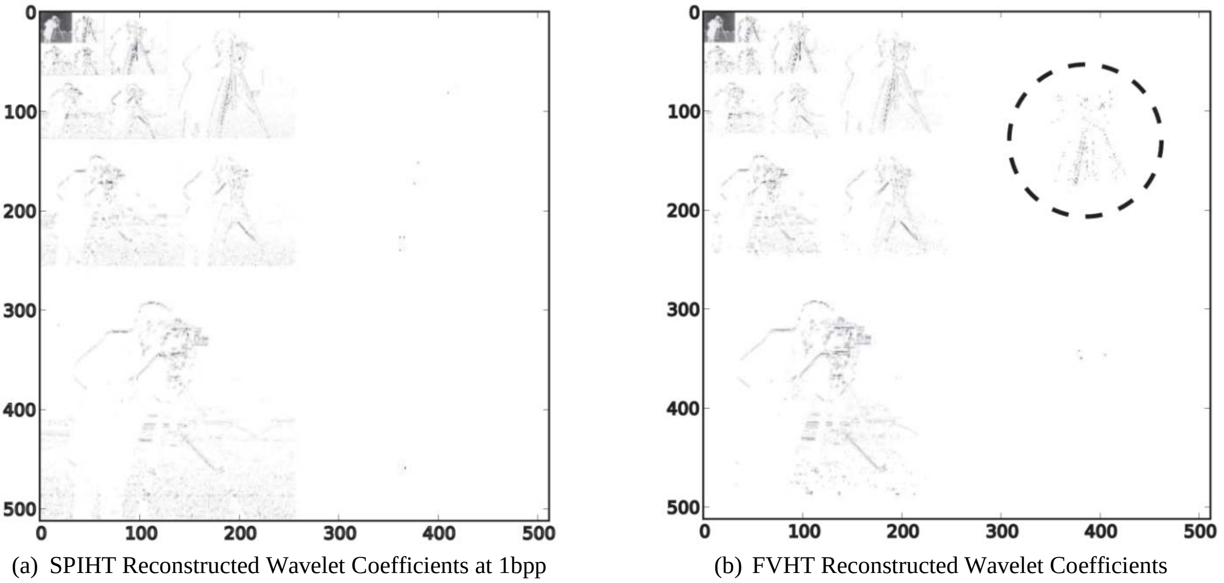
where 255 is the maximum value of the pixel for an 8-bit image. MSE is the mean squared error defined by

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}\left[I(i,j) - I_q(i,j)\right]^2 \tag{17}$$

where I is the original image, I_q is the reconstructed image and m and n are the number of rows and columns of the image respectively.

Figure 8 shows a comparison of the PSNR of tested images compressed with both the SPIHT and the FVHT at different ratios. It can be seen how the SPIHT shows better performance on overall distortions. However, the main advantage of FVHT is to preserve the area around the fixation point. In order to test the performance of FVHT over smaller areas, several sub-images of different sizes were extracted from the reconstructed and original images. Each subimage was centered on the fovea center. The size of the first subimage was 80 pixels radius r.

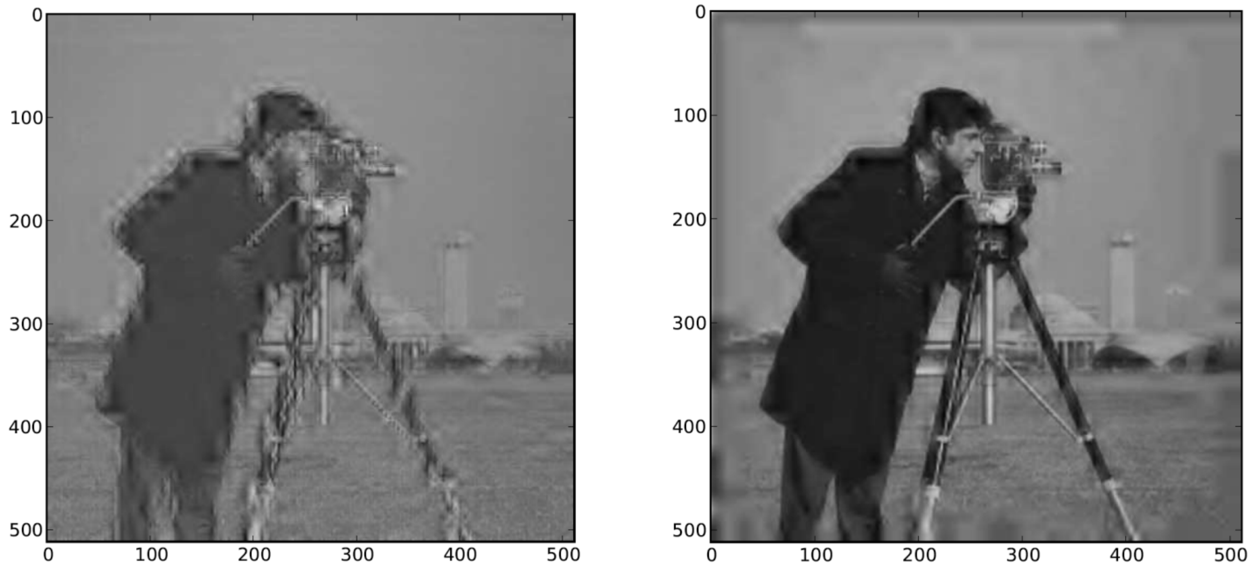
This radius was chosen because this is the edge of the ROI when $\alpha = 0.3$. The total size of the each sub-image was $2r \times 2r$ pixels. Several subimages were taken with incremental size of one pixel radius each. Figure 9 shows a PSNR comparison of the SPIHT algorithm and the proposed FVHT algorithm. Two compression ratios are shown in Figure 9, namely, 1 bpp and 4 bpp. As expected, the FVHT algorithm has better performance



(a) SPIHT Reconstructed Wavelet Coefficients at 1bpp

(b) FVHT Reconstructed Wavelet Coefficients at 0.06-1 bpp range

Figure 6. Reconstructed Wavelet Coefficients Comparison using SPIHT and FVHT compression algorithms



(a) Reconstructed SPIHT compressed image “cameraman” at 1bpp

(b) Reconstructed FVHT compressed image “cameraman” at 0.06-1 bpp range

Figure 7. Reconstructed Image Comparison using SPIHT and FVHT compression algorithms at 1bpp and 0.06-1bpp compression ratios respectively

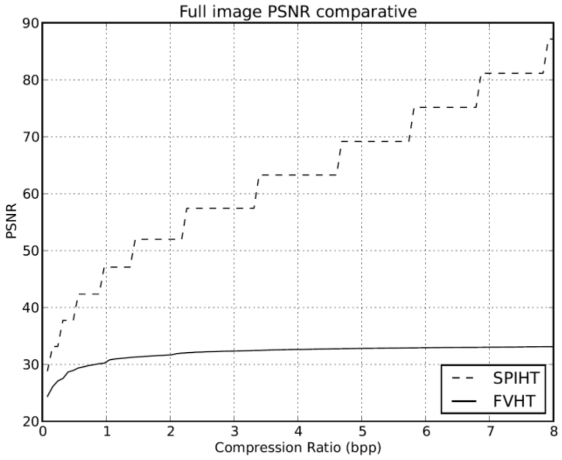


Figure 8. PSNR comparison of compressed “cameraman” image using SPIHT and FVHT

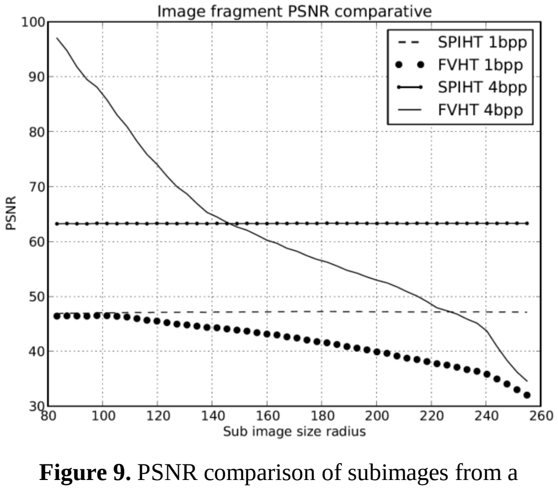


Figure 9. PSNR comparison of subimages from a reconstructed “cameraman” image compressed with SPIHT and FVHT

than SPIHT over small areas around the fovea centre and slowly decrease the quality of the reconstructed image of areas farther from the fovea center. When the compression ratio is 1 bpp both FVHT and SPIHT show the same PSNR over the fovea area. However, as the amount of bpp used increases, FVHT has better performance than SPIHT over small areas around the fovea center and slowly decrease the quality of the reconstructed image of areas farther from the fovea center as shown in Figure 9 with compression ratio of 4 bpp.

6. Conclusions and future work

Fovea compression together with ROI allows to control the bit rate compression of given areas and preserve image perception quality to the human eye.

Compression with hierarchical trees can be further improved as described in [7] by labeling beforehand each coefficient and using unbalanced quadtrees, however it will decrease the memory performance of the algorithm. The proposed approach reduces the memory usage by calculating on each pass the compression ratio. Also, each coefficient has different compression ratio. Thus, the fovea effect can be smoother than with fixed labeling. Future work will be focused on applying the proposed algorithm to video compression using automatic foveation such as in [12] and increasing the performance of FVHT based on faster versions of SPIHT such as the one proposed in [26].

Acknowledgments

The first author gratefully acknowledges the financial support from CONACYT, México.

References

[1] **T. Acharya, P.-S. Tsai**. *JPEG2000 Standard for Image Compression*. Hoboken, NJ, USA: John Wiley & Sons, Inc., Oct. 2004.

[2] **E.-C. Chang, S. Mallat, C. Yap**. Wavelet Foveation. *Applied and Computational Harmonic Analysis*, 2000, Vol. 9, No. 3, 312–335.

[3] **E.-C. Chang, C. K. Yap**. Wavelet approach to foveating images. In: *Proceedings of the thirteenth annual symposium on Computational geometry - SCG'97*, 1997, pp. 397–399.

[4] **K.-W. Chen, C.-W. Lin, T.-H. Chiu, M.-Y. Chen, Y.-P. Hung**. Multi-resolution design for large-scale and high-resolution monitoring. *IEEE Transactions on Multimedia*, 2011, Vol. 13, No. 6, 1256–1268.

[5] **Z. Chen, W. Lin, K. N. Ngan**. Perceptual video coding: challenges and approaches. In: *2010 IEEE International Conference on Multimedia and Expo (ICME)*, Jul. 2010, pp. 784–789.

[6] **I. B. Ciocoiu**. ECG signal compression using 2D wavelet foveation. In: *Proceedings of the 2009 International Conference on Hybrid Information Technology - ICHIT'09*, 2009, Vol. 13, pp. 576–580.

[7] **A. Cuhadar, S. Tasdoken**. Multiple arbitrary shape ROI coding with zerotree based wavelet coders. In: *Proceedings of the IEEE International Conference on Multimedia and Expo, ICME 2003*, 2003, pp. 157–160.

[8] **J. C. Galan-Hernandez, V. Alarcon-Aquino, O. Starostenko, J. M. Ramirez-Cortes**. DWT Foveation-based multiresolution compression algorithm. *Research in Computing Science*, 2010, 197–206.

[9] **J. C. Galan-Hernandez, V. Alarcon-Aquino, O. Starostenko, J. M. Ramirez-Cortes**. Wavelet-based foveated compression algorithm for real-time video processing. In: *Electronics, Robotics and Automotive Mechanics Conference (CERMA)*, 2010, pp. 405–410.

[10] **J. C. Galan-Hernandez, V. Alarcon-Aquino, O. Starostenko, J. M. Ramirez-Cortes**. Foveated ROI compression with hierarchical trees for real-time video transmission. In: *Proceedings of the Third Mexican conference on Pattern recognition*. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 240–249.

[11] **D. Goodenough, A. Dyk, T. Han, A. Jazayeri, J. Li**. Impacts of lossy compression on hyperspectral products for for estry. In: *IEEE International Geoscience and Remote Sensing Symposium 2004 (IGARSS'04) Proceedings*, Sep. 2004, Vol. 1, pp. 465–468.

[12] **L. Itti**. Automatic foveation for video compression using a neurobiological model of visual attention. *IEEE Transactions on Image Processing*, 2004, Vol. 13, No. 10, 1304–1318.

[13] **D. Kancelkis, J. Valantinas**. A new le Gall wavelet-based approach to progressive encoding and transmission of image blocks. *Information Technology and Control*, 2012, Vol. 41, No. 3, 239–247.

[14] **K.-H. Park, H.-W. Park**. Region-of-interest coding based on set partitioning in hierarchical trees. *IEEE Transactions on Circuits and Systems for Video Technology*, 2002, Vol. 12, No. 2, 106–113.

[15] **S. Lee, A. Bovik**. Fast algorithms for foveated video processing. *IEEE Transactions on Circuits and Systems for Video Technology*, 2003, Vol. 13, No. 2, 149–162.

[16] **V. V. Lukin, M. S. Zriakhov, N. N. Ponomarenko, S. S. Krivenko, M. Zhenjiang**. Lossy compression of images without visible distortions and its application. In: *Signal Processing (ICSP), 2010 IEEE 10th International Conference on*, Oct. 2010, pp. 698–701.

[17] **M. Reale, S. Canavan, L. Yin, K. Hu, T. Hung**. A multi-gesture interaction system using a 3D iris disk model for gaze estimation and an active appearance model for 3D hand pointing. *IEEE Transactions on Multimedia*, 2011, Vol. 13, No. 3, 474–486.

[18] **F. Ribeiro, D. Florencio**. Region of interest determination using human computation. In: *Multimedia Signal Processing (MMSP), 2011 IEEE 13th International Workshop on*, Oct. 2011, pp. 1–5.

[19] **A. Said, W. Pearlman**. A new, fast, and efficient image codec based on set partitioning in hier archical trees. *IEEE Transactions on Circuits and Systems for Video Technology*, 1996, Vol. 6, No. 3, 243–250.

[20] **D. Salomon**. *Data Compression: The Complete Reference*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.

[21] **S. Samanta, S. Saha, B. Chanda**. A simple and fast algorithm to detect the fovea region in fundus retinal im age. In: *Emerging Applications of Information Technology (EAIT), 2011 Second International Conference*, Feb. 2011, pp. 206–209.

[22] **J. Shapiro**. Embedded image coding using zerotrees of wavelet coefficients. *IEEE Transactions on Signal Processing*, 1993, Vol. 41, No. 12, 3445–3462.

[23] **L. D. Silverstein**. Foundations of vision. *Color Research & Application*, 2008, Vol. 21, No. 2, 142–144.

[24] **G. R. Suresh, S. Sudha, R. Sukanesh**. Shape adaptive wavelet transform for magnetic resonance images coding. *International Journal of Electronics*, 2009, Vol. 96, No. 6, 613–622.

[25] **P. Tsai**. Tree structure based data hiding for progressive transmission images: a review of related works. *Fundamenta Informaticae*, 2010, Vol. 98, 257–275.

[26] **J. Valantinas, D. Kancelkis**. Speeding-up image encoding times in the SPIHT algorithm. *Information Technology and Control*, 2011, Vol. 40, No. 1, 7–11.

Received December 2012.