Foveated ROI Compression with Hierarchical Trees for Real-Time Video Transmission

J.C. Galan-Hernandez[1,⋆], V. Alarcon-Aquino[1], O. Starostenko[1], and J.M. Ramirez-Cortes[2]

[1] Department of Computing, Electronics, and Mechatronics
Universidad de las Americas Puebla
Sta. Catarina Martir, Cholula, Puebla. C.P. 72810, Mexico
{juan.galanhz,vicente.alarcon}@udlap.mx
[2] Department of Electronics
Instituto Nacional de Astrofisica, Optica y Electronica
Tonantzintla, Puebla Mexico

Abstract. Region of interest (ROI) based compression can be applied to real-time video transmission in medical or surveillance applications where certain areas are needed to retain better quality than the rest of the image. The use of a fovea combined with ROI for image compression can help to improve the perception of quality and preserve different levels of detail around the ROI. In this paper, a fovea-ROI compression approach is proposed based on the Set Partitioning In Hierarchical Tree (SPIHT) algorithm. Simulation results show that the proposed approach presents better details in objects inside the defined ROI than the standard SPIHT algorithm.

Keywords: Compression, Fovea, ROI, SPIHT, Wavelet Transforms.

1 Introduction

Video and image compression can help in reducing the communication overhead. Lossy compression is a common tool for achieving high compression ratios; however, more information from the image is lost as the compression rate increases. Compression algorithms based on regions with different compression ratios are important for applications where is needed to preserve 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 and then the background is compressed at higher ratios than n while all ROIs are compressed at lower ratios than n achieving a better reconstruction of the ROIs. Standards like MPEG4 and JPEG2000 define an operation mode using ROIs.

The proposed approach for ROI coding over real-time video transmission is to take advantage of the structure of the human retina, called fovea, for increasing

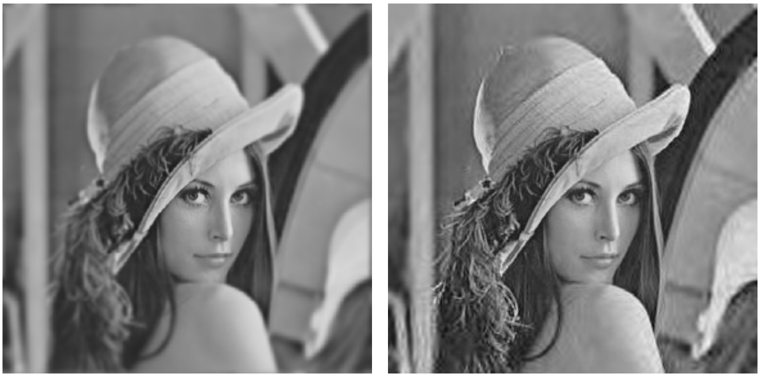
[⋆] The authors gratefully acknowledge the financial support from the National Council of Science and Technology and the Puebla State Government, under the contract no. 109417.

J.-F. Martínez-Trinidad et al. (Eds.): MCPR 2011, LNCS 6718, pp. 240–249, 2011.

the quality of the perception of each reconstructed frame while maintaining a high data quality over the ROI. Such ROI is defined by a motion detection algorithm. This approach is based on the use of a Lifting Wavelet Transform and a modified version of the SPIHT algorithm that allows to define Foveated areas of the image.

1.1 Previous Works

Proposals for wavelet based fovea compression are presented in [1]-[2]. Th idea of these approaches is to modify the continuous wavelet transform that decimate the coefficients using a weight function. Another approach using fovea points over a wavelet is discussed in [3]. Instead of using a Fovea operator over the Continuous Wavelet Transform (CWT), a quantization operator $q(x)$ is applied to each coefficient of the discrete wavelet transform (DWT). Such quantization operator is defined by a weight window. Figure 1 depicts the results from both methods 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 DWT coefficients of the image. It can be seen that the CWT-based fovea approach shows a softer behavior in the image (especially in the upper right corner) than the DWT-based fovea algorithm.



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

Fig. 1. Different foveating methods using wavelet transforms

The Set Partitioning In Hierarchical Tree algorithm (SPIHT) does not allow to define ROIs. In [2] and [4], different proposals for ROI compression with the SPIHT algorithm are presented. In this paper we report a fovea-ROI compression approach based on a modified version of the SPHIT algorithm. The remainder of this paper is organized as follows. Section 2 reports a description of classical video compression. In Section 3 an overview of foveated compression is given. Section 4 describes the SPIHT algorithm. Section 5 reports the proposed approach, Section 6 presents results and Section 7 reports conclusions and future work.

2 Video Compression

Lately, video coding has evolved into two dominant standards: MPEG and ITU-T H.26x. Such recommendations are based on a classic video encoding framework [5] shown in figure 2.

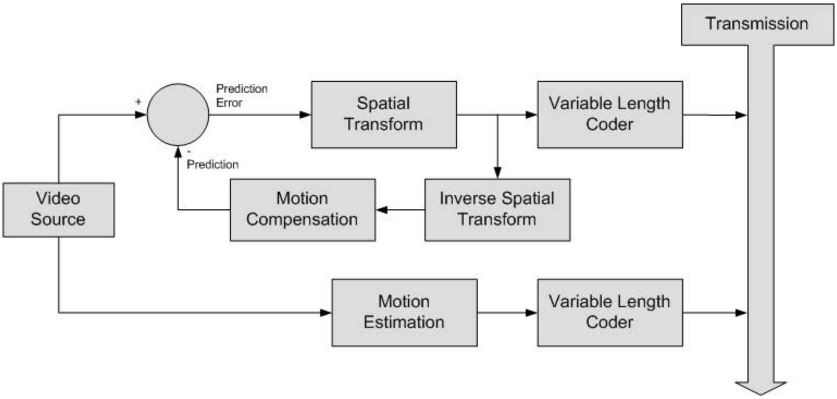


Fig. 2. Classic Video Encoding Framework

Two main parts in video compression shown in figure 2 are spatial transform and motion estimation. Spatial transformation is successively applied to individual video frames to take advantage of the high degree of data correlation in adjacent image pixels (spatial correlation).

Motion estimation exploits temporal correlation. Classic video coding takes the difference from two frames $e_n = f_{n-1} - f_n$ where f_i is the video frame i and e is called Motion Compensation Error Residual (MCER) [6]. Usually, a video sequence does not change but only in small segments from frame to frame. Using the MCER instead of the original frame reduces the amount of data to be transmitted because MCER will have more redundant data (zeros) overall.

However, the use of motion estimation adds an acummulative error over the coding because the coder uses the original frames for calculating the MCER and the decoder only have the decoded frames that, when used lossy compression algorithm, are not a perfect reconstruction of the original frame. For improving the quality of the compression, a feedback from the video encoded frames are used for calculating a motion compensation vector. Such motion compensation can be calculated either from the coder alone such in classic compression or by the coder using feedback from the decoder such in Distributed Video Coding (DVC) [7,8].

3 Foveated Compression

Foveated images are images which have a non-uniform resolution [1]. Researchers have demonstrated that the human eye experiments a form of aliasing from the fixation point or fovea point to the edges of the image [9]. 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 [10]

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

where $I(x)$ is a given image and $I_0(x)$ is the foveated image. The function s is called the weighted translation of s by x. There are several weighted translation functions, such the ones defined in [11].

For wavelets, foveation can be applied in both the Wavelet Transform [1], and the wavelet coefficients [12]. 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

$$\theta_{j,m,k,n} = \langle T\psi_{j,m}, \psi_{k,n}\rangle = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \psi_{j,m}(t)\psi_{k,n}(x)\frac{1}{|x|}g\left(\frac{t-x}{\alpha|x|}\right)dtdx$$

where $\{\phi_{l_{0,n}}\}_{0\leq n\leq 2^{l_0}} \cup \{\psi_{j,n}\}_{j<l_0, 0<n<2^{-j}}$ define an orthonormal wavelet basis, $\phi_{l_{0,n}}(\cdot)$ and $\psi_{j,n}(\cdot)$ represent scaled and translated versions of the mother wavelet $\psi(\cdot)$.

4 SPIHT

The algorithm called Set Partitioning In Hierarchical Trees (SPIHT) is a compression scheme based on wavelets proposed in [13]. It has important properties such as high compression ratios and progressive transmission.

The SPIHT is based on bit-plane encoding and takes advantage of a property wavelet coefficient. When a one level wavelet decomposition is applied to an image, four bands are obtained: an LL band or approximation coefficients band, and three detail coefficients called HL, HH, LH. Higher levels of decomposition yields into more detail coefficients HL_n, LH_n and HH_n where n is the level of decomposition in which the sub band belongs. If $C(\cdot)$ is a set of wavelet coefficients from a wavelet decomposition of the image I and L is the level of decomposition, there is a relation [14] between a coefficient $C(i,j)$ with $C(i,j) \in HL_n \cup LH_n \cup HH_n$ and $1 < n \leq L$ and the coefficients $C(2i,2j)$, $C(2i,2j+1)$, $C(2i+1,2j)$ and $C(2i+1,2j+1)$. $C(i,j)$ is known as a parent and $C(2i,2j)$, $C(2i,2j+1)$, $C(2i+1,2j)$ and $C(2i+1,2j+1)$ coefficients are called *Offsprings.* Applying recursively such property from the highest band L to $L-1$ yields into a hierarchical tree known as quadtree with root $C(i,j)$.

Given a threshold T if all coefficients from a quadtree are lower than T, such quadtree is called a zerotree [15]. Zerotrees are common in wavelet decompositions and the SPIHT exploits such property for compression. If a quadtree is a zerotree, the SPIHT only outputs a zero instead of sending bits for each coefficient of the zerotree. The 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. Such lists are used for the two main steps of the algorithm: significance pass and refinement pass.

The significance pass checks all elements $C(i,j)$ with $(i,j) \in LIS$, if $|C(i,j)|$ is higher than a threshold T_l outputs a 1 followed by the sign of $C(i,j)$ and (i,j) are deleted from LIS and stored into LSP, also, a matrix of thresholds W_Q is updated with $W_Q(i,j) = T_l$. Then, checks all the coefficients from quadtrees that its root $(i,j) \in LIS$. If a quadtree is not a zerotree, at least one of the coefficients $|C(i',j')|$ that belongs to that quadtree is higher than the threshold T_l. If $C(i',j') \in HL_\delta \cup LH_\delta \cup HH_\delta$ with $1 <= \delta <= L$, each coefficients $C(k,m) \in HL_\phi \cup LH_\phi \cup HH_\phi$ with $1 <= \phi < \delta$ are classified and its position inserted into LIS if $|C(k,m)|$ are lower than T_l or if $|C(k,m)|$ is higher than T_l the significance pass outputs a 1 and its sign, the matrix of thresholds W_Q is updated with $W_Q(k,m) = T_l$ and inserted into LSP. All positions (k,m) where $C(k,m) \in HL_\delta \cup LH_\delta \cup HH_\delta$ are also stored in LIS if $C(k,m) < T_l$.

In the refinement pass, with a given threshold T_l, each coefficient $C(i,j)$ with $(i,j) \in LSP$ is evaluated. If $|C(i,j)| \in [W_Q(i,j), W_Q(i,j) + T_l)$ then outputs a 0 else if $|C(i,j)| \in [W_Q(i,j) + T_l, W_Q(i,j) + 2T_l)$ then outputs a 1 and $W_Q(i,j) = T_l$.

SPIHT is defined as a five steps algorithm:

1. Initialization. The threshold $T = 2^{\lfloor \log_2(\max(|C(i,j)|)) \rfloor}$ with $C(i,j) \in LL \cup HL_n \cup LH_n \cup HH_n$ and $1 <= n <= L$. Each $(i,j) \in LL \cup HL_L \cup LH_L \cup HH_L$ is inserted into LIP and each $(i,j) \cup HL_L \cup LH_L \cup HH_L$ is inserted into LIS.
2. Significance pass
3. Refinement pass
4. $T = T/2$
5. Return to 2

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

5 Proposed Approach

Our proposal is to mix fovea compression with ROI compression using the SPIHT algorithm. In this paper this algorithm is referred to as FVHT (Fovea Hierarchical Trees), which is a modified version of the SPIHT. The FVHT is applied to individual frames of a video stream for a real-time video transmission. The mix of both will yield into a better quality perception of each individual frame and preserve information around the ROI that can be useful for an observer instead of only making the ROI bigger. The wavelet decomposition is calculated with the Lifting Wavelet Transform (LWT) [16]. The LWT uses the lifting scheme and factorizes orthogonal and biorthogonal wavelet transforms into elementary spatial operators called liftings. The advantage of the LWT is three reduced operations to nearly two for its calculation [17]. The block diagram of the proposed approach is depicted in figure 3.

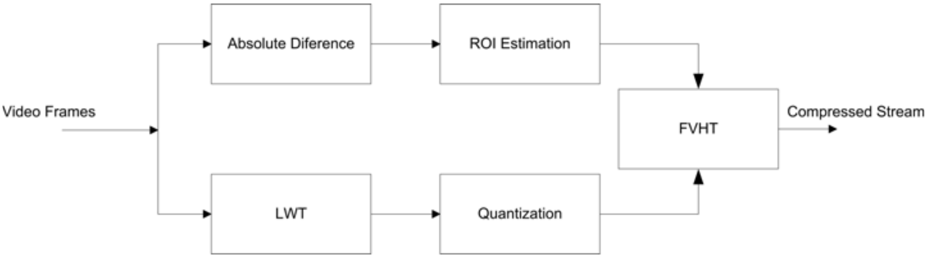


Fig. 3. Block diagram of the proposed approach

Given a video transmission of a moving object with frames $F_0, F_1, ..., F_n$ taken by a steady camera. The proposed algorithm is described in the following steps:

1. If $i \neq 0$ then $ROI_i = F_i - F_{i-1}$
2. Calculate the centroid c_i of the ROI if any
3. Calculate the wavelet decomposition W_i of F_i using the LWT.
4. W_i is quantized into integers generating the new coefficients set W_i^q
5. FVHT is applied to W_i^q and outputs the resultant bit stream
6. Return to step 1 until no more frames are left.

The first step of the proposed algorithm obtains the absolute difference between the last frame and the current one for motion detection, other algorithms may also be considered if better accuracy or motion compensation is needed. Then, if a moving object is detected by the pixel difference, the centroid and size is estimated. This will be the ROI of the frame determined by step 2. The next step calculates the LWT wavelet coefficients of the current frame. The LWT is used because of its low computational complexity in order to meet a good overall performance. The following step will quantize the wavelet coefficients as integers with a fixed q. Then, FVHT is applied to the quantized coefficients. Note that the FVHT algorithm is a proposed modified version of SPIHT that allows fovea regions.

The proposed algorithm is intended for a DVC scheme [7,8] where no feedback is allowed from the decoder or for IP cameras. Over DVC without feedback, exploiting motion compensation error residual is unreliable because the displaced frame difference cannot be calculated either by the coder or the decoder. Without displaced frame difference, an accumulative error is added over time on the decoding step reducing the quality of the decoded frames over time [6]. So, time correlation is not included in the proposed algorithm. However, if feedback is possible, the proposed algorithm can be easily implemented over a classic video scheme.

5.1 Fovea Hierarchical Trees

As stated previously, Fovea Hierarchical Trees is a modified version of SPIHT that compress using fovea regions. The algorithm is fed with the coefficients, ROI centroid and radius, fovea decaying length and a monotonic increasing function $g(x)$ with $g : R \to (b, L]$ with b as the lowest bit rate and L is the highest bit rate of the compression that defines how the compression bit rate will increase as the pixels move farther from the ROI centroid, the resultant decompressed image will have Lbpp.

On each pass of the algorithm, a distance function $D(i, j)$ of each coefficient position is evaluated and used to determine the bit rate of encoding using the decaying function $g(D(i, j))$. If the current bit rate is lower than $g(D(i, j))$ then the coefficient is encoded, otherwise it is discarded. Each quadtree will be evaluated besides the distance of the root 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 will have 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 are discarded from the LSP. The list LIS will remain the same as in the SPIHT algorithm.

6 Results

The proposed approach was implemented with a logarithmic decaying function and an arbitrary uniform scalar quantization $\delta = 0.01$ and a biorthogonal wavelet 9/7 with five levels of decomposition using the LWT as in the standard JPEG2000 which is considered as benchmark [18], other wavelets may also be considered. In figure 4 is shown a comparative analysis of a frame of the video sequence “walk” compressed using the SPIHT and the FVHT algorithms with different bit rates.

Figure 4a is the original frame with the fovea area marked with a white circle, figure 4b, 4c and 4d are the frame compressed with SPIHT at 0.06bpp ratio, 1bpp ratio and 3bpp ratio respectively. Figure 4e is the frame compressed with FVHT with a logarithmic decaying function from 0.06 bpp to 1 bpp with a centroid in $(256, 256)$ and a ROI area of 50px and a fovea decaying of 50px. Figure 4f is the frame compressed with FVHT with a logarithmic decaying function

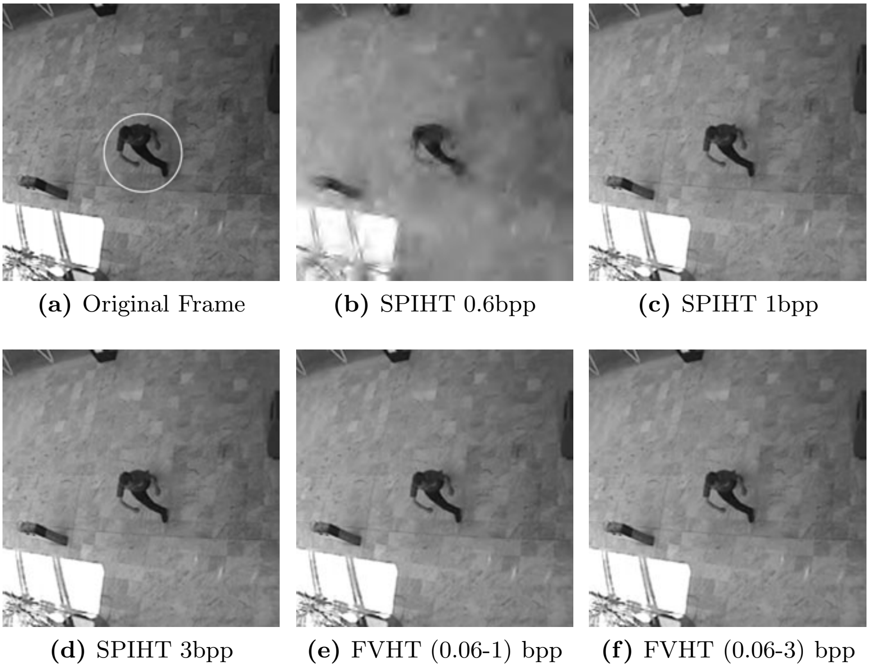


Fig. 4. Comparison of a frame from "walk" sequence with two compression algorithms (SPIHT and FVHT) and several bpp

Table 1. PSNR comparison of different areas of the frame walk with two compression algorithms (SPIHT and FVHT) and several bpp

Algorithm	bpp	25x25 px	50x50 px	75x75 px	100x100 px	125x125 px
SPIHT	3bpp	48.5818	49.0957	49.9044	50.0628	51.1757
SPIHT	1bpp	41.3977	42.1692	43.3924	43.6584	44.8385
FVHT	0.06-3bpp	41.5309	42.2027	43.4104	43.6687	44.8451
FVHT	0.06-1bpp	41.3977	42.1692	43.3924	43.6584	44.8385

from 0.06 bpp to 3 bpp with a centroid in $(256, 256)$ and a ROI area of 50px and a fovea decaying of 50px. Figures 4e and 4f present better detail in objects inside the defined ROI area like the floor texture. The same texture is less detailed as the pixels are farther from the ROI, while the images compressed with SPIHT, 4b, 4c and 4d, presents the same level of details over all. In table 1 a comparative of peak to noise signal ratio (PSNR) for the different compressions used in figure 4 is shown.

Table 1 shows the PSNR of the different compression methods and bpp against the original video frame at different steps. The first step takes a square of the decompressed frame of 25×25 pixels dimension with centroid at the fovea center

(256, 256) and compares it with a same squared region of the original video frame. Each step increases the dimensions of the compared region by 25. The proposed method shows a lower PSNR than the classic SPIHT compression. This is due to the fact that PSNR penalizes heavily the compression resolution drop from the fovea. However, PSNR cannot reflect the antialiasing effect that occurs on the human eye with a fovea compression [1]. Such antialising can be perceived in figure 4 as an increased image quality when examining it directly with the eyes fixed at the fovea center.

7 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 [2] by labeling beforehand each coefficient and using unbalanced quadtrees, however it will decrease the memory performance of the algorithm. Future work will be focused on defining and evaluating different distance functions and estimation of individual bpp on different regions of the foveated image as well as determining a better metric for measuring the quality of a foveated compression method. Other compression algorithms may also be considered for reducing the high memory usage of the reported algorithms.

References

1. Chang, E.C., Yap, C.K.: Wavelet Approach to Foveating Images. In: Proceedings of the thirteenth annual symposium on Computational geometry - SCG 1997, pp. 397–399 (1997)
2. Cuhadar, A., Tasdoken, S.: Multiple arbitrary shape ROI coding with zerotree based wavelet coders. In: Proceedings of the IEEE International Conference on Multimedia and Expo, ICME 2003, pp. 157–160. IEEE, Los Alamitos (2003)
3. Galan-Hernandez, J.C., Alarcon-Aquino, V., Starostenko, O., Ramirez-Cortes, J.M.: DWT Foveation-Based Multiresolution Compression Algorithm. Research in Computing Science, 197–206 (2010)
4. Park, K.-H., Park, H.W.: Region-of-interest coding based on set partitioning in hierarchical trees. IEEE Transactions on Circuits and Systems for Video Technology 12(2), 106–113 (2002)
5. Bovik, A.: The Essential Guide to Video Processing. Academic Press, London (2009)
6. Hanzo, L., Cherriman, P.J., Streit, J.: Video Compression and Communications. John Wiley & Sons, Ltd, Chichester (2007)
7. Girod, B., Aaron, A., Rane, S., Rebollo-Monedero, D.: Distributed Video Coding. Proceedings of the IEEE 93, 71–83 (2005)
8. Martinez, J.L., Weerakkody, W.A.R.J., Fernando, W.A.C., Fernandez-Escribano, G., Kalva, H., Garrido, A.: Distributed Video Coding using Turbo Trellis Coded Modulation. The Visual Computer 25(1), 69–82 (2008)
9. Silverstein, L.D.: Foundations of Vision. Color Research & Application 21, 142–144 (2008)

10. Ciocoiu, I.B.: ECG signal compression using 2D wavelet foveation. In: Proceedings of the 2009 International Conference on Hybrid Information Technology - ICHIT 2009, vol. 13, pp. 576–580 (2009)
11. Bovik, A.C.: Fast algorithms for foveated video processing. IEEE Transactions on Circuits and Systems for Video Technology 13(2), 149–162 (2003)
12. Galan-Hernandez, J.C., Alarcon-Aquino, V., Starostenko, O., Ramirez-Cortes, J.M.: Wavelet-Based Foveated Compression Algorithm for Real-Time Video Processing. In: Robotics and Automotive Mechanics Conference 2010 IEEE Electronics, september 2010, pp. 405–410. IEEE, Los Alamitos (2010)
13. Said, A., Pearlman, W.: A new, fast, and efficient image codec based on set partitioning in hierarchical trees. IEEE Transactions on Circuits and Systems for Video Technology 6(3), 243–250 (1996)
14. Tsai, P.: Tree Structure Based Data Hiding for Progressive Transmission Images A Review of Related Works. Fundamenta Informaticae 98, 257–275 (2010)
15. Shapiro, J.: Embedded image coding using zerotrees of wavelet coefficients. IEEE Transactions on Signal Processing 41, 3445–3462 (1993)
16. Sweldens, W.: The Lifting Scheme: A Custom-Design Construction of Biorthogonal Wavelets. Applied and Computational Harmonic Analysis 3, 186–200 (1996)
17. Mallat, S.: A Wavelet Tour of Signal Processing:The Sparse Way, 3rd edn. Academic Press, London (2008)
18. Acharya, T., Tsai, P.S.: JPEG2000 Standard for Image Compression. John Wiley & Sons, Inc., Hoboken (2004)