

# We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

185,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index  
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?  
Contact [book.department@intechopen.com](mailto:book.department@intechopen.com)

Numbers displayed above are based on latest data collected.  
For more information visit [www.intechopen.com](http://www.intechopen.com)



# Many-Core Algorithm of the Embedded Zerotree Wavelet Encoder

*Jesús Antonio Alvarez-Cedillo, Teodoro Alvarez-Sanchez, Mario Aguilar-Fernandez and Jacobo Sandoval-Gutierrez*

## Abstract

In the literature, the image compression was implemented using a variety of algorithms; such as vector quantization and subband coding and transform-based schemes. The current problem is that the selection of an image compression algorithm depends on criteria of compression ratio, but the quality of reconstructed images depends on the technology used. Some papers about of the wavelet transform-based coding show this field as an emerging option for image compression with high coding efficiency. It is well known that the new wavelet-based image compression scheme JPEG-2000 has been standardized. This chapter shows the developed novel algorithm executed in parallel using the embedded Zerotree wavelet coding scheme, in which the programs integrate parallelism techniques to be implemented and executed in the many-core system Epiphany III.

**Keywords:** image compression, vector quantization, subband coding, embedded systems, Zerotree wavelet

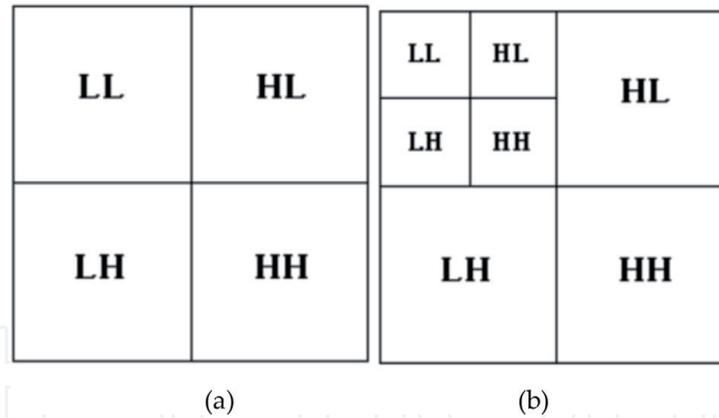
## 1. Introduction

Wavelet transform consists in uses a compact multi-resolution representation of an image, allows uses the energy compaction to exploit redundancy and to achieve compression [1].

The discrete wavelet transform (DWT) usually was used in the literature using a two-channel wavelet filter bank in a recursive process [2].

A two-dimensional image of the DWT type is usually calculated using a separable approach, this consists of scanning the input image in a horizontal direction, and passing it through the decomposition filters passes low and passes high [3]. All the selected data are subsampled vertically in order to classify the low frequency and high-frequency data in the horizontal direction. The result produces output data which is scanned vertically; the filters are repeated to generate characteristic frequency subbands [4]. After the subsampling stage, the transformation generates four LL, LH, HL and HH subbands, where each image represents 25% of the original image size [5–8].

In this particular case, the energy was concentrated in the low-frequency LL subband, so it represents a low-resolution version of the original image. The most frequent subbands contain very detailed information in three directions (horizontal, vertical and diagonal).



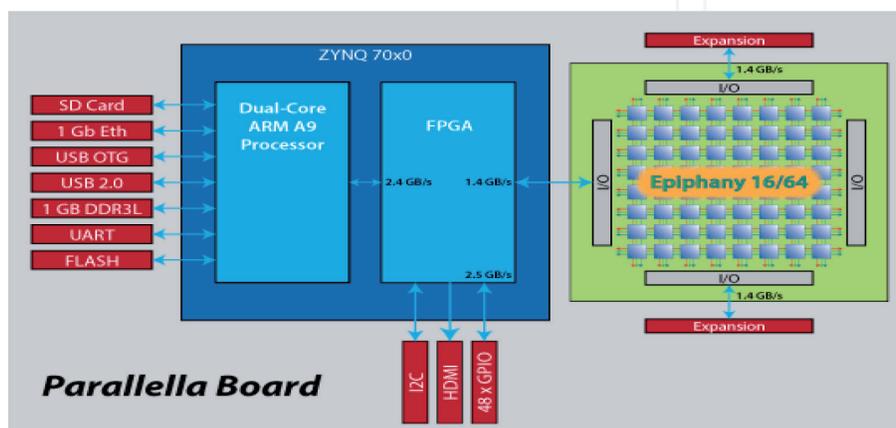
**Figure 1.** (a) Wavelet decomposition in subbands and (b) higher and lower level coefficients for an original image.

At the end of the process, the image was decomposed by applying the 2-D DWT algorithm to the LL subband [9]. With this iterative process, multiple levels of transformation will be generated where the energy is fully compacted and represented with few low-frequency coefficients [10]. The above can be seen in **Figure 1(a)** and **(b)**. **Figure 1** shows an example of decomposition of three levels of an image using the wavelet transformation, and both images defined the parent/child relationship between levels.

## 2. Many-core technology

Epiphany III is a low-cost embedded system formed with the main memory and 16 cores distributed in a mesh, as shown in **Figure 2**. The system is characterized by having low energy consumption and a high level of parallelism, concurrency and high computing power. All these features and in combination, allow data to be processed at different levels of software and hardware, thereby performing operations at each core [11, 12].

A multi-core system needs a shared memory space that consists of 232 bytes for this system. The memory addresses for access are accessed as unsigned numbers, from 0 to 232, together, they represent 230 of 32-bit words in which any core has access concurrently through a 2D mesh type topology [13]. The use of this interconnection topology avoids overloading or blocking access to shared memory that is reported in the literature as a factor that affects the high-performance system [14–17].



**Figure 2.** Components of the Epiphany III architecture [11].

This system has an energy consumption of 1 W per 50 Gigafllops in simple precision calculations; these cores measure 65 nm. This technology is RISC and can be run at a speed of 1 GHz, and the 16 cores have the same architecture [11, 14].

With this memory capacity in the Epiphany III system, it is possible to process an image of (128 × 128 pixels) that is equivalent to 16 kB of memory.

### 3. Embedded Zerotree coding

A wavelet represents a waveform of limited duration that has an average value close to zero. A wavelet transform modifies a signal from the time domain to the whole time scale domain. Wavelet coefficients are two-dimensional, given this, an image can be represented using trees due to the subsampling that is performed in the transformation.

Fourier analysis allows dividing a signal into sine waves of various frequencies, due to this the wavelet analysis is the breaking of a signal in displaced and scaled versions of the original or mother wavelet.

In zerotree coding, each wavelet coefficient of an arbitrary scale can be related to a set of coefficients in the next more excellent scale.

Zerotree root (ZTR) represents a low-scale zero value coefficient for which all larger scale coefficients have that same value. By specifying a ZTR, the encoder can track and reset all related coefficients on a larger scale.

The EZW encoder is a particular type of encoder used to encode image or sound signals of any dimension; it offers the advantages found in denoising algorithms.

When using the wavelet transform on an input image, the embedded zerotree encoding (EZW) will allow the encoder to quantify the coefficients using a binary encoding to create a representation of the image [18]. EZW uses the direct relationship between the upper and lower level coefficients (parents and children) to obtain maximum coding efficiency [19, 20].

In order to perform the EZW encoding, it is necessary to perform the following steps:

STEP 1: Determine the initial threshold using bit plane coding, where the subsequent iterations, the threshold ( $T_i$ ) is reduced by half, and the coefficients  $<2T_i$  are only encoded in each flow.

EZW involves two passes, as a recursive process:

- Dominant pass.
- Subordinate pass.

STEP 2: Two individual lists are defined:

- a. Dominant pass: Contains the coordinates of those non-significant coefficients.
- b. Subordinate pass: Contains the magnitudes of the significant coefficients for each threshold, involves the use of two passes:
  1. Key stage.
  2. Low pass.

In the dominant step, the magnitude of the wavelet coefficients is compared to an arbitrary threshold value; the essential data is determined, and the coefficients

are defined with an absolute value. The scanning was done with spatial frequencies; two bits are used to define the sign and the position of the significant coefficients. The positive significant coefficient and the non-significant coefficients are above an arbitrary threshold, usually starting with the two highest powers, two below the maximum wavelet value, where the wavelet coefficient is insignificant and has a significant descendant.

STEP 3: Dominant pass (significant pass):

Let wavelet coefficients in the dominant list are compared with  $T_i$  to determine the importance and, if significant, its sign. The resulting significance map is coded and sent by a zero tree. The inclusion of the ZTR symbol increases the coding efficiency because the encoder maximizes the correlation between image scales.

STEP 4: Four symbols are used to form a code:

1. Zerotree root.
2. Zero isolated.
3. Significant positive.
4. Significant negative

The EZW technique can be significantly improved using entropy coding as a preoccupation to achieve better compression [22].

STEP 5: Define the entropy code.

The low pass filter follows, where significant coefficients are detected and refined under the successive approach quantification (SAQ) approach [21].

STEP 6: Define refinement pass.

STEP 7: The entropy code sequence of 1 and 0 is defined, and adaptive AC is used, and send STOP.

#### 4. Proposed algorithm

The development of proposed EZW (embedded zerotree wavelet) image coding has attracted considerable attention among researchers [23–25]. It is the most popular wavelet-based compression algorithm and is widely used in several image-based applications. In this paper is used the recursive transformation method for multi-level decomposition, where the result data is then preprocessed before of the zerotree compression, the block diagram of a wavelet-based image coding algorithm is shown in **Figure 3**.

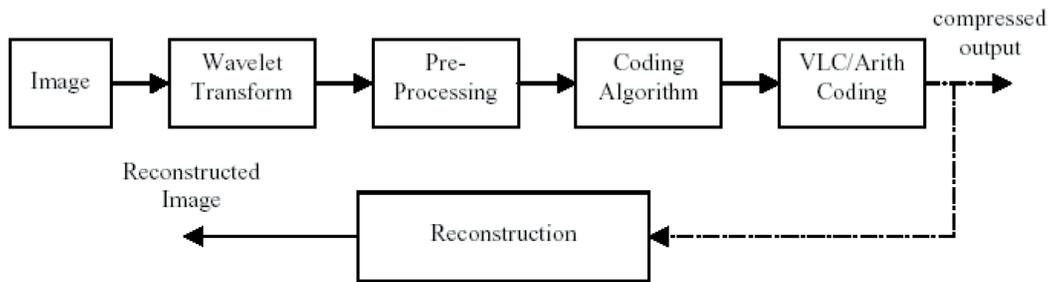
Our objective of this paper is to show our proposed algorithm to enhance the compression of an image eviting to minimal loss during reconstruction.

Our algorithm was applied as a preprocessing stage; this allows to eliminate unused data in the transformed image that is not important and significant in the reconstruction of the image.

However, it is necessary to mention that more bits are required during compression and the processing time increases so that a parallel proposal can help.

When compensation is used to a greater extent between the compression ratio and reconstruction in image quality, it is possible to eliminate irrelevant data in the image where higher compression is achieved with a slight reduction in quality [21].

In an image transformed by wavelet, the coefficients represent a low-resolution image in the LL subband. The high-frequency subbands contain subbands of specific data in each direction.



**Figure 3.**  
 Block diagram of a wavelet-based image coding algorithm.

The use of the three subbands contributes to a smaller scale within the image reconstruction process because the coefficients are mostly zero, and few large values correspond to border information and textures in the image [22].

In this article, an algorithm was proposed to reduce less essential data in order to achieve higher compression, and thus preserve high-value coefficients, thereby eliminating the lowest values of the minimum value subband.

Our proposed algorithm uses the weight calculation method to have the minimum value subband for each level. The weight of each subband is calculated by adding the absolute value in all subband coefficients. Only three subbands (LHi, HLi and HHi) with a minimum weight are seen in detail.

Eq. (1) shows it.

$$\begin{aligned}
 W_s &= \sum \text{abs}(\text{coefficients in subbands}) (\min(\text{subband}_i)) \\
 &= \text{Subband with minimum weight, at the level } I
 \end{aligned} \tag{1}$$

After finding the required subbands in each level, the algorithm reduces the depreciable data in these subbands depending on the importance of the data for preserver the reconstruction. In this age, the majority of the values are close to zero, and the coefficients have the smallest data in each subband, where it is used as a threshold to eliminate low-valued data in that subband.

The coefficients whose value is higher than a set threshold value are retained, and the value is near to zero are deleted. In our experiments, it was used different two threshold values to show the effect of compressed output and reconstructed image.

In the zerotree coding, the reduction of low valued significant coefficients in minimum weight subbands, result in higher compression ratio with a slight loss in decoded PSNR. Our results show that this algorithm shows better efficiency with a cost of negligible loss in picture quality.

## 5. Parallel implementation

In the Epiphany, III System is used a memory model and also a programming model, in a flow system that includes single instruction multiple data (SIMD), single program multiple data (SPMD), master-slave programming, multiple instruction multiple data (MIMD).

Can also be programmed, to data flow, static, dynamic, systolic array, multi-threads, message passing and sequential communication processes (SCP) [26].

First, the operating system performs a review of the hardware of the Epiphany system and then begins to configure the cores that are in the mesh-type topology and then the distributed information of the matrices A, B. This is due to the structural nature of the epiphany system, and finally the distribution of tasks in small blocks that is appropriate (**Figure 4**).

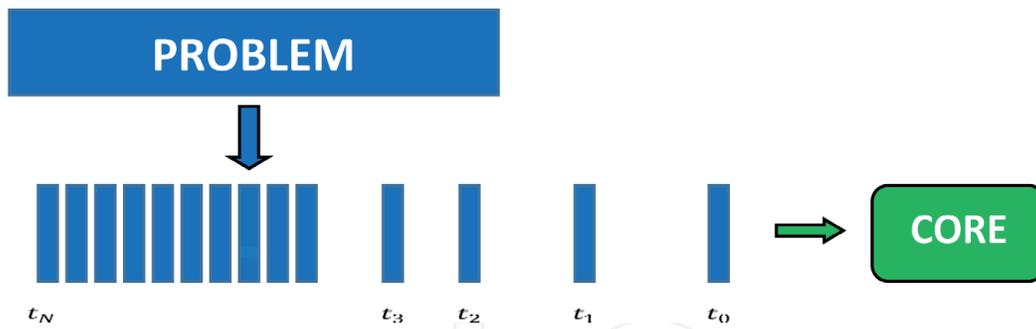


Figure 4.  
Sequential execution.

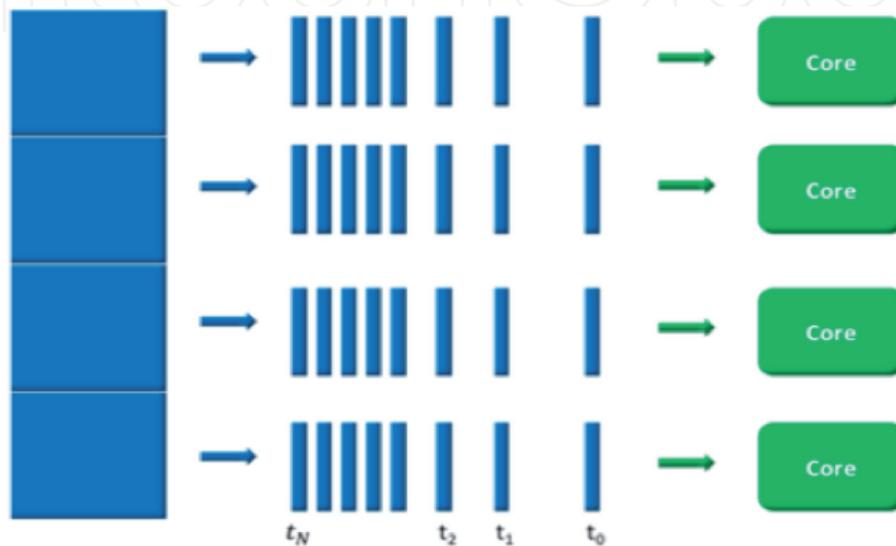


Figure 5.  
Execution SPMD.

For this distribution, the programming model single program multiple data (SPMD) is used [27], which is responsible for distributing the execution for each of the cores, as shown in **Figure 5**.

## 6. Distributions of data and tasks

Depending on the problem. It can choose between two approaches to decompositions: data decomposition or task decomposition.

Establish strategies to choose decompositions: decomposition of instructions or decomposition of tasks.

Decomposition of instructions: this refers to the distribution of instructions for the cores, which will handle the processing. Then, it describes the process of distributing the instructions that can execute in parallel on different cores.

In which the results of each one of the executions of each nucleus, of this type of executions in parallel accelerates  $n$  times faster if a single nucleus were executed. Except for the minimum delay involved in the initial distribution of workload or instructions, and the final collection of data results, resulting in linear acceleration with the number of cores.

Decomposition of tasks: this refers to the distribution of tasks for each of the cores, which will be responsible for processing tasks. That is to say that the whole program is devoted to tasks. A task is a sequence of the program that can be executed in parallel, concurrent with other tasks. This approach is beneficial when tasks maintain high levels of independence.

If each task and algorithm are different, then both can be implemented, the functional decomposition, in which the particular characteristics of each type of task or algorithms will be used to execute them in the nucleus for that purpose.

Recalling that the new technologies of embedded systems have integrated circuits with several cores, in which the cores perform threads that are the tasks that are executed in parallel, so the applications will be made in less time. This allows increasing the parallelism and the concurrence in hardware and software. It is also important to make the distribution between cores, the tasks, as well as the synchronization between the cores.

Matrices are mathematical operations that scientists use. In which, the algorithm used for matrix multiplications that run in parallel is described by IBM [28]. The importance of data communication between neighboring cores according to the Cannon algorithm [29] is also mentioned. The memory in each core represents a challenge in the implementation because it is limited, which makes it necessary to use the available memory space for communication between the cores. These are some important factors for the system dedicated to parallel processing.

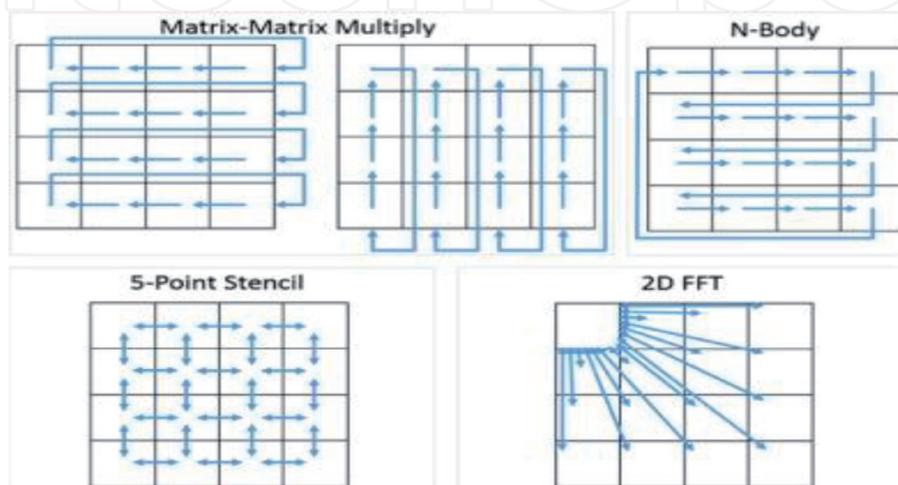
**Figure 6** shows the multiplications of matrices for each core. It also shows the sending of data for the execution of the tasks, between each core, so the Epiphany III system incorporates the mechanism, message passing, to synchronize the sending of the threads avoiding conflicts between the cores or accesses to the shared global memory.

**Figure 6** shows the data flow for matrix multiplication that runs in specific numbers of steps, this is determined by the quadratic root of  $\sqrt{P}$ . Then  $P$  is the number of cores, in which the matrix multiplication is the data set of size  $\sqrt{P} \times \sqrt{P}$ .

At each repetition, element  $C$  of the product matrix obtained, then matrix  $A$  moves down and matrix  $B$  moves to the right. This example can be programmed using the standard high-level ANSI programming language “C.” The Epiphany III system provides specific functions to simplify the programming of many cores, thanks to the open operating system, but its use is not mandatory for programmers.

The implementation of the algorithm, in the Epiphany III system with 16 cores, operating at 1 GHz, which solves matrix multiplication of  $128 \times 128$  in 2 ms. The performance of the Epiphany III system grows linearly, using appropriate programming and data distribution models, and this is seen in the cost/performance of an optimized system [11, 30–33].

To exploit the parallelism, in the hardware platform epiphany, it is encouraged that the application is broken down into tasks (code portions). Thus, each of the cores could run a program part in parallel to other cores.



**Figure 6.**  
*Data flow in matrix multiplication.*

The decomposition of tasks must be followed by a synchronization of the different parties involved to ensure data consistency.

In **Figure 6**, it represents the matrix mathematically. It also shows the simplification of matrix multiplication. This multiplication is represented by (2).

$$C_{ij} = \sum_{k=0}^{N-1} (A_{ik} B_{kj}) \quad (2)$$

The input matrices are A and B, C is the resulting matrix of  $A_i$  and  $B_j$ , in which they have as coordinate (i, j), which is (row, column), these are elements of the matrix. The procedure to program matrix multiplication in a single core is shown below.

```

1  for (i=0; i<M; i++)
2  for (i=0; j<M ;j++)
3      for (i=0; k<K; k++)
4  C[i][j] += A[i][k] * B[k][j]
5
6

```

The previous code used the standard language C, which compiles and executes in a single core.

If these matrices A, B, will proceed to run on each of the cores in the system and the result of each element of the matrix C. are placed in the local memory of each of the cores.

## 7. Experiment results

Different image outputs of the wavelet-based compression were shown in **Figure 7**.

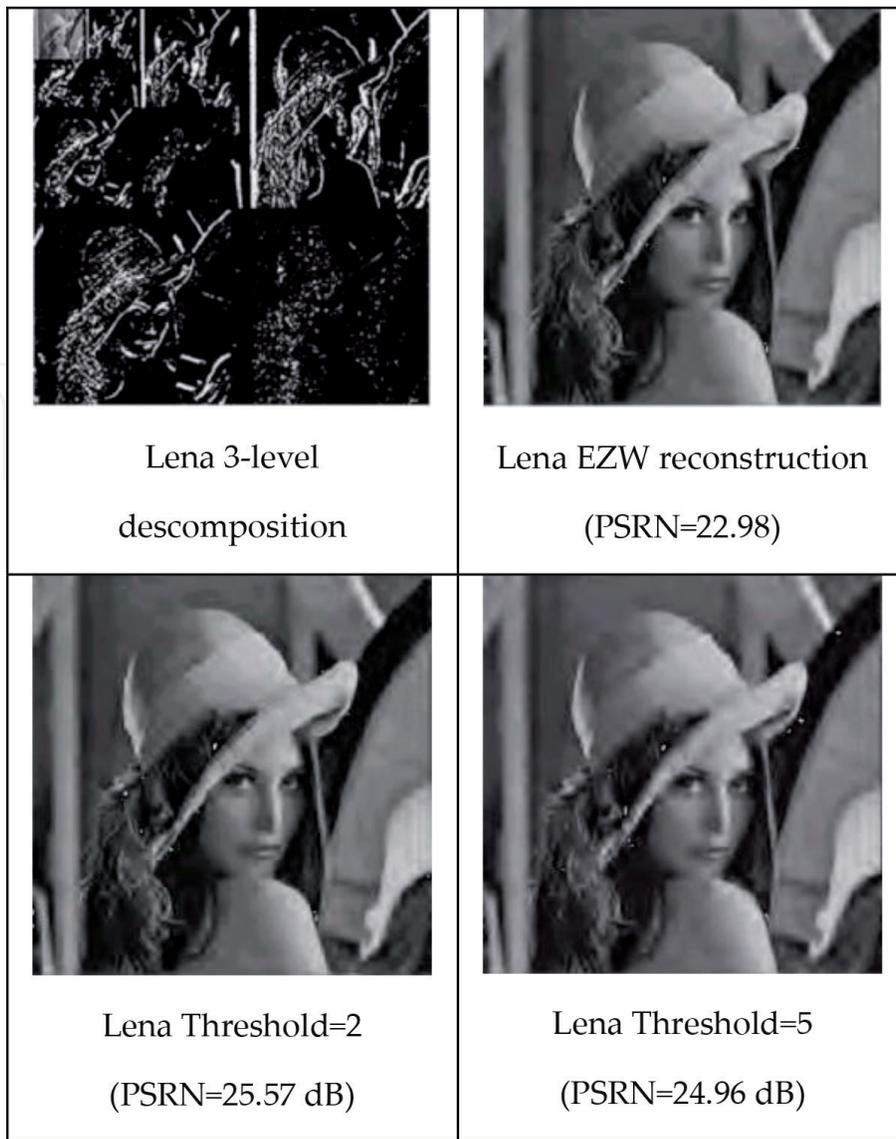
**Table 1** shows the values when was applied our proposed, **Table 2** shows the obtained values using adaptive arithmetic coding.

Image	Min weight subband			EZW method		Proposed method threshold = 2		Proposed method threshold = 5	
	L3	L2	L1	Bytes	PSNR	Bytes	PSNR	Bytes	PSNR
Lena	HH	HH	HH	63,456	22.98 dB	62,256	25.57 dB	61,465	24.96 dB

**Table 1.**  
Obtained values.

Image	Min weight subband			EZW method		Proposed method threshold = 2		Proposed method threshold = 5	
	L3	L2	L1	Bytes	PSNR	Bytes	PSNR	Bytes	PSNR
Lena	HH	HH	HH	3024	22.98 dB	2952	25.57 dB	2830	24.96 dB

**Table 2.**  
Obtained values using the adaptive arithmetic coding.



**Figure 7.**  
*Output images.*

## 8. Conclusion

The above method exploits the property of tradeoff between compression ratio and output PSNR and reduces the least essential data in order to attain further compression. The better compression ratio is achieved compared to original EZW coder after applying threshold with a slight reduction in PSNR during reconstruction.

## Acknowledgements

We appreciate the facilities granted to carry out this work to the Instituto Politécnico Nacional through the Secretariat of Research and Postgraduate with the SIP 20194986 and 20195024 projects. To the Interdisciplinary Unit of Engineering and Social and Administrative Sciences, Center for Technological Innovation and Development in Computing and Digital Technologies Research and Development Center. Likewise, to the Program of Stimulus to the Performance of the Researchers (EDI) and the Program of Stimulus COFAA.

IntechOpen

### Author details

Jesús Antonio Alvarez-Cedillo<sup>1\*</sup>, Teodoro Alvarez-Sanchez<sup>2</sup>,  
Mario Aguilar-Fernandez<sup>1</sup> and Jacobo Sandoval-Gutierrez<sup>3</sup>

1 Instituto Politécnico Nacional, UPIICSA, Iztacalco, Granjas México, Mexico

2 Instituto Politécnico Nacional, CITEDI, Mesa de Otay, Mexico

3 Universidad Autónoma Metropolitana, Lerma de Villada, Mexico

\*Address all correspondence to: jaalvarez@ipn.mx

### IntechOpen

---

© 2019 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. 

## References

- [1] Yamamoto A. Wavelet analysis: Theory and applications. Transform. 1994. DOI: 10.1051/jp1:1997114
- [2] Shensa MJ. The discrete wavelet transform: Wedding the À Troun and Mallat algorithms. IEEE Transactions on Signal Processing. 1992. DOI: 10.1109/78.157290
- [3] Tzanetakis G, Essl G, Cook P-R. Audio analysis using the discrete wavelet transform. In: Proceedings of the WSES International Conference Acoustics and Music: Theory and Applications (AMTA 2001). 2001
- [4] Filters W, Transforms W. Preview of Wavelets, Wavelet Filters, and Wavelet Transforms. Space & Signals Technologies LLC; 2009
- [5] Misiti M, Misiti Y, Oppenheim G, Poggi JM. Wavelets and Their Applications. 2010. DOI: 10.1002/9780470612491
- [6] Letelier JC, Weber PP. Spike sorting based on discrete wavelet transform coefficients. Journal of Neuroscience Methods. 2000. DOI: 10.1016/S0165-0270(00)00250-8
- [7] Nason GP, Silverman BW. The discrete wavelet transform in S. Journal of Computational and Graphical Statistics. 1994. DOI: 10.1080/10618600.1994.10474637
- [8] Goodman RW. Discrete wavelet transforms. In: Discrete Fourier and Wavelet Transforms. 2016. DOI: 10.1142/9789814725781\_0003
- [9] Tzanetakis G, Essl G, Cook P-R. Audio analysis using the discrete wavelet transform. In: Proceedings of the WSES International Conference Acoustics and Music: Theory and Applications (AMTA 2001). 2001
- [10] Mitra SK. Digital signal processing—Computer-based approach. Microelectronics Journal. 2001. Sanjit K. Mitra.pdf. DOI: 10.1016/S0026-2692(98)00072-X
- [11] Alvarez-Sanchez T, Alvarez-Cedillo JA, Sandoval-Gutierrez J. Many-core parallel algorithm to correct the gaussian noise of an image. In: Communications in Computer and Information Science. Vol. 948. 2019. DOI: 10.1007/978-3-030-10448-1\_7
- [12] Vajda A. Programming Many-Core Chips. 2011. DOI: 10.1007/978-1-4419-9739-5
- [13] Boyd-Wickizer S, Clements A, Mao Y, Pesterev A, Kaashoek M, Morris R, et al. An Analysis of Linux Scalability to Many Cores. MIT Web Domain; 2010
- [14] Hibbs AE, Thompson KG, French D, Wrigley A, Spears I. Optimizing performance by improving core stability and core strength. Sports Medicine. 2008. DOI: 10.2165/00007256-200838120-00004
- [15] Manferdelli JL, Govindaraju NK, Crall C. Challenges and opportunities in many-core computing. Proceedings of the IEEE. 2008. DOI: 10.1109/JPROC.2008.917730
- [16] Bates D. Introduction to the Matrix Package. R Core Development Group; 2012. DOI: 10.1016/j.amjmed.2014.09.011
- [17] Guz Z, Bolotin E, Keidar I, Kolodny A, Mendelson A, Weiser UC. Many-core vs. many-thread machines: Stay away from the valley. IEEE Computer Architecture Letters. 2009. DOI: 10.1109/L-CA.2009.4
- [18] Shapiro JM. Embedded image coding using Zerotrees of wavelet

coefficients. IEEE Transactions on Signal Processing. 1993. DOI: 10.1109/78.258085

[19] Said A, Pearlman WA. A new, fast, and efficient image codec based on set partitioning in hierarchical trees. IEEE Transactions on Circuits and Systems for Video Technology. 1996. DOI: 10.1109/76.499834

[20] Hong ES, Ladner RE. Group testing for image compression. IEEE Transactions on Image Processing. 2002. DOI: 10.1109/TIP.2002.801124

[21] Amit Y, Geman D. Shape quantization and recognition with randomized trees. Neural Computation. 1997. DOI: 10.1162/neco.1997.9.7.1545

[22] Blanco-Velasco M, Cruz-Roldán F, Moreno-Martínez E, Godino-Llorente JI, Barner KE. Embedded filter bank-based algorithm for ECG compression. Signal Processing. 2008. DOI: 10.1016/j.sigpro.2007.12.006

[23] George R, Manimekalai MAP. A novel approach for image compression using zero tree coding. In: 2014 International Conference on Electronics and Communication Systems, ICECS 2014. 2014. DOI: 10.1109/ECS.2014.6892611

[24] Zhou J, Huang PS, Chiang F-P. Wavelet-based pavement distress image compression and noise reduction. Wavelets XI. 2005. DOI: 10.1117/12.612926

[25] Shen K, Delp EJ. Wavelet based rate scalable video compression. IEEE Transactions on Circuits and Systems for Video Technology. 1999. DOI: 10.1109/76.744279

[26] Duncan R. A survey of parallel computer architectures. Computer. 1990. DOI: 10.1109/2.44900

[27] Omiecinski E. Highly parallel computing. Information and

Software Technology. 2003. DOI: 10.1016/0950-5849(90)90035-p

[28] Lord N, Golub GH, Van Loan CF. Matrix computations. The Mathematical Gazette. 2007. DOI: 10.2307/3621013

[29] Williams S, Oliker L, Vuduc R, Shalf J, Yelick K, Demmel J. Optimization of sparse matrix-vector multiplication on emerging multicore platforms. Parallel Computing. 2009. DOI: 10.1016/j.parco.2008.12.006

[30] Haibo SB, Rong C, Yandong C, Kaashoek F, Morris R, Pesterev A, et al. Corey: An Operating System for Many Cores. OSDI '08; 2008

[31] Guz Z, Bolotin E, Keidar I, Kolodny A, Mendelson A, Weiser UC. Many-core vs. many-thread machines: Stay away from the valley. IEEE Computer Architecture Letters. 2009. DOI: 10.1109/L-CA.2009.4

[32] Diaz J, Muñoz-Caro C, Niño A. A survey of parallel programming models and tools in the multi and many-core era. IEEE Transactions on Parallel and Distributed Systems. 2012. DOI: 10.1109/TPDS.2011.308

[33] Hill MD, Marty MR. Amdahl's law in the multicore era. Computer. 2008. DOI: 10.1109/MC.2008.209