Reduced memory wavelet transform coding using post-processing for SPIHT algorithm

Roger F. Larico Chavez, Yuzo Iano, Osamu Saotome, Rangel Arthur and Rogerio Seiji Higa

***Abstract*— This paper proposes an image compression scheme using a personalized storage Discrete Wavelet Transform (DWT). In image compression schemes based on DWT, the module that generates these wavelet coefficients is sequentially attached to some encoding bitplanes. As the level of DWT decomposition increases the quantity of bits required to represent the wavelet coefficients is increased. A significant amount of memory is required to store these coefficients especially when the level of decomposition of DWT is high. In this paper, a post-processing method is proposed to set the amplitude of the variable coefficients. This is accomplished, depending on the level of the coefficient and the planes of most significant bits of the last levels can be used to store other bitplanes from other levels. The results show a significant reduction in memory consumption for processing the algorithm that uses SPIHT wavelet decomposition characteristics and a post-processing.**

***Index Terms*— Wavelet transform, reduce memory, image compression, processing, SPIHT.**

I. INTRODUCTION

Nowadays, the discrete wavelet transform (DWT) represents an important tool for compression of multimedia signals. DWT allows to efficiently represent the high frequency components in images, achieving high compression ratios when combined with sophisticated algorithms such as EZW (Embedded Zerotree Wavelet coding), SPIHT (Set Partitioning in Hierarchical Trees) [1], JPEG-2000 [2], and the recommendation CCSDS image compression (The Consultative Committee for Space Data Systems) [3, 4].

The DWT is applied in image fusion [5] as a tool for a specific processing in multimedia signals; it is also used on network devices [6], in image recognition [7] and other applications [8, 9]. These applications can be implemented in embedded systems, including image compression that requires a considerable processing for storage. In this work, our focus is on reducing the amount of memory during the processing of WT coding using SPIHT. The DWT can be represented to *M* bits per coefficient. In Fig. 1 we can observe the traditional way in which the value of *M* can be reduced for use by encoding bit planes. The most significant bit (*MSB*) of each plane are coded according to a compression scheme (the criterion of significance) until reaching a rate (lossy $N < M$) or scanning of all bit planes (lossless $N = M$).

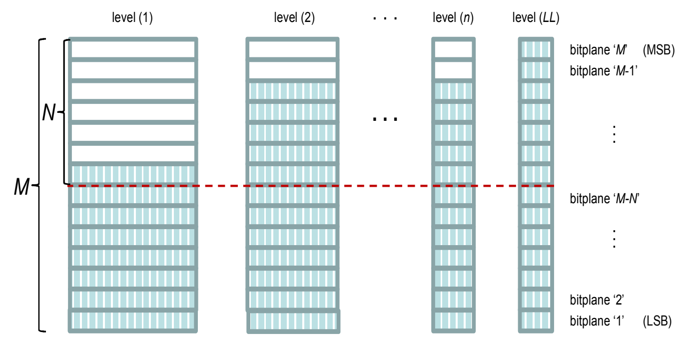

Fig. 1. Planes of bits used for each level in the DWT.

As shown in Fig. 1, it is possible to use only the first *N* most significant bits (*MSB*) of *M* bits from DWT and thus apply bitplane processing algorithms, depending on the application. In this case, the characteristic of DWT to concentrate energy into the LL_i coefficients can be used, which represent an approximation of the image. The other subbands (HL_i, LH_i, HH_i) represent the details of the signal for each level.

In the case of SPIHT compression algorithm, it is possible to use this technique [10] as shown in Fig. 1. DWT can be processed using SPIHT which cuts the least significant bits. The SPIHT algorithm progressively processes the more significant planes.

The state of the art research have recently tried more efficient solutions to the problem of memory-constrain in the development of hardware encoder bitplanes. One alternative is to reduce the amount of memory, reducing the number of lists [11]. Other fronts seek to modify the SPIHT coder [12] or using other methods. Alternatively, the modulus of the DWT SPIHT algorithm can be modified to reduce memory usage. In order to reduce memory, DWT implementations use methods for calculating the transform recursively [13] or methods based on line and also calculating coefficient by coefficient (line-based) [14]. These methods reduce the necessary memory usage of module DWT emphasizing the method associated

Manuscript received June 10, 2012. This work was supported in part by the Itasat Project (University of Campinas and Technological Institute of Aeronautics).

R. Larico Chavez, Y. Iano and R. S. Higa are with the School of Electrical and Computer Engineering, University of Campinas, SP, Brazil (e-mail: rlarico@decom.fee.unicamp.br, yuzo@decom.fee.unicamp.br, rhiga@decom.fee. unicamp.br).

R. Arthur is with the School of Technology, University of Campinas, SP, Brazil (e-mail: rangel@ft.unicamp.br).

O. Saotome with the Institute Technological of Aeronautic, SP, Brazil (e-mail: osaotome@ ita.br).

with the calculation. In [15] it is used post-processing of DWT amplitudes reducing the amount of bits to represent the coefficients. The previous techniques have to store the result inside the encoder anyway, the main idea is to diminish the required space without affecting the other modules of the device.

In this paper we propose the coding scheme shown in Fig. 2, which is an alternative way of storing the DWT coefficients inside the encoder. A post-processing is applied to the DWT, which consists of a reordering and an interface (Part II). After that the SPIHT traditional algorithm can be applied for image compression (Part III).

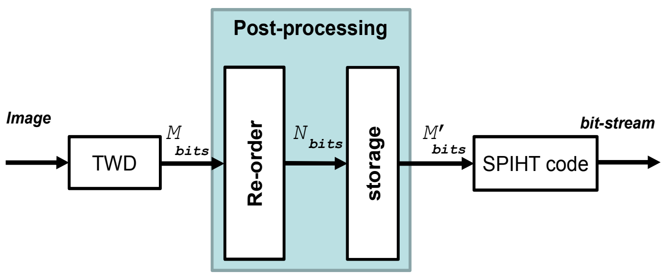

Fig. 2. Coding scheme using the SPIHT and the proposed post processing which decreases the number of bits required for each bit plane of M by N bits.

II. WAVELET TRANSFORM WITH POST-PROCESSING

Coding that uses the bitplanes methodology like the SPIHT algorithm delivers an efficient and good compression. It also allows the progressive transmission and low processing complexity [1].

For coding SPIHT, the wavelet transform coefficients must be stored in a memory bank for processing. This is because the coefficients are first checked for significance (access to the most significant bit planes) and later refined in each of the coding steps. This would require a large amount of memory space, especially when the image size is large and the decomposition of the DWT is a high level [16].

In [15] the storage was performed until a specific bitplane, the other bitplanes, the least significant ones, were approximated giving good results. The used bitplane format was a standard unmodified bitplane. This could also be applied in this proposal, but here the idea. In our proposal the caractheristics of the DWT like energy compaction and decay of the coefficients by level are explored. For example, more bits are required in the highest level to represent a coefficient than in the lowest level, where the number of bits needed are the lowest.

Table 1 shows the number of bits required for each coefficient at every level of DWT for a set of eight images (Airplane, Baboon, Lenna, Barbara, Goldhill, Peppers, Sailboat and Satellite). The DWT is biorthogonal (bior4.4) and only the integer part is used. The calculation was done in a Matlab module as informative tool.

In Table 1, the forth column corresponds to the mean of bits necessary to represent a coefficient in a given subband. This is

TABLE I
NUMBER OF BITS REQUIRED TO REPRESENT EACH COEFFICIENT IN THE DWT LEVELS

Level DWT	Subband DWT	Size	Mean bits *	Max bits *
Level 6	LL_6	8 × 8	15	15
Level 6	HL_6, LH_6, HH_6	8 × 80	11	13
Level 5	HL_5, LH_5, HH_5	16 × 16	9	13
Level 4	HL_4, LH_4, HH_4	32 × 32	8	12
Level 3	HL_3, LH_3, HH_3	64 × 64	7	11
Level 2	HL_2, LH_2, HH_2	128 × 128	6	10
Level 1	HL_1, LH_1, HH_1	256 × 256	5	9

* Mean and maximum of bits used per wavelet coefficient.

valid because there are many coefficients at each level, and the overall average value includes many values that are near zero and few high values or peaks that do not affect this average [16]. The last column of Table 1 has the maximum number of bits required to represent all the coefficients of the respective subband. It is observed that with increasing levels of DWT decomposition, there is a increase in the number of bits required to represent the coefficients [17]. It is shown the highest rates are located in the region LL of higher level (characteristic of energy concentration), and the number of bits needed to represent is 15 for this test group.

Thus, the implementation of traditional SPIHT bitplanes encoder requires a minimum number of variable bits for each subband. In this case, DWT requires 512 × 512 coefficients of 15 bits each, which requires a large amount of memory storage. Using 15 bits for every coefficient, for example, in the subband HL_1, LH_1, HH_1 the first level consisting of 3/4 parts of the entire array, the actual usage is only 9 bits at maximum. Thus, 6 bits per coefficient to these three regions are wasted. For the next levels, a similar behavior appears, totaling approximately 37.78% allocation of unused memory (last column of Table 1).

The Fig. 3 shows the idea of obtaining the positions of MSB_i (see eq. 2) bit planes for each i-th level (step 1). It is observed a different position for each level, setting the position vector *vMSB* (signaled by the red arrow).

$$vMSB = \{MSB_1, ..., MSB_i, ..., MSB_n, MSB_{LL}\} \quad (1)$$

In the representation of each coefficient, there is a sign bit (in this representation it is the most significant coefficient). In this Fig. 3, note that in LL_n, the signal value is known (it is

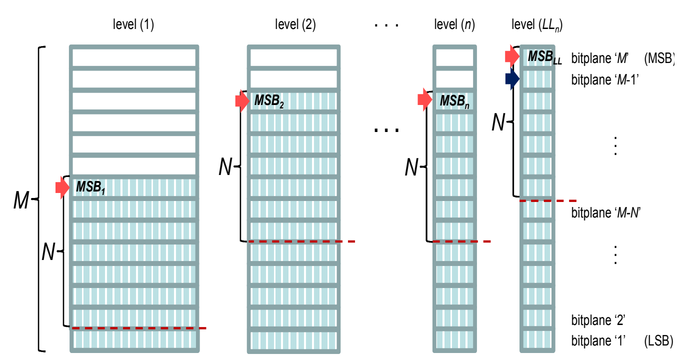

Fig 3. Vector to indicate the MSB for each subband.

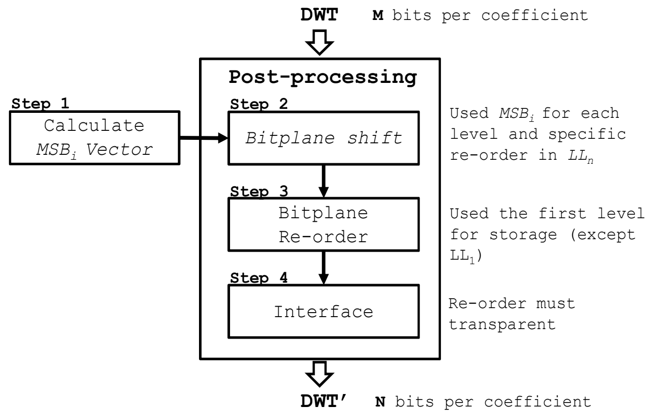

Fig 4. Flowchart of the proposed wavelet transform with post-processing.

always positive for an image approximation). Thus, the bit plane MSB_{LL} - *1* (blue arrow) is the most significant in LL_n.

In order to reduce the memory needed to store the coefficients of the DWT it is proposed a method of post-processing to calculate the new amplitudes of the coefficients (step 2). A rearrangement of a level 1 bit plane (that uses less bits) is used to compensate other higher level, so the storage size is equivalent to N bits (step 3). This is possible since the area of level *'n'* equals three times the area of level *'n+1'*. So, a bit plane level 1 (without LL) equals three bit planes for the remaining levels. Finally, the interface considers as zero the part that cannot be saved (step 4).

A. *Steps of post-processing*

The post-processing is schematically suggested in Fig. 4. The four steps are explained and exemplified below.

The first step is to get the vector of MSB_i. This vector is calculated with a subset criterion as seen in the last two columns of Table 1 or by training using other techniques. In this proposal a tool was created which calculates and provides information such as those presented in Table 1. Also, if there is a proper control of overflow, it is possible to use a weighted average of the mean bit (second last column) and the maximum number of bits (last column).

The second step, shown in Fig. 5, corresponds to

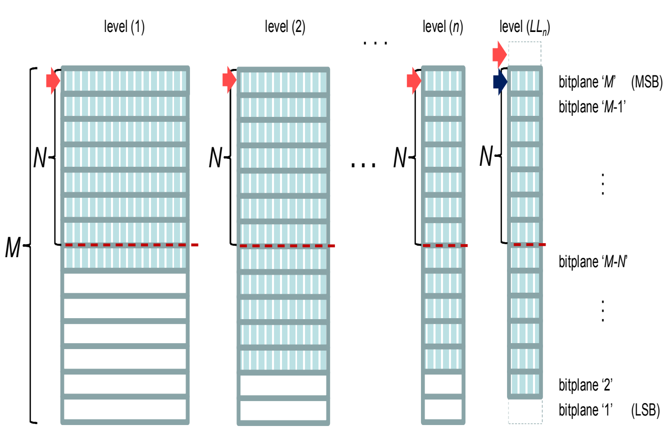

Fig. 5. Reordering step of using the vector *vMSB* from the proposed post-

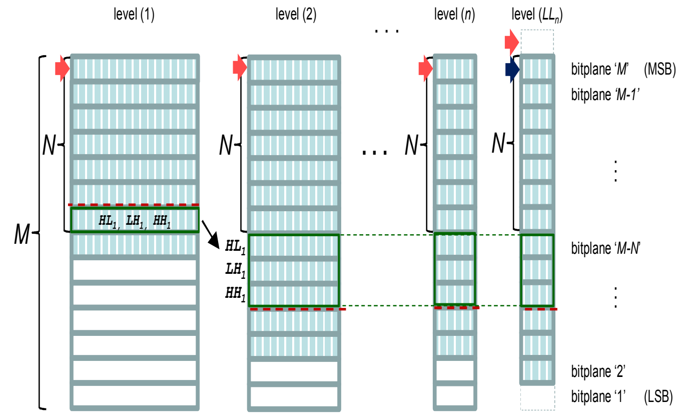

Fig 6. Step to use the latest plane required (level 1) to store bit planes from the other DWT levels.

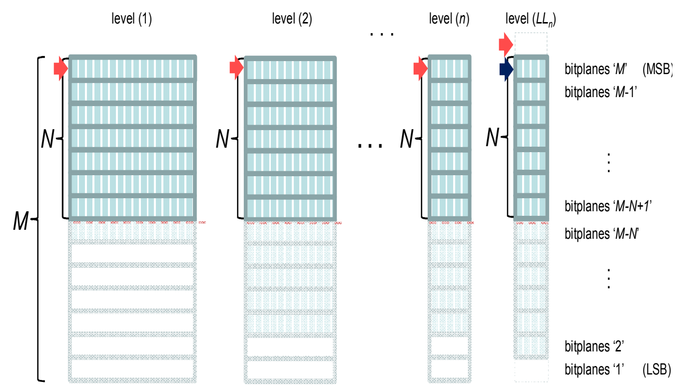

Fig. 7. Representation of the physical memory used: N of M-bits, after post-processing.

reorganization (shift) from each level. This operation is intended to allow that the most significant bit MSB_i corresponds to the bit plane M at all levels. Specifically, the reordering of bit planes in the region LL_n corresponds to $MSB_{LL} - 1$. The plane MSB_{LL} is not considered because it does not change in this representation (the signal is always positive). Thus, the vector *vMSB* is a reference of the new order (fixed).

The third step shown in Fig. 6 uses a bit plane M-N $+1$ of level 1 to save in that region the bit planes *M-N, MN-1, M-N-2* of the other levels and LL_n. Each quadrant HL_1, LH_1, HH_1 of level 1 is then the data bits of the respective planes.

After this step (step 3) the physical memory is already reduced as they are using only N bits per coefficient (equivalent). In Fig. 7, it is observed the memory after these steps, using M-N $+1$ to M bit planes. The rest can be used for other purposes, unallocated or simply released.

In the fourth step (Fig. 8), the new regulations must be transparent to applications. Thus, the storage interface shows the N-bit physical and virtual M_v bits with zeroed bit planes (according to the preceding steps and the vector *vMSB*).

To represent a wavelet coefficient at any level i, N physical bits were used and M_v virtual bits were retrieved. From the M_v= M virtual bits provided by interface the least significant of each level are normally lost (see "zeroed" on Fig. 8). In the specific case of level 1, only the N-bit planes of that level are saved. For other levels, i=*{2:n}*, N+*3* bits are always saved

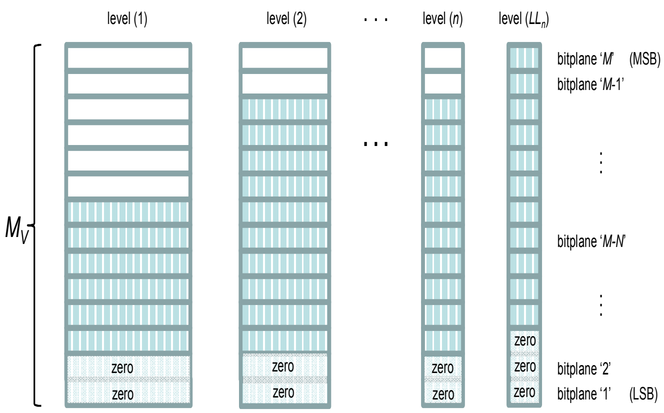

Fig. 8. Memory interpreted by the interface in the fourth step of the proposal.

and for the region LL_n $N+4$ bits are saved.

III. SPIHT CODING

The SPIHT coder has an algorithm that explores the similarities between subbands in wavelet decomposition of an image. Firstly, the algorithm uses the coefficients considered more important. Therefore, generates a bitstream from the bits of these coefficients, refined step by step. Thus, it is possible to get the original image progressively. This method uses encoding of bit planes.

This work uses the traditional SPIHT [1] implementation where DWT uses M bits to represent the coefficient. The SPIHT algorithm is detailed in [1, 10, 16]. Basically, the SPIHT encoder uses a partitioning of trees in order to maintain the insignificant wavelet coefficients grouped into best larger subsets [18]. In coding, a coefficient is considered significant if its value is greater than or equal to the threshold T, or as insignificant if its value is less than T. There are two steps in the coding of the SPIHT, sorting and refinement step. The general diagram, with emphasis on access to memory, is shown in Fig. 9.

The traditional bitplanes coding requires the array of DWT to be calculated and stored in a memory for SPIHT encoding.

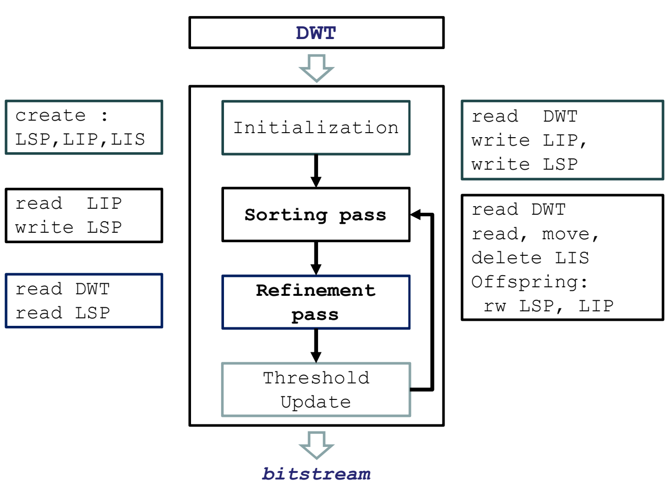

Fig. 9. Block diagram of the SPIHT algorithm emphasizing memory access.

This requires a large memory space that is only used for reading. As shown in Fig. 8, the memory access of the DWT can be checked in the algorithm in [1] as a feature this encoder can access a bitplane step-by-step instead of the full coefficient. Because of this behavior, segmentation produced by the non-sequential coefficient proposal is not a problem for this type of scheme.

The strip-SPIHT coding in [19] shows an implementation that uses little memory for the SPIHT coding. It stores a few lines of wavelet coefficients in a strip-buffer and then the SPIHT encoding is made in a strip-base form, calculating part of DWT and generating the SPIHT bitstream. In the same area of research, the work published in [20], which uses lower levels of decomposition DWT in conjunction with a new tree structure SOT-C, managed to further reduce the memory required for the coding scheme in SPIHT-based strip. The published work [20] uses a specific encoding module in the DWT making a coding for each subband to reconstruct the coefficient, by adding the dequantization value ξ [15, 10].

IV. SIMULATIONS AND DISCUSSION

Setting configurations

The simulation software used was the Matlab. Also, in this software, it was created a tool for DWT bits processing. It was also used the traditional SPIHT encoder [1] customized for bit to bit debugging. Both were inserted in the developed generic procedure simulation as shown in Fig. 10. The set of 512 × 512 pixels images used for testing were: Airplane, Baboon, Lenna, Barbara, Goldhill, Peppers, Sailboat and Satellite. A DWT was used biorthogonal (bior4.4), 6 decomposition levels, using the integer part of coefficients.

The vector *vMSB* can be customized according to data in Table 1. In this proposal, the construction of this vector obeys the rule given below (eq. 2) to optimize the planes used.

$$vMSB = \{N-1, N+3, ..., N+3, ..., N+3, N+5\} \qquad (2)$$

where $2 < N < 12$ the restriction of DWT performance that generates coefficients with 15 bits for this test. Thus, the threshold value of the vector is: $vMSB = \{11.13, \ldots, 13,15\}$.

The results generated for comparison used 6 to 9 bits for the proposal and 11 to 14 bits in a system with the same modules but without the proposal. These settings were used because they generated the same performance range.

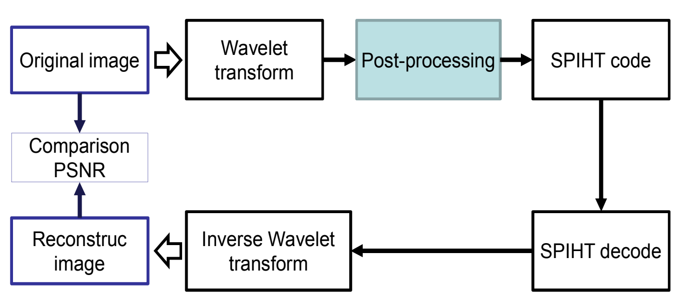

Fig. 10. Test run for DWT, the proposed post-processing and SPIHT.

Simulations results

The simulation results are shown in Fig. 11, where the traditional SPIHT uses DWT with $N = 11, 12, 13, 14, 15$ (bits) and the proposed SPIHT uses DWT plus post-processing with $N = 6, 7, 8, 9$ (bits).

Fig. 11 represents the performance curve SPIHT in PSNR with controlled rate from 0.2 to 1.2 bpp (bit per pixel). In this figure, it can be seen that N decreases and the performance reaches a level where it could not be improved anymore. However, this level in SPIHT post-processing (*pos-proc*) for N=9, rate (up to 1 bpp) is better than the traditional SPIHT N=12 and equal performance with $N = 13, 14$ or 15 bits. This level is defined in the proposal by the number of bits set to zero, so the algorithm only sees M_v. That also can disrupt the operation of the wavelet when the zeroed bit planes are not homogeneous (for $N = 6, 7$ a slight decline is generated after the1.0 bpp). The PSNR, on this curve, for $N = 8$ at a rate of up to 1.0 bpp has a negligible variation, up to 0.6 bpp is equal to the original performance with $M = 15$ using less 7 bit planes, or used only 53% of the original (for $M = 15$) with very close performance.

In Fig. 12.a an 11-bit configuration SPIHT without post-processing is marked. It should be noted that this curve is the closest to $N = 8$ used in the proposal. The improvement is about 2dB. In Fig. 12.b it is noted that for a rate between 0.2 to 0.4 bpp, the proposed $N = 7$ behaves the same as for $N = 12$ (or higher). In Fig 12.c is noted that for a rate of 0.6 to 0.8 bpp, the proposal with $N = 8$ the behavior is similar and very close to $N= 12$ (or higher). In Fig. 12.d observed that a rate of 0.2 to 1.0 bpp, the proposal with $N = 9$ behavior is equal to $N = 14$. In each of these comparisons it can be seen that the value of *'n'* decreases with the use of post-processing.

In summary, the requirement of using SPIHT and DWT with a post-processing $N = 9$ provides a performance similar to that which does not use it with $N = 15$ (greater than 0.8 bpp). For rates lower than 0.8 bpp, the performance is equivalent to the original $M =15$. Thus, it can be affirmed that

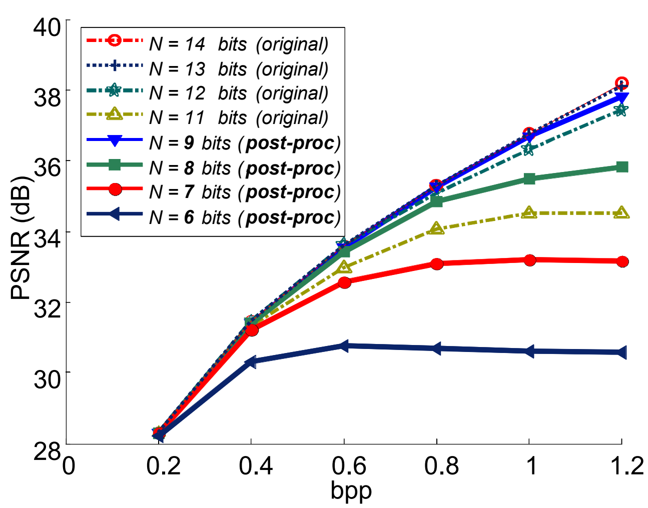

Fig. 11. Performance of the proposed coding using traditional SPIHT and DWT with post-processing in terms of signal to noise ratio (PSNR).

the post-processing method proposed here is important to reduce the amount of memory to be used in the encoder module DWT.

V. CONCLUSION

The post-processing method for DWT proposed here reduces the number of bits required to represent each wavelet coefficient. Calculating errors were introduced in the least significant bits, and the lost is of little significance for $N = 9$ at 0.8 to 1.0 bpp. The best configuration that were viable for a lossy compression scheme was found at $N = 8$ bits.

In the proposal for each compression rate is related a PSNR quality, as shown in Fig 12. Then, for each application an ideal rate could be set to minimize the quantity of bits used.

Simulation results show that the performance of traditional coding using SPIHT and DWT with the proposed post processing has PSNR equivalent to high rates for SPIHT compression.

ACKNOWLEDGMENT

The authors thank the group of Laboratory Visual Communications (LCV), CNPq, Fapesp, Capes, Capes-RHTVD and Faepex.

REFERENCES

[1] A. Said and W. Pearlman, "A new, fast, and efficient image codec based on set partitioning in hierarchical trees," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 6, no. 3, pp. 243 –250, jun 1996.

[2] D. S. Taubman and M. W. Marcellin, *JPEG 2000: Image Compression Fundamentals, Standards and Practice*. Norwell, MA, USA: Kluwer Academic Publishers, 2001.

[3] X. Xu and Y. Zhou, "Design of image data compression ip core based on processor local bus," in *Database Technology and Applications (DBTA), 2010 2nd International Workshop on*, nov. 2010, pp. 1 –4.

[4] J. Zhang, M. Jing, and W. Shaochuan, "Robust transmission of progressive images in the deep space communication," in *Wireless Communications, Networking and Information Security (WCNIS), 2010 IEEE International Conference on*, june 2010, pp. 4 –8.

[5] K. Kannan and S. Arumuga Perumal, "Optimal decomposition level of discrete wavelet transform for pixel based fusion of multi - focused images," in *Conference on Computational Intelligence and Multimedia Applications, 2007. International Conference on*, vol. 3, dec. 2007, pp. 314 –318.

[6] S. Rein and M. Reisslein, "Low-memory wavelet transforms for wireless sensor networks: A tutorial," *Communications Surveys Tutorials, IEEE*, vol. 13, no. 2, pp. 291 –307, quarter 2011.

[7] Y. Jin, Y. Wang, Q. Ruan, and X. Wang, "A new scheme for 3d face recognition based on 2d gabor wavelet transform plus lbp," in *Computer Science Education (ICCSE), 2011 6th International Conference on*, aug. 2011, pp. 860 –865.

[8] Y. Iano, F. da Silva, and A. Cruz, "A fast and efficient hybrid fractal-wavelet image coder," *Image Processing, IEEE Transactions on*, vol. 15, no. 1, pp. 98 –105, jan. 2006.

[9] J. Kulkarni, "Wavelet transform applications," in *Electronics Computer Technology (ICECT), 2011 3rd International Conference on*, vol. 1, april 2011, pp. 11 –17.

[10] W. A. Pearlman and A. Said, "Image wavelet coding systems: Part ii of set partition coding and image wavelet coding systems," *Found. Trends Signal Process.*, vol. 2, pp. 181–246, March 2008. [Online]. Available: http://dl.acm.org/citation.cfm?id=1482330.1482331

[11] P. Lamsrichan, "A fast algorithm for low-memory embedded wavelet-based image coding without list," in *Electrical Engineering/Electronics, Computer, Telecommunications and*

Information Technology (ECTI-CON), 2011 8th International Conference on, may 2011, pp. 979 –982.

[12] Y. Sun, H. Zhang, and G. Hu, "Real-time implementation of a new low-memory spiht image coding algorithm using dsp chip," *Image Processing, IEEE Transactions on*, vol. 11, no. 9, pp. 1112 – 1116, sep 2002.

[13] J. Oliver and M. Perez Malumbres, "On the design of fast wavelet transform algorithms with low memory requirements," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 18, no. 2, pp. 237 –248, feb. 2008.

[14] C. Chrysafis and A. Ortega, "Line based reduced memory, wavelet image compression," in *Data Compression Conference, 1998. DCC '98. Proceedings*, mar-1 apr 1998, pp. 398 –407.

[15] L. W. Chew, L.-M. Ang, and K. P. Seng, "Reduced memory spiht coding using wavelet transform with post-processing," in *Intelligent Human-Machine Systems and Cybernetics, 2009. IHMSC '09. International Conference on*, vol. 1, aug. 2009, pp. 371 –374.

[16] W. A. Pearlman and A. Said, "Set partition coding: Part i of set partition coding and image wavelet coding systems," *Found. Trends Signal Process.*, vol. 2, pp. 95–180, February 2008. [Online]. Available: http://dl.acm.org/citation.cfm?id=1482328.1482329

[17] A. Klappenecker, F. U. May, and A. Nueckel, "Lossless image compression using wavelets over finite rings and related architectures," A. Aldroubi, A. F. Laine, and M. A. Unser, Eds., vol. 3169, no. 1. SPIE, 1997, pp. 139–147. [Online]. Available: http://link.aip.org/link/?PSI/-3169/139/1

[18] K. Sayood, *Introduction to data compression*, ser. Morgan Kaufmann series in multimedia information and systems. Morgan Kaufmann Publishers, 2000. [Online]. Available: http://books.google.com.br/-books?id=ChSOjgiY84YC

[19] R. K. Bhattar, K. Ramakrishnan, and K. Dasgupta, "Strip based coding for large images using wavelets," *Signal Processing: Image Communication*, vol. 17, no. 6, pp. 441 – 456, 2002. [Online]. Available: http://www.sciencedirect.com/science/article/pii/-S092359650200019X

[20] L. W. Chew, L.-M. Ang, and K. P. Seng, "New virtual spiht tree structures for very low memory strip-based image compression," *Signal Processing Letters, IEEE*, vol. 15, pp. 389 –392, 2008.

Roger F. Larico Chávez, bachelor's at Systems engineering for San Agustin University, Peru in 2002. Masters and doctor degree in the School of Electrical and Computer Engineering, University of Campinas, SP, Brazil in 2006 and 2012 respectively. Currently he is researcher in Visual Communication Laboratory on the same University. His research interests include image and video processing, compression and transmission.

Yuzo Iano, received his PhD. in electrical engineering in 1986. Currently he is an associate Professor in electrical engineering at Unicamp (State University of Campinas, Brazil). He also works at Visual Communication Laboratory (LCV) on the same University. He is responsible for some digital signal processing (sound and image) projects. His research interests include video and audio coding, digital video and audio compression and digital signal transmission.

Osamu Saotome, bachelor's at Engenharia Eletrônica from Instituto Tecnológico de Aeronáutica (1974), master's and doctorate at Processamento Digital de Sinais from Tokyo Institute Of Technology in 1984 1987 respectively. Has experience in Electric Engineering, focusing on Electronic Circuits, acting on the following subjects: Real time systems, electronic devices and systems, algorithms, digital signal processing and radar receiver.

R. Arthur, He received the B.S. degree in electrical engineering from the University of São Paulo State (Unesp), Ilha Solteira, Brazil, in 1999, and the M.Sc. and the Ph.D. degrees in electrical engineering in 2002 and 2007, respectively, from the University of Campinas, SP, Brazil. Currently he is professor at the Telecommunication Division at the School of Technology, Unicamp, Brazil. His current research interest includes digital television in single frequency network environments, video compressing based on wavelets transform, video codec projects using digital signal processors and FPGA, turbo and LDPC coding and mobile health devices.

Rogério Seiji Higa, is currently a PhD. candidate at UNICAMP (State University of Campinas, Brazil). He received his Masters Degree in Electrical Engineering at UNICAMP in 2008. He also works at Visual Communication Laboratory (LCV) on the same University. His research interests include image processing and computer graphics.

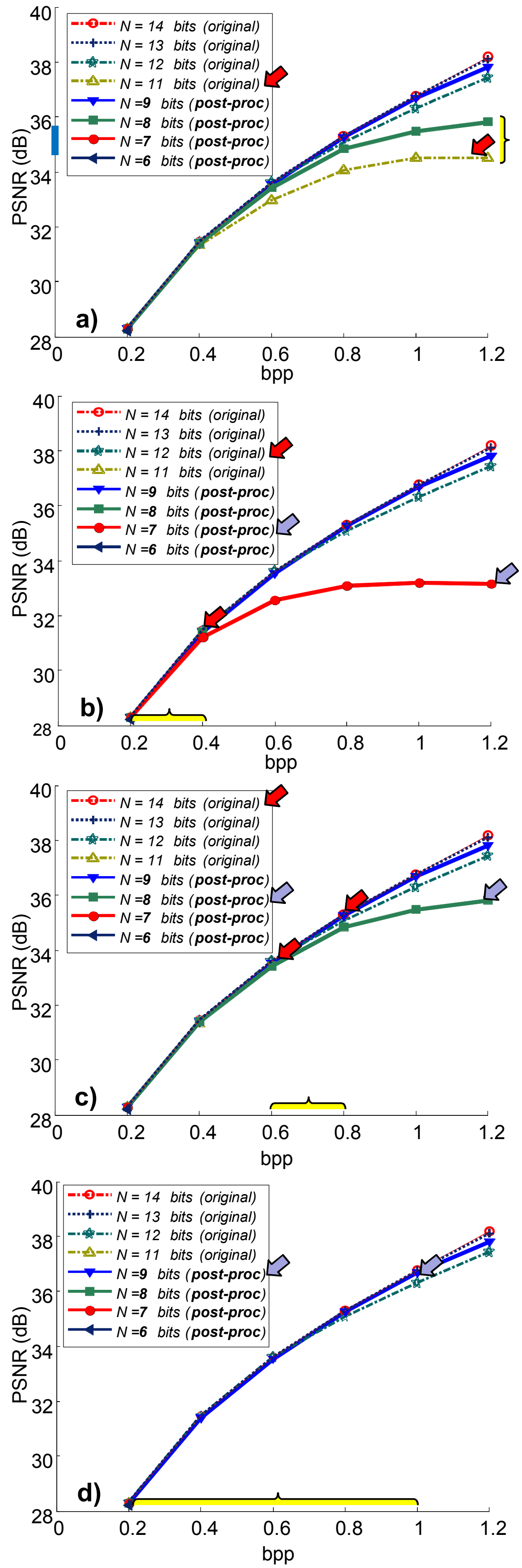

Fig. 12. Analysis of some points on the performance outcome.