Efficient Image transmission in Wireless Sensor Networks using Wavelet coded Preprocessing technique

Arputha Manesha.R
PG Student, Department of CSE,
PSN College of Engineering and Technology,
Tirunelveli, INDIA.

Velladurai . M
Assistant professor, Department of CSE,
PSN College of Engineering and Technology,
Tirunelveli, INDIA.

Abstract - In the recent years, the wireless technology would have known an exponential growth, which has an impact on developing and improving the field of telecommunications beyond the means of transmission wire to the radio frequency communication. The Wireless Sensor Network (WSN) is enrolled in this context. It's a collection of component (nodes) organized into a cooperative network. Image transfer in WSNs presents major challenge which raises issues related to its representation, its storage and its transmission. Image transmission challenges including limited bandwidth of cellular networks, restricted computational power, limited storage capability, and battery constraints of the appliances. In this paper we proposed a new scheme that has two stages in first stage we perform preprocessing using Bayesian technique next wavelet based compression .An efficient Wavelet based compression scheme that can significantly minimize the energy required for wireless image communication while meeting bandwidth constraints of wireless and network image quality. Based on Discrete Wavelet Transform, we propose an efficient image compression scheme, enabling significant reduction in computation energy needed with minimal degradation of image quality. The Proposed system better compression ratio and reduced transmission delay.

Keywords: Wireless Sensor Network; Image compression; Energy optimization, High-Pass Sub-band technique, Bayesian .DWT

1. INTRODUCTION

A wireless sensor network (WSN) is a wireless network consisting of spatially distributed autonomous devices using sensors to cooperatively monitor physical or environmental conditions. The development of such networks was originally motivated by military applications such as battlefield surveillance. However, wireless sensor networks are now used in many civilian application areas, including environment and habitat monitoring, healthcare applications, home automation, and traffic control [1-2]. As depicted in Fig. 1, data collected by sensors is transmitted to a special node equipped with higher energy and processing capabilities called "Base Station" (BS) or "sink". The BS collects filters and aggregates data sent by sensors in order to extract useful information. One of the major challenges in enabling image transfer services will be the need to process and wirelessly transmit very large volumes of data. This will impose severe demands on the battery resources of image-based applications as well as the bandwidth of the wireless network. Typically, images are compressed in order to save consumed energy.

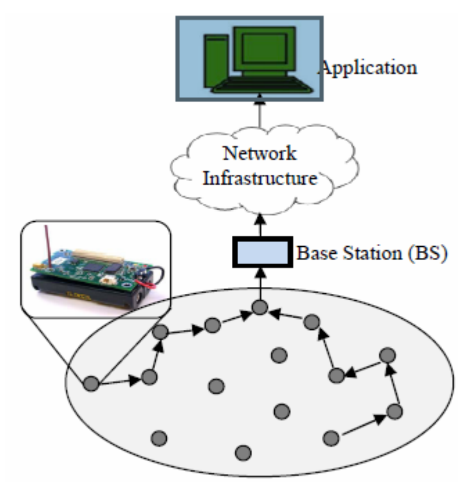

Fig-1 Wireless sensor networks

In this context, image transmission improvement over WSN is mainly done by the implementation of distributed image compression algorithm embedded in order to reduce the number of bits needed to represent an image by removing the spatial and spectral redundancies, thus reducing the energy consumption. The distributed image compression enables the sharing of computation load among sensors. This technique is based on the fact that an individual node does not have sufficient computational power to completely compress a large volume of image data to meet the application requirements; this is not possible unless the node distributes the computational task among other nodes.

In this tutorial we introduce communications and networking generalists without a background in signal processing to a range of wavelet transform techniques culminating in recently developed signal processing techniques that require only very small memory for wavelet transforms. In particular, the recently developed fractional wavelet filter [15] requires less than 1.5 Kbyte of RAM to transform an image with 256 × 256 8-bit pixels using only 16-bit integer arithmetic, as illustrated in Figure 1. Thus, the fractional wavelet filter works well within the limitations of typical low-cost sensor nodes [1], [2].

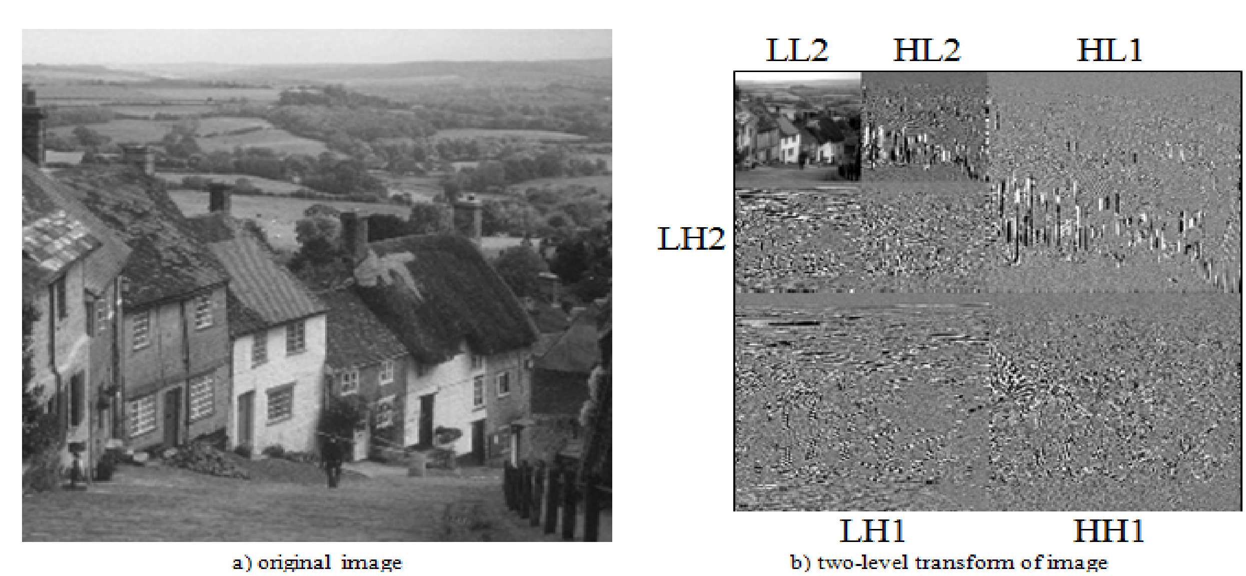

Fig-2 Two-level wavelet transform computed on a low-cost 16 bit microcontroller using less than 1.5 kByte of memory with the *fractional wavelet filter*. Each one-level transform results in four subbands denoted by LL, LH, HL, and HH . The contrast of each subband was adjusted to fill the entire intensity range.

II Wavelet –Transform

Over the last decade or so, wavelets have had a growing impact on signal processing theory and practice, both because of their unifying role and their successes in applications. Filter banks, which lie at the heart of wavelet-based algorithms, have become standard signal processing operators. Unlike Fourier transform, whose basis functions are sinusoids, wavelet transforms are based on small waves, called wavelets, of varying frequency and -limited duration. The Fourier Transform is probably the most popular transform used to obtain the frequency spectrum of a signal. But the Fourier Transform is only suitable for stationary signals, i.e., signals whose frequency content does not change with time. The Fourier Transform, while it tells how much of each frequency exists in the signal, it does not tell at which time these frequency components occur. Signals such as image and speech have different characteristics at different time or space, i.e., they are non-stationary. Most of the biological signals too, such as, Electrocardiogram, Electromyography, etc., are non-stationary. To analyze these signals, both frequency and time information are needed simultaneously, i.e., a time-frequency representation of the signal is needed. In, as frequency increases, the time resolution increases; likewise, as frequency decreases, the frequency resolution increases. Thus, a certain high frequency component can be located more accurately in time than a low frequency component and a low frequency component can be located more accurately in frequency compared to a high frequency component.

The 5/3 wavelet transform is well suited for mosaic images, the interleaving of color components are automatically taken care and it transforms as a smooth channel with out de-interleaving. The low and high pass filters of the 5/3 wavelet are

$$f_L = (-1/8,1/4,3/4,1/4,-1/8)$$

$$f_H = (-1/2,1,-1/2)$$

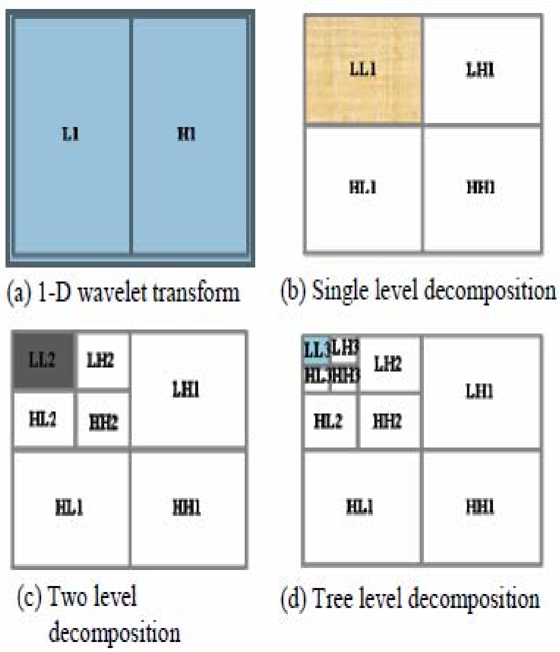

Fig.3. Illustration of wavelet spectral decomposition.

After applying 2-D low pass filter can be evaluated as $F_{LL} = f_L^T f_L$. The LL sub-band can be interpreted as a the luminance of the color image. The 2-D 5/3 HH high pass filter can be evaluated as $F_{HH} = f_H^T f_H$. Similarly the other sub-band filters can be evaluated. For reversible- integer – integer wavelet transform is performed in order to get lossless transformation . The following computations are performed.

$$d[n] = d0[n] – floor(½(s0[n+1] + s0[n]))$$

$$s[n]=s0[n]+floor(¼(d[n]+d[n-1])+½)$$

The input signal, low pass subband signal, and high pass subband signal are denoted as x[n], s[n] and d[n] and respectively. For convenience, we also define the quantities s0[n]= x[2n], d0[n] = x[2n+1]. The equations (3-1) can be written in the convenient form as follows

$$H[n] = x[2n+1] - floor(½(x[2n+2] + x[2n]))$$

$$L[n] = x[2n] + floor(¼(H[n] + H[n-1]) + ½)$$

According to [Mic00] , the 5/3 wavelet transforms requires the least computation among the various s reversible integer to integer wavelet transformations. For processing a coefficient it takes 5 Add operations, 2 shift operations and 0 multiplications , with a total of seven operations are required. The amount of memory needed by a transform based coder can influenced by the amount of memory required to store transform coefficients. All the stored values are integers. Analysis reveals that all four sub-bands LL,HL, LH and HH contain low frequency components of either chrominance or luminance signals. The sample computation of 5/3 wavelet transformation is described below.

One major difficulty in applying the discrete two-dimensional wavelet transform to a platform with scarce resources is the need for large random access memory. Implementations on a personal computer (PC) generally keep the entire source and/or destination image in memory; also, horizontal and vertical filters are applied separately. As this is generally not possible on resource-limited platforms, re- cent research efforts have examined memory-efficient wavelet transform techniques. Significant research efforts have gone into the implementation of the wavelet transform on field programmable gate arrays (FPGA), see for instance [16]–[19]. The FPGA-platforms are generally designed for one special purpose and are typically inappropriate for a sensor node that has to perform a variety of tasks including communication and analysis of the surrounding area [1], [2], [20]. This tutorial does not cover FPGAs; instead we focus on image wavelet transform techniques for a general microcontroller with very small RAM. The traditional approach to build a camera sensor node has been to connect a second platform with a more capable processor to the sensor node [12]. Instead, this tutorial considers a sensor node where a small camera is directly connected to the microcontroller through the universal asynchronous receiver/transmitter (UART) interface and the wavelet transform is performed on the microcontroller, which is extended by a directly connected multimedia flash memory card [5].

The multi-hop transmission path from a sensor node to a sink node in wireless sensor networks is exploited in a few studies, e.g., [21], [22], for distributing the wavelet coefficient computations over several nodes. We focus on the computation of the wavelet transform at one given node in this tutorial.

III PREPROCESSING

A Images With Quasi-Sparse Histograms

One of the major drawback of global off-line or online histogram packing is that if even of the intensity values appear only once or just a few number of times in the image, they will be considered by the histogram packing procedure as having equal importance as those that occur most frequently. In other words, images having "quasi-spares' histograms cannot benefit from this method.

After off-line histogram packing this image occupies 8 401 bytes, instead of 8 822 bytes when encoded normally with JPEG-LS (a gain of 4.8%). The analysis of the packed histogram of the "yahoo" image reveals that it still maintains a sparse appearance, even after been packed. However, in a strict sense, it is not sparse, because all bins concerning intensities lower that 156 are non-zero, although some of them account only for a few occurrences.

Based on this observation it seems reasonable to ask the following questions. Is there a way to avoid the intensities that have a very low number of occurrences and, therefore, to transform the quasi-sparse histograms in histograms that are strictly sparse?? If this is feasible, is it advantageous in terms of overall compression gain?

To investigate how this characteristic can be used to improve compression we propose the following approaches. Let P(Ii) denote the number of occurrences of intensity Ii in a given image, where [P(Ii) = NaNc is the total number of pixels of that image, Nr and Nc being the number of rows and columns, respectively. Let us also consider the set I={I0,I1,......,Is-1)) of the S<Nr intensities (Nr is the number of different intensities in the image), with P(Ii)>P(IJ), Therefore, I is the set of the S most used intensities. Moreover, we assume with loss of generality that Ii<Ij, Ai,i. Then, the following one-to-one order-preserving mapping in N0 is constructed.

h=(I0-0, I1-1, ..., Is-1-SS-1))

The algorithm proceeds as follows. If the image sample being processed, x belongs to the set I, then the packing procedure generates a output value y=h(x). Otherwise, i.e., if xe I, then the packing procedure generates an output value y= S and stores x (for example, in an auxiliary file). Therefore, when a given intensity that does not belong to the mapping is found, an escape symbol is generated. In our implementation, we use S as the escape symbol, i.e. the first integer not in the co-domain of h.

The success of this method depends, fundamentally, on how the increase in bit-rate generated due to storing the value of x €X is compensates by a more "compression-friendly" histogram packed image. The curves represent compression gain in relation to normal JPEG-LS encoding. Concerning the examples given with image "yahoo", the best result was obtained using a mapping with the 28 most frequent intensities (i.e., S=28), corresponding to an overall compressed size if 6 849 bytes. i.e. and improvement of 22.4% over normal (unpacked) compression, far better than the 4.8% obtained with off-line histogram packing.

B. Images With Locally Sparse Histograms

Generally, image data are not stationary. Therefore, a (global) histogram may not express correctly how intensities are used in different parts of the image. To illustrate this problem we refer to Kodak images. (768 rows x 512 columns, with different intensity values).

A simple analysis of this histogram would probably lead us to the conclusion that, due to its quasi-sparse appearance, the method described in the previous section would be the most appropriate, in order to obtain improvements concerning the compression of this image. In fact, it is not. Unfortunately, a simple inspection of the degree of sparseness that a given histogram exhibits is not enough to infer the impact of histogram packing in the loss less compression of an image. As can be observed, using analysis windows of 1 024 0r 4 096 pixels, the mean number of different intensities that is used simultaneously is significantly smaller (les than 10) than the number given by a global histogram analysis (249). For larger window sizes this number grows rapidly .To explore this characteristic, we implemented a packing procedure which, basically, performs off-line histogram packing on consecutive image segments of predefined size.

In effect, the normal JPEG-LS encoding of the Kodak images requires 58 792 bytes, while off-line histogram packing is only able to improve this number by 0.3%. However, using the local histogram packing approach, this number increases to 18.6% for Ws = 65 536, to 27.4% or Ws = 16 384, to 31.2% for Ws = 4096 and to 42.1% using Ws = 1 024. Moreover for Ws = 672, i.e., the number of pixels per image row, the improvement goes to a dramatic 60.6% (23 164 bytes), instead of the 58 792 bytes required by normal JPEG-LS)

IV PROPOSED TOPOLOGY

A. BAYESIAN ESTIMATION

The near-optimal threshold which is more suitable for image preprocessing. This approach can be formally described as Bayesian estimation. The formulation is grounded on the empirical observation that the wavelet coefficients in a sub band of a natural image can be summarized adequately by a generalized Gaussian distribution (GGD). This observation is well-accepted in the image processing community.

The goal is to find the soft-threshold that minimizes this Bayesian risk. The Bayesian risk is described as:

$$R(T) = E(Y-X)^2 = E_x E_{y/x}((X-X)^2$$

where $X=\eta_y(Y)$, $Y/X\sim N(x,\sigma^2)$, $X\sim f_{m,d}(x)$

B WAVELET CODING PREPROCESSING

The proposed wavelet coding preprocessing based on Bayesian estimation is performed in the following steps. After the wavelet transform, the following scheme is applied:

Step1. Preprocessing: Select proper threshold $T(\sigma_x)$ based on Bayesian estimation, and modified all sub image coefficients as:

$$\hat{C}_{ij} = \begin{cases} C_{ij}\}\sigma^2/\sigma_x, & \text{if } |c_{ij}| \geq T(\sigma_x) \\ 0 & \text{else} \end{cases}$$

where C_{ij} is the sub image coefficients. $T(\sigma_x)$ is the preprocessing threshold.

Step2. Initialization: In all preprocessed sub bands, find out the maximum absolute value MAX($\hat{c}_{ij}$) of all preprocessed coefficients.

$$\text{Set } T_0^{map} = \text{MAX}(\hat{c}_{ij})/2 + \Delta,$$

where T_0^{map} is the initial respond for quantization, and Δ is a small constant.

Step3. Significant map: If$(\hat{c}_{ij}) > T_k^{map}$, where T_k^{map} is the quantization threshold of the layer K. We add (i,j) to the significant map and encode it with '1S' where S is the sign bit, and modify it as:

$$(\hat{c}^{MAP}_{ij}) = \begin{cases} (\hat{c}_{ij}-1.5xT_K^{MAP} & (\hat{c}_{ij}\geq 0(S=1)) \\ (\hat{c}_{ij}+1.5xT_K^{MAP} & (\hat{c}_{ij}\leq 0(S=0)) \end{cases}$$

Step4. Refinement map: For $\hat{c}_{ij}$ who has been contained by the significant map in earlier layers, we encode the bit at layer k with a refinement bit 'D' and change the value $\hat{c}_{ij}$ to

$$(\hat{c}^{MAP}_{ij}) = \begin{cases} (\hat{c}_{ij}-0.25xT_K^{MAP} & (\hat{c}_{ij}\geq 0(S=1)) \\ (\hat{c}_{ij}+0.25xT_K^{MAP} & (\hat{c}_{ij}\leq 0(S=0)) \end{cases}$$

Step5. Iteration: Set $T_{k+1}^{map} = T_k^{map}/2$ and repeat Steps 3 ~ 4 for k = 0,1,2,..... . Note that each sub band may have different iteration number at the same time in Step 3.

To save computation energy, we propose an adaptive image transmission approach in WSNs consisting of technique to skip computation of certain high-pass coefficients of an image. This technique attempts to conserve energy by skipping the least significant sub-band. This technique is called "*SHPS*: Skipped High Pass Sub-bands". Fig.4 illustrates the distribution of high-pass and low-pass coefficients after applying 1-D wavelet transform to the 512*512 Lena image. We observe that the high-pass coefficients are generally represented by small integer values. Indeed, the most of the high-pass coefficients are less than 0.2. Since the image presents a low pass spectrum, high-pass filtering is skipped. Therefore, high-pass coefficients not computed resulting in a minimal image quality loss. Since the *new* technique is implemented by making specific modifications on the wavelet transform, all the images can still keep main information as 'Lena' when the high pass sub-bands are skipped.

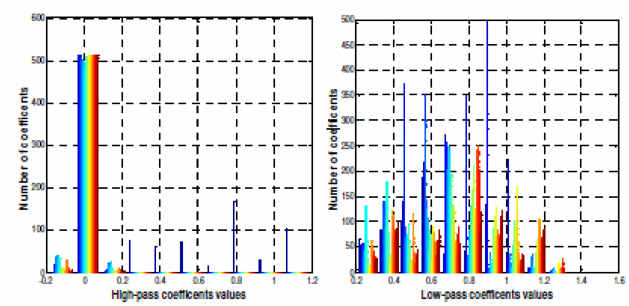

Fig.4.Numerical distribution of high-pass/low-pass coefficients after wavelet transform through 1-D DWT

Using the estimation technique presented, we have developed new technique which conserves energy by skipping the computation of high-pass coefficients. This technique attempts to conserve energy by skipping the least significant sub-band

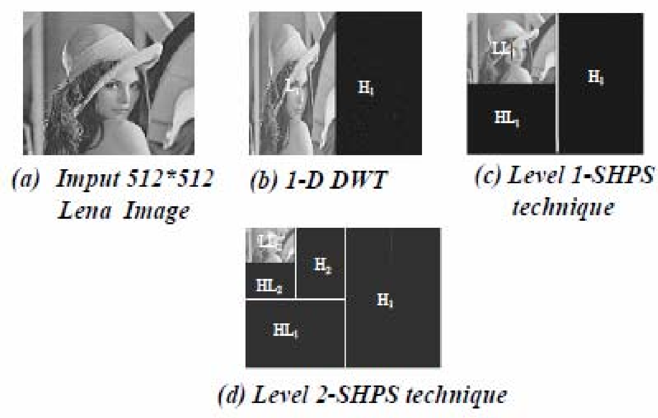

Fig. 5. The Proposed technique compression on 512*512 Lena image

V SIMULATION RESULTS

In this section, we used the Lena image sample, and measured the saving computation energy and the PSNR of the compressed image. The results are presented in Fig. 6. We observe that the *New* technique leads to significant energy savings at nominal loss in image quality. This technique can save up to 25% for level 1 decomposition and 33% for level 3 decomposition. However, the computation energy of *new* technique is associated with loss in image quality: PSNR=26dB for level 1 and PSNR=24dB for level 3. The above experiments demonstrate that depending on the image quality desired by a wireless service, and the state of the battery of the wireless appliances, by applying the *proposed* technique at different levels, different trade-offs can be obtained between the image quality obtained and the energy expended in compressing the image and transmitting the compressed image.

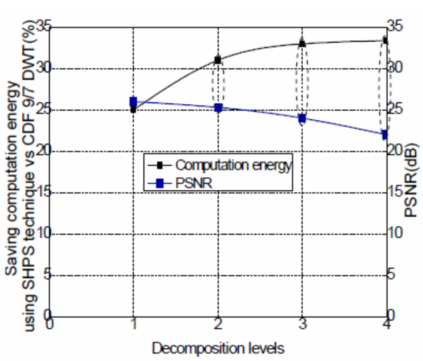

Fig. 6. Effects of NEW technique on image quality and computing energy

To get an idea of the impact on image quality, we present visual comparisons of two versions of the Lena image obtained. The image shown in Fig.7 (a) is obtained by using the *CDF 9/7 DWT*, while the image shown in Fig. 7(b) is obtained using the *new* technique through level 3. The PSNRs of the two images are 29 dB (*CDF 9/7 DWT*) and 24 dB (*new*) respectively.

(a) (b)

*Fig. 7. The image quality after, CDF 9/7 DWT and New techniques using the Lena 512*512 grayscale image.*

Conclusion

Image transfer in WSNs requires very large amounts of data to be transmitted, creating tremendously high energy and bandwidth requirements that cannot be fulfilled by limited growth in battery technologies, or restricted computational power and limited storage capability. This paper presents a potential solution to the emerging problem, by developing a novel technique for image compression called *SHPS*. Its mean objective is to optimize computation energy consumed with research must be focused on multipath routing which may enhance the performance of the distributed image compression.

REFERENCES

[1] S.A.Hussain , M.I. Razzak , A. A. Minhas' M. Sher, Energy Efficient Image Compression in wireless Sensor Networks International Journal of Recent Trends in Engineering, Vol 2, No. 1, November 2009

[2] A.E. Gamal, Collaborative Visual Sensor Networks, 2004, http:// mediax.stanford.edu/projects/cvsn.html.

[4] L. Ferrigno, S. Marano, V. Paciello, and A. Pietrosanto, "Balancing computational and transmission power consumption in wireless image

sensor networks”, IEEE29 International Conference on Virtual Environments, Human- Computer Interfaces, and Measures Systems (VECIMS), Giardini Naxos, Italy, July 2005.

[5] Min Wu and Chang Wen Chen, “Multiple bitstream image transmission over wireless sensor networks”, Proceedings of IEEE Sensors, volume 2, pages 727–731, October 2003.

[6] Huaming Wu and Alhussein A. Abouzeid, “Energy efficient distributed JPEG2000 image compression in multi hop wireless networks”, 4th Workshop on Applications and Services in Wireless Networks (ASWN 2004), pages 152–160, August 2004.

[7] H.M. Wu, A.A. Abused, Energy, “efficient distributed image compression in resource-constrained multichip wireless networks”, Computer Communication (Elsevier) 28 (14) (2005) 1658–1668.

[8] R. Wagner, R. Nowak, and R. Baraniuk. “Distributed image compression for sensor networks using correspondence analysis and super-resolution”, Proceedings of IEEE International Conference on Image Processing *(ICIP)*, volume 1, pages 597–600, September 2003.

[9] N. Boulgouris and M. Strintzis. “A family of wavelet-based stereo image coders”. IEEE Transactions on Circuits and Systems for Video Technology, 12(10):898–203, October 2002.

[10] A. Wang. A. Chandrakasan, “Energy efficient system partitioning for distributed wireless sensor networks”, Proceedings of the International Conference on Acoustics, Speech, and Signal processing (ICASSP-2001), Salt Lake City, Utah, 2001.

[11] S.S. Pradhan, J. Kusuma, K. Ramchandran, “Distributed compression in a dense microsensor network”, IEEE Signal Processing Magazine 19 (2) (2002) 51–60.

[12] D. Devaguptapu, B. Krishnamachari, “Applications of localized image processing techniques in wireless sensor networks”, SPIE’s 17th Annual International Symposium on Aerospace/Defense Sensing, Simulation, and Controls (Aerosense’03), Orlando, Florida, 2003.

[13] D. Ganesan, B. Greenstein, D. Perelyubskiy, D. Estrin, J.Heidemann, “An evaluation of multi-resolution storage for sensor networks”, Proceedings of the First International Conference on Embedded Networked Sensor Systems, ACM Press, 2003, pp. 89–102.

[14] S.D. Servetto, “Sensing lena–massively distributed compression of sensor images”, Proceedings of the IEEE International Conference on Image Processing (ICIP) (special session on ‘Distributed Source Coding’), Barcelona, Spain, 2003.

[15] S. Bandyopadhyay, E.J. Coyle,“An energy-efficient hierarchical clustering algorithm for wireless sensor networks”, INFOCOM, 2003.

[16] Vincent Lecuire, Cristian Duran-Faundez, and Nicolas Krommenacker, “Energy-Efficient Transmission of Wavelet-Based Images in Wireless Sensor Networks”, Eurasip journal on Image and Video Processing, 11 pages,2007.

[17] H. Wu, A.A. Abouzeid, “Energy efficient distributed JPEG2000 image compression in multihop wireless networks” ASWN 2004. pp 152-160.