



Low power hardware-based image compression solution for wireless camera sensor networks

Med Lassaad Kaddachi ^{a,*}, Adel Soudani ^a, Vincent Lecuire ^b, Kholdoun Torki ^c, Leila Makkouai ^b, Jean-Marie Moureaux ^b

^a Laboratoire d'électronique et micro-électronique (LAB-IT06) Faculté des Sciences de Monastir 5019, Monastir (FSM), Tunisia

^b Centre de Recherche en Automatique de Nancy, Nancy-Université, CNRS, Campus Sciences, BP 70239, 54506 Vandoeuvre-lès-Nancy Cedex, France

^c Circuits Multi-Project, 46, Avenue Félix VIALLET, 38031 GRENOBLE Cedex, France

ARTICLE INFO

Article history:

Received 10 June 2010

Accepted 8 April 2011

Available online 29 April 2011

Keywords:

Image compression

Wireless sensor networks

System-on-chip

Low power hardware

VLSI

ABSTRACT

In this paper, we present and evaluate a hardware solution for user-driven and packet loss tolerant image compression, especially designed to enable low power image compression and communication over wireless camera sensor networks (WCSNs). The proposed System-on-Chip is intended to be designed as a hardware coprocessor embedded in the camera sensor node. The goal is to relieve the node microcontroller of the image compression tasks and to achieve high-speed and low power image processing. The interest of our solution is twofold. First, compression settings can be changed at runtime (upon reception of a request message sent by an end user or according to the internal state of the camera sensor node). Second, the image compression chain includes a (block of) pixel interleaving scheme which significantly improves the robustness against packet loss in image communication. We discuss in depth the internal hardware architecture of the encoder chip which is planned to reach high performance running in FPGAs and in ASIC circuits. Synthesis results and relevant performance comparisons with related works are presented.

© 2011 Elsevier B.V. All rights reserved.

1. Introduction

The technological advances in wireless communication and micro-electronics allow today to build small, inexpensive, and battery-powered sensing devices with on-board processing and communication capabilities. These devices, called wireless sensor nodes, can be densely deployed over large regions of space in order to collect data from their surrounding and to send them to a sink node. The data packets are transmitted in multi-hop mode through a self-organized and infrastructure-less wireless sensor network which is built by coordination between the sensor nodes [18].

There is a wide range of applications envisioned for such wireless sensor networks, including, e.g., environmental monitoring, habitat sensing, precision agriculture, disaster management, structural health monitoring, target imaging and tracking, and military operations. However, the energy supply to these nodes is scarce due to the limited capacities of the batteries. This makes the energy efficiency as one of the fundamental problems in wireless sensor networks and consequently, special challenges for energy-efficient data processing and communication must be addressed.

More recently, there is a growing interest for the applications which require image sensing in order to achieve object detection, localization, tracking, and counting [2,5]. In this case, some nodes in the wireless sensor network are equipped with a small CMOS camera [1,6,7]. Such a sensor network is referred to as Wireless Camera Sensor Network (WCSN). However, digital images are usually represented by a very large amount of bits, and hence image-based applications raise up the problems related to energy consumption, memory size in the sensor nodes and available bandwidth in the wireless links [3]. Considering that the radio transceiver is one of the most power greedy components of sensor nodes, image compression at the source node seems a natural solution to make significant energy savings at the camera node as well as at the nodes forwarding packets towards the sink, and hence to extend the network lifetime [1,2,5]. Besides, the compression of image data contributes to reduce the risk of network congestion, and consequently the probability of packet loss. However, most current camera sensor platforms are software-based, as is the well-known Cyclops sensor board [6] and CMUCam3 [7]. Some works [8–10] have shown that popular algorithms such as JPEG, JPEG2000 or SPIHT are generally not efficient in software implementations because they lead to greater energy consumption than the transmission of the uncompressed image. That is due to the resource limitation of the software-based platforms in terms of processor speed. In the case of the Cyclops camera for example, where the software runs in TinyOS operating system environment, the processing time of the 2D-Discrete Wavelet Transform (DWT) on an

* Corresponding author.

E-mail addresses: lassaad.kaddachi@isigk.rnu.tn (M.L. Kaddachi), adel.soudani@issatso.rnu.tn (A. Soudani), Vincent.Lecuire@cran.uhp-nancy.fr (V. Lecuire), Kholdoun.Torki@imag.fr (K. Torki), Leila.Makkouai@cran.uhp-nancy.fr (L. Makkouai), Jean-Marie.Moureaux@cran.uhp-nancy.fr (J.-M. Moureaux).

8-bpp image (128×128), for the case of the 5/3 filter bank, is around 8 s. In the same way, the processing time of the Loeffler 2-D 8-point Discrete Cosine Transform (DCT) is around 7 s. As a result, hardware-based solutions are greatly sought after.

This paper is then a contribution in this field. Its main scope is to present a hardware solution for image compression at source node. This solution, presented as a CMOS circuit, is intended to be embedded in the camera sensor node. It will be considered as a coprocessor for tasks related to image compression and data packetization, which unloads the main microcontroller so that it will spend less time in active mode [11,12]. The proposed image compression scheme is designed to be robust against packet loss so that image communication could be achieved using a lightweight protocol without any error-correction functionality (i.e., without any kind of ARQ or FEC-based mechanism). This yields a significant reduction in energy consumption for the radio transceiver unit. Performance of our hardware solution is evaluated considering FPGA and ASIC circuits.

This paper is organized as follows. In Section 2, we discuss the related work and present our contribution. Section 3 describes the key features of the proposed image compression scheme and gives detail about the application-level performance. Section 4 provides the hardware specification of the proposed CMOS circuit and the analysis of its architecture, which is designed for WCSNs. Section 5 presents the synthesis results of our proposal using FPGA and ASIC circuits. Section 6 provides performance comparison between our solution and others found in the related literature. Finally, the last section provides concluding remarks and directions for future work.

2. Related work and contribution

There is a big concern among the research community about the implementation of image compression schemes on resource-constrained wireless camera nodes. The relevant parts of these researches aim to optimize both physical resources required at the node and networking constraints.

In Ref. [13], Yu et al. proposed an energy efficient JPEG 2000 image transmission system for WCSNs. JPEG 2000 is a DWT-based image compression standard which provides some error resilient coding schemes. The proposed system aims to minimize the overall processing-and-transmission energy consumption, given the expected end-to-end distortion constraint. However, the wavelet transform and the EBCOT coding process both require high computational resources and huge available memory size.

Lecuire et al. in Ref. [14] proposed to combine a DWT-based image coding scheme with a semi-reliable transmission scheme to achieve energy conservation. The DWT allows for image decomposition into separable subbands for multiresolution representation and packet prioritization purposes. The semi-reliable transmission scheme enables nodes located between the source camera sensor node and the sink to drop some packets in accordance with the packet priority and the batteries' state-of-charge. Such an image transmission approach provides a graceful tradeoff between the reconstructed image quality and the lifetime of the nodes forwarding data packets but it is not very efficient for the source camera sensor due to the computationally intensive DWT.

In Ref. [4], Lu et al. presented a low-complexity image compression scheme for WCSNs. This scheme is based on the Lapped Biorthogonal Transform (LBT), zerotree coding, and Golomb and Multiple Quantization (MQ) coders. The LBT is more suitable than the DWT because it requires much less calculation and memory space. For the same reasons, the Golomb and MQ coding algorithm fits more than Huffman coding or arithmetic coding. A distributed implementation scheme of the image compression algorithm is also proposed to overcome the computation resources and energy limitation of individual nodes by sharing the processing tasks among multiple

wireless sensor nodes. Certainly, collaborative in-network processing fits well the needs of resource-constrained wireless camera sensor networks. However, enabling scalable and dynamic cooperation among sensor nodes inevitably requires a lot of communication between them whereas this kind of communication is usually not very robust against loss.

Wu and Abouzeid presented in Ref. [15] a hop-by-hop reliability algorithm based on the generation and multiple copies sending of the same data bit stream after encoding it using Reed–Solomon (RS) codes. The data transits through cluster heads and other relaying nodes that are randomly chosen within every cluster. RS encoding and decoding algorithms are done at each relaying node, which chooses the strength of the RS code according to the estimated channel error probability. This scheme increases error robustness. However, it does not optimize the end-to-end performance, but rather makes additional energy cost and delay due to the extra processing performed at the relaying nodes.

Makkaoui et al. in Ref. [16] presented a zonal JPEG-based scheme which reduces the number of DCT coefficients to be computed, quantized and encoded. As a result, this scheme reduces the computation time and the energy consumption at each stage of the whole compression chain. This approach is attractive but the performance evaluation has been performed for a software implementation only.

In Ref. [17], Panigrahi et al. have developed a hardware/software architecture in order to optimize the energy consumption in WCSNs. This is based on JPEG compression algorithm which can be dynamically adapted to wireless communication. However, this compression scheme needs to be enhanced against packet error transmission.

It has been proved through these related works and others which are not cited here that the software implementation of image compression schemes is not suitable for WCSNs since it requires long processing time and leads to high energy consumption. These restrictions do not play in favor of many WCSN-based applications.

From these considerations, we believe that a hardware-based image compression solution will resolve the problem of the required energy and time processing. In fact, the solution that we propose is a specific hardware embedded circuit which has to relieve the main microcontroller of the node from the image processing tasks, and consequently which will spend less time in active mode and thus will decrease its energy consumption. Although several works exist on hardware-based image compression [19–23], the interest of our solution is twofold. First, compression settings can be changed at runtime (upon reception of a request message sent by an end user or according to the internal state of the camera sensor node). Second, the image compression chain includes a (block of) pixel interleaving scheme which significantly improves the robustness against packet loss in image communication. Our proposal is described in the following section.

3. The proposed image compression scheme

The image compression scheme that we propose lies on the <transformation-quantization-codeword assignment> conventional structure. Our goal is to have a low-complexity and low-memory scheme for meeting the hardware implementation requirements. The key features of our image compression scheme are:

- The transformation stage is based on the 2-D 8-point DCT algorithm, i.e., the image is divided in blocks of 8×8 pixels and each block is encoded independently. 2-D 8-point DCT is very popular in image compression but this transform is computationally intensive, and hence is energy consuming. Several fast DCT algorithms can be found in the literature. In the 1-D DCT domain, the Loeffler–Litgenberg–Moschytz (LLM) algorithm [24], with 11 multiplications and 29 additions, is the most efficient (11 multiplications is the theoretical lower bound). In order to reduce the computational complexity of the DCT, some algorithms such as BindDCT [25], Cordic DCT [26] and Cordic–Loeffler DCT [27] approximate multiplications

with add and shift operations, but at the expense of increasing the image distortion. Among them, the Cordic Loeffler algorithm (CL-DCT), with 38 additions and 16 shifts, offers the best tradeoff between the computational complexity and the image quality. Consequently, we adopted the CL-DCT fast algorithm which operates in the 1-D DCT domain. The 2-D 8-point DCT can be obtained by applying first the algorithm over the rows then over the resulting columns, with a cost of 608 additions and 256 shifts. An efficient implementation of CL-DCT in ASIC chips is given in Ref. [28].

- The quantization stage uses the quantization table that is recommended in the Annex of the JPEG standard [29]. This stage is processed to reduce the entropy of the DCT coefficients to be encoded.
- The codeword assignment stage is based on zigzag ordering, and Golomb and MQ coders [4,30,31]. The MQ coder is an approximate implementation of arithmetic coding tailored for binary data. Using Golomb and MQ coders instead of Huffman coding or arithmetic coding is significantly profitable from the standpoints of computational complexity and memory requirement.
- The robustness against packet loss is obtained by combining the independent block coding paradigm with the interleaving of the encoded blocks. On the one hand, independent block coding provides very strong support for error resilience because this ensures that encoded blocks correctly received at the sink will always be decodable (assuming, of course, that each packet sent by the camera node contains an integer number of encoded blocks). On the other hand, the use of interleaving scheme spatially de-correlates neighboring blocks of pixels by putting them into packets that are far distant apart from each other in the transmission sequence. If the interleaving scheme is properly designed, a missing block of pixels is likely to be surrounded by correctly received pixels.

For performance validation, we compared the rate–distortion ratio reached by our image compression scheme with those obtained by the standard JPEG compression scheme. Note that JPEG differs from our scheme in the transformation and the codeword assignment stages. JPEG uses the LLM algorithm instead of CL-DCT, and run-length and Huffman coders instead of Golomb and MQ coders. The Peak Signal to Noise Ratio (PSNR) was used as image quality measure. We selected a set of test images including the famous 512×512 Lena, Barbara, Goldhill, Baboon and Peppers. All the test images are 8-bpp grayscale images. Fig. 1 shows, for the case of Lena, the rate–distortion curves for both compared algorithms, considering that the quantization factor

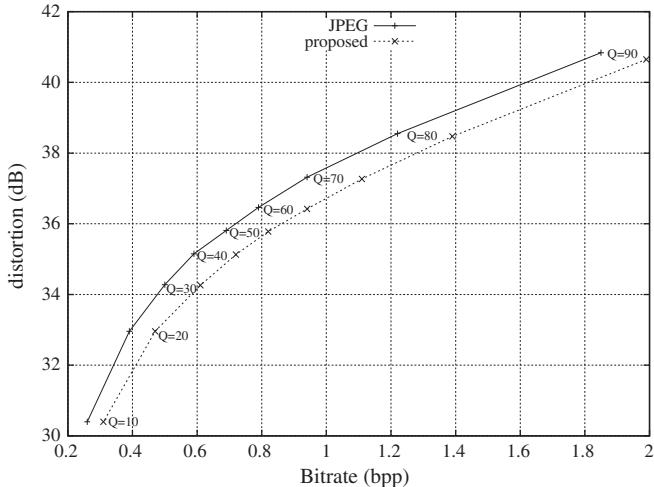


Fig. 1. Rate–distortion ratio comparison between JPEG and our proposal for the Lena test image.

was decreased by step of 10. The results show that JPEG offers slightly better rate–distortion ratio than our proposed scheme. We observed that JPEG yields an increase of PSNR between 0.5 and 1.0 dB of the reconstructed image for a given bitrate. Similarly, JPEG yields a decrease between 0.1 and 0.2 bpp of the bitrate for a given image PSNR.

Results are similar for other test images and are given in Table 1. This table shows that the difference between the PSNRs reached by both JPEG and our scheme is not significant.

Although JPEG outperforms our scheme from the standpoint of rate–distortion, our scheme is highly better than JPEG in software-based implementation from the standpoint of computational cost as shown in Table 2. This table provides the computational cost per 8×8 block of the transformation, quantization, and codeword assignment stages. We have considered the well-known Cyclops camera sensor [6], and consequently, the ATmega128L was used as the target platform. In Cyclops, the microcontroller operates at 7.3728 MHz with an active power consumption of 23 mW. The number of cycles was obtained using WinAVR20100110, a suite of executable, open source software development tools for the Atmel AVR series of RISC microprocessors. The results show that our scheme operates more than 7 times faster than JPEG.

Table 3 gives both execution time and energy consumption related to a 128×128 8-bpp image compressed at 0.5 bpp. The results prove clearly the efficiency of our proposal compared to JPEG in software-based implementation. Of course, same results are expected in hardware-based implementations. That will be studied in following sections.

Finally, our scheme is also better than JPEG from the standpoint of memory requirement. Indeed, both schemes require the same quantization and zigzag tables, but our scheme requires a 792-byte long table for MQ coder whereas JPEG requires two 704-byte long tables for Huffman coder.

4. Hardware specification

The goal of the hardware implementation is to achieve very short time and low power image processing. As said before, the proposed hardware solution is designed in the way that compression settings

Table 1
Rate–distortion ratio for different images.

Image (512×512 pixels 8 bpp)	PSNR (dB)	
	JPEG	Proposed
Lena	34.28	33.27
Barbara	27.89	26.90
Peppers	33.43	32.62
Goldhill	31.03	30.35
Baboon	23.68	23.15

Table 2
Computational cost of both JPEG and proposed schemes per 8×8 block compressed at 0.5 bpp on the ATmega128L microcontroller.

	Transform [cycles]	Quantization [cycles]	Coding [cycles]	Total [cycles]
JPEG	552,729	77,141	60,491	690,361
Proposed	4644	77,141	8857	90,642

Table 3
Execution time and energy consumption of both JPEG and proposed schemes for a 128×128 8-bpp image compressed at 0.5 bpp on the ATmega128L microcontroller.

	Execution time [s]	Energy consumption [mJ]
JPEG	23.97	551.33
Proposed	3.15	72.39

can be changed at runtime, for example upon reception of a request message sent by an end user or according to the internal state of the camera sensor node. Such a capability is attractive in the context of WCSNs for allowing a fine tuning of the tradeoff between user requirements and the potentially changing and unpredictable network conditions. The parameters which can be dynamically adjusted are:

- The *image size*, this parameter affects both of required energy and response time of the circuit. The control of this parameter can be ensured by the camera node to optimize its lifetime in regard to the available power in its battery. It can also be adjusted by the end user according to application requirements and network reliability (i.e., when the image quality decreases at the receiver side, the end user can ask for image size decrease in hope to reduce the traffic load in the network, and consequently the packet loss probability in the network).
- The *quantization factor*. This parameter changes the values of coefficients in the quantization table. It tunes the compression ratio of the image data, and hence the amount of data to be sent over the network. As a result, it affects the radio energy consumption and the response time of the wireless camera node. Of course, this parameter also affects the image quality obtained at the decoder side. Thanks to the adjustment of the quantization factor, the end user can ask for a quality increase of the received image according to an event occurrence, and then, at the end of the event, ask for a reduction of the image quality to reduce energy consumption in the camera node and extend network lifetime.

The proposed circuit supports image sizes up to 512×512 in 8-bpp grayscale format. It produces a packet elementary stream as output so that data packets are ready to be sent with no additional processing requirement at the main microcontroller.

Because our circuit is designed to work as a coprocessor for the node microcontroller (Fig. 2), it should communicate with this later through specific buses and signals to ensure the different steps required until full packet transmission.

Table 4
Interaction signals between processor and the node microcontroller.

Interaction signals between the processor chip and the node microcontroller	Function description	Signal width
config_QOS (microcontroller \rightarrow circuit)	QOS configuration for compression setting	1 bit
get_data (microcontroller \rightarrow circuit)	Signal requesting data	1 bit
image_coded (circuit \rightarrow microcontroller)	The end of coding process	1 bit
data_out (circuit \rightarrow microcontroller)	Available data at the output bus	1 bit
Bus_config_micro (microcontroller \rightarrow circuit)	Bus carrying the parameters of configuration	8 bits
data_dct_micro (circuit \rightarrow microcontroller)	Bus carrying the data packets	32 bits

Table 4 sums up the different internal signal's specifications between the circuit and the node microcontroller. These signals ensure compression setting request and handle packet transmission to the node microcontroller.

One important feature of the designed architecture is that each 8×8 data block is transferred between external memory and internal memory one time only. It will be processed and coded, then transmitted to the main microcontroller through the *data_dct_micro* bus. This was intended, from one side, to avoid heavy interactions between the process block and the external memory and from the other side to speed up the capability of the node to transmit more images in time.

The internal architecture of the proposed circuit had been designed as a set of synchronous blocks interacting through internal signals. It is a single chip core which contains the different processing modules. The different blocks of the compression circuit was described as sets of Extended Finite States Machine (EFSMs) in the RTL level.

The challenges in the development of this architecture are:

- Reducing the number of arithmetic operations in order to keep as possible low power consumption and short time responses. At this level, the use of the fast multiplierless CL-DCT algorithm helps to reduce the complexity of the compression scheme compared to the works presented in Refs. [19,20] that are based on the LLM DCT.

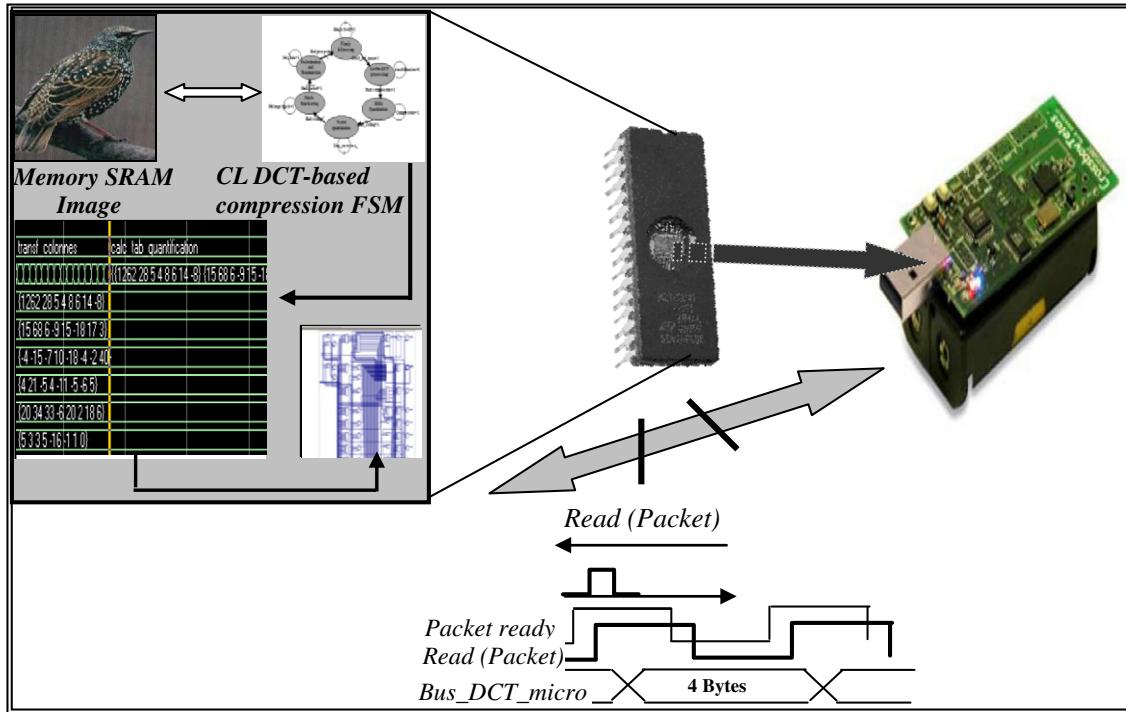


Fig. 2. Hardware image compression implementation at the source node.

- Minimizing internal memory space required for internal operations and transmitting the results at the end of coding process to the node microcontroller. As a result, this approach decreases the number of clock cycles required for interactions with the external memory. The more is heavy this interaction, the more the consumed energy and execution time increase.

Fig. 3 sums up the hardware architecture for image compression scheme based on CL-DCT transform. The proposed architecture for images compression and transmission was partitioned in four main blocks. These four blocks are 2-D DCT, quantization, Golomb and MQ coding, and packetizer.

In the following, we are going to detail the different blocks used in the proposed architecture.

- SRAM Memory:** It is an external memory generated by the STMicroelectronics company. It is used for saving the image data collected by the camera sensor, corresponding to the input samples.
- The CL-DCT:** It is used for image transformation in order to avoid the complexity problem of the classical DCT use and to keep its general advantage in getting independent-blocks that can be sent through the wireless channel, without specific correlation between them. As said before, the 2-D 8-point DCT can be obtained by applying first the algorithm over the rows then over the resulting columns. The designed architecture for the 2-D 8-point uses 1267 memory bits. The latency of this block is of 102 clock cycles. This reduction in number of clock cycles is due to the multiplierless properties used in the computation of the CL-DCT algorithm. Besides, 2-D DCT uses almost three times more logic cells than all other modules. This feature promises low-complexity and low power solution with hardware implementation of the total compressor chip [28].
- The JPEG quantization block:** After the CL-DCT transform, the quantization step is processed to reduce the entropy of the DCT coefficients [17,18]. The standard JPEG defines an efficient quantization table that allows a high compression rate, with minimal loss in quality, and thus we adopt it. The coefficients of CL-DCT block are divided by the quantization table [18,33]. The compression ratio is controlled by the image quality factor (Fig. 3) that can be an integer value from 1 to 100. As this factor decreases, the image quality

decreases, and the amount of data to be transmitted also decreases, which leads to less required energy to transmit the image. The designed architecture for the quantization uses 843 memory bits. The latency of this block is of 8 clock cycles. The first four cycles are used to compute the quantization table. The second four cycles are used to compute the division results.

- The Golomb coding:** It is one lossless compression scheme which encodes the quantized coefficients into a form meaningful for the MQ coder. An implementation of Golomb coding can be found in Ref. [30]. A Golomb code has two components. The first component is called unary coding. It is r/m of '1' symbol, followed by a single '0'. The second component (truncated binary encoding) is $r \bmod m$ coded as an ordinary binary number with $\log_2(m)$ bits. When m is the parameter which can be changed for each symbol being coded and will be sent to the decoder as side information. The encoding procedure has a very simple realization and has been referred to as Rice coding in the literature in the case where m is power of 2 [34,35]. The key factor behind the effective use of Golomb–Rice codes is the estimation of the parameter m to be used for a given sample or block of samples. In our design, we have fixed the parameter m as an integer value equal to 2.
- The MQ coding:** It is an approximate implementation of arithmetic (lossless) coding tailored for binary data [31,36]. The encoding process begins by estimation of the frequencies of occurrence of each binary symbol (of the Golomb codes). The method starts with a certain interval (to be defined); it uses the probability of each coefficient to divide the current interval into sub-intervals. Specifying a sub-interval requires more bits, so the number constructed by the algorithm grows continuously. To achieve compression, the algorithm is designed such that a high-probability symbol reduce the interval less than a low-probability symbol, with the result that high-probability symbols contribute to fewer bits of the output. The output of MQ coding is interpreted as a number in the range [0, 1]. The most important advantage of MQ coding is its flexibility: it can be used in conjunction with any other coding method that can provide a sequence of event probabilities. This leads to higher efficiency and a better compression ratio in general [31,36].

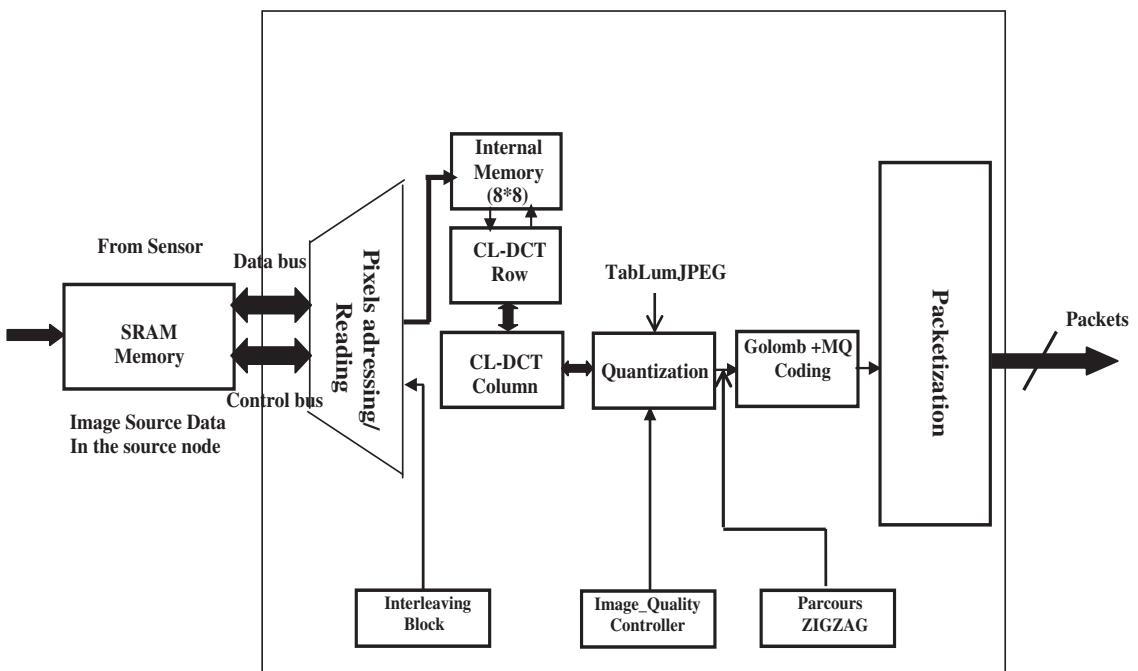


Fig. 3. Block diagram of image compression circuit.

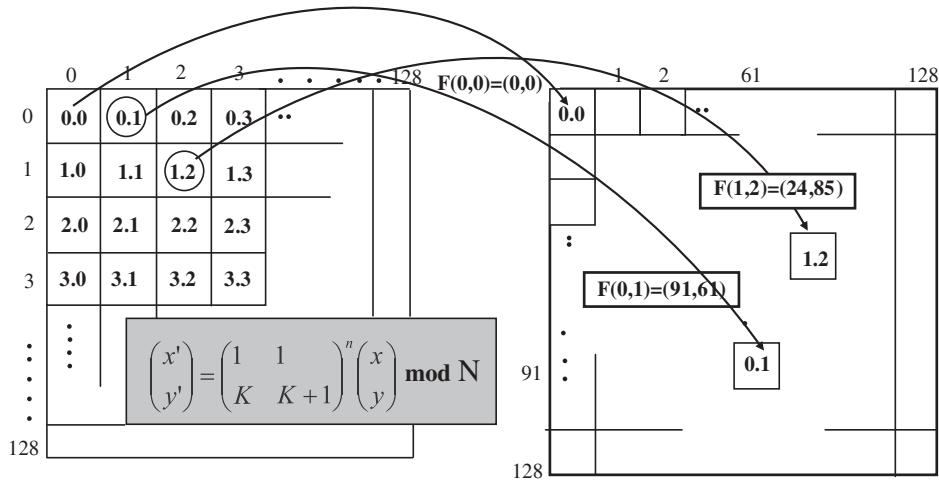


Fig. 4. Coefficients interleaving for error robust transmission.

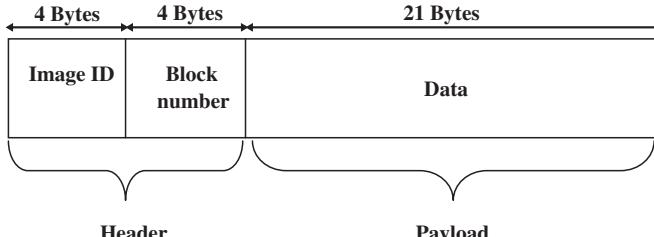


Fig. 5. Used packets structure.

• *Block Interleaving for Robust Image Communication* [8]: High packet loss rate is the ultimate characteristic of wireless sensor networking systems. This characteristic prevents these systems from ensuring reliable packet transmissions. To overcome this reliability weakness, multiple retransmissions are often required between sources and sink nodes. This strategy decreases network availability and misleads energy saving approach which in turn reduces network lifetime. For image transmission, the use of DCT transform makes possible to build independent data packets. This approach can be enhanced by a block interleaving scheme based in Torus Automorphisms (TA) [37] before transmission (Fig. 4).

In Ref. [8], Duran-Faundez et al. show a significant increase of the image quality under high packet loss conditions when this mechanism is applied in comparison to a sequential image transmission approach, while preserving similar energy consumptions, time and low-complexity.

• *Packetization*: It divides the compressed image into packets in order to send them over the network. The packet format is standardized by TinyOS platform [8]. It defines the most used length of 29-byte packets. We have specified our own packet format for transmitting images from a source node to a sink node (Fig. 5). The header is composed of 8 bytes.

The first 4-byte field contains the image ident. The second 4-byte field indicates the block sequence number in the packet. The payload data of the specified packet is composed of 21 bytes.

Performance evaluation and comparison of our hardware solution are studied in the following sections.

5. Prototyping circuits and performances analysis

As being discussed, the hardware implementation is being adopted in the hope to reduce required power for image compression and transmission in WCSNs [11,12,21].

The proposed encoder architecture for image compression and transmission was described with VHDL language as sets of Extended Finite States Machines (EFSMs) at the RTL level. This description has been used for implementations in both of FPGA and CMOS ASIC [38] circuits.

The target circuits used for FPGA prototyping are Xilinx Spartan-3 and Altera Flex10KE available as prototyping environments in our Laboratory [39]. We have used the Simplify 9.1.2 and ISE 10.1 tools for the synthesis. To design the ASIC circuits, we have used other dedicated tools; Design Vision for synthesis with different CMOS standard cell libraries and Encounter for the placement and routing steps when performing the layout of the circuit at CMP (Circuits Multi Projects) of the TIMA Laboratory.

The main goal of the hardware study is to point out the circuit characteristics and to check its adequacy for WSN use. As, being discussed in the previous sections, the online compression settings are controlled through the quantization factor and the image size parameters. We note that the quantization factor doesn't influence the hardware resources neither the processing time. It only changes the compression ratio.

To permit the image size change, we have designed the circuit with the capability to process a maximum image size of 512×512 pixels coded on 8 bpp.

5.1. Synthesis for FPGA circuits

The FPGA circuit solution is, nowadays, considered as one of the most used prototyping environments. In fact, with the increase of the available programmable cells in these circuits, it is being possible to implement a wide range of sophisticated algorithms. Furthermore, with their adaptability to the re-use concept [38], the FPGA solutions are considered more than a validation step of research approach. They are often used as a target platform in the final product for telecommunication and multimedia applications.

Table 5 presents a summary of the synthesis results for the different circuit blocks and for the whole encoder architecture, considering Altera circuit (FLEX 10KE FPGA). We have pointed out, in this table, the

Table 5
Synthesis results of the FPGA encoder.

	Logic cells	Period (ns)	Frequency (MHz)	Latency (cycles)
2-D CL-DCT	1822	17.15	58.3	102
Quantization	302	18.77	53.25	8
Golomb coding	88	6.78	147.33	5
MQ coding	322	14.8	67.5	48
Packetizer	98	8.79	113.75	32
Encoder circuit	2632	16.45	60.77	195

hardware resource occupancy (logic cells), the required latency for the image compression and the processing frequency of the four main encoder blocks and of the complete encoder circuit. We note that the synthesis results do not include the cost of the external SRAM memory.

The results, in Table 5, show that both 2-D CL-DCT and quantization block have the lowest processing frequencies within the four main blocks. Given that, the frequency of the global encoder circuit will be slowed by these modules and especially by the 2-D CL-DCT module.

The use of the 2-D CL-DCT, as a fast transform method compared to the classical DCT, helps to speed up the response time of the circuit. This block facilitates the whole designed circuit to exceed, in terms of frequency, the performance of some encoder architectures being presented in the literature [19–21].

The encoder circuit, when mapped to a FLEX 10KE FPGA, reaches a minimum period of 16.45 ns, allowing a processing rate of 60.77 Mpixels per second. With this clock cycle, the CL-DCT based encoder is able to process more than 231 images ((512×512) pixels, 8 bits/pixel) per second (i.e., 927 images ((256×256) pixels, 8 bits/pixel)).

We have studied, for the same architecture, the performances of image compression circuit based on CL-DCT and using the entropy coding algorithm. This solution is a JPEG-like image compression scheme that is used, in our study, to compare its performances to the proposed circuit.

Using the same target FPGA, the JPEG-like compressor circuit reaches a minimum period of 23.25 ns, allowing a processing rate of 43 Mpixels/s. The JPEG-like compressor is able to process more than 164 images ((512×512) pixels, 8 bits/pixel) per second. When compared to our proposed, JPEG-like circuit has less performances than our CL-DCT based-algorithm circuit.

Table 6 sums up the estimated processing time and the required power for the encoder circuit.

This table shows that the estimated processing time ranges between 2.86 ms for (128×128) pixels image size and 8.03 ms for (512×512) pixels image size. It shows also that the estimated energy consumption is 254.68 mJ for the first case and 538.01 mJ for the second one.

Using Spartan-3 FPGA circuit, the performances of the encoder circuit are within the same scale. In fact, with the minimum reached period of the encoder is 15.06 ns, the proposed encoder is able to process 66.4 Mpixels/s. In this case, more than 253 images ((512×512) pixels, 8 bits/pixel) per second are processed. The power consumption is around 96 mW.

5.2. Synthesis using CMOS standard-cell

The studied CL-DCT based-algorithm encoder had been prototyped as an ASIC circuit. It is a single chip core which contains the different processing modules particularly the SRAM generated by STMicroelectronics company with different integration technologies. This memory has been optimized for short time access and low power processing. It has a size of (128×128) pixels. Each pixel is stored on 8 bits. This memory size allows the circuit to deal with lower image sizes.

As being previously explained, we have optimized as much as possible, during the specification process, the internal memory space duplication to make the size smaller. The bit number per bus cycle

transferred between the memory and the compression tasks has been fixed to 8 bits.

We have studied the ASIC characteristics of the proposed compression circuit with different integration technologies. We have used, for this purpose, the 45 nm, 90 nm, 0.13 μm, 0.18 μm and 0.6 μm CMOS standard cell libraries of Synopsys tool. The main addressed features of the circuit in this study are the frequency, power and area. Table 7 reports a summary of these characteristics. These results concern the processing core of the circuit without connection plots. We note that the power consumption increases significantly with clock frequency.

As it is shown in this table, the maximum operating frequency of the CL-DCT based encoder is obtained when synthesized with 45 nm standard cell libraries. The proposed encoder circuit reaches a minimum period of 3.75 ns, allowing a processing rate of 266 Mpixels/s. It occupies 0.23 mm² and consumes 13.35 mW.

When mapped to 0.6 μm CMOS standard cell libraries, the CL-DCT based encoder will clock slowly (37 MHz). In this case, both power consumption and occupied area are clearly significant (38.5 mm² and 1.02 W respectively). The proposed encoder chip is able to process 37 Mpixels/s.

Table 7 resumes the comparison with the JPEG-like compression circuit. The results show that the frequencies and the total cell's areas, of our proposed circuit, obtained with the different integration technologies are better than those of the JPEG-like solution. The power consumption of our circuit is more attractive than the JPEG-like circuit for the integration technologies 90 nm, 0.13 μm and 0.18 μm. It is a little big higher than the JPEG-like circuit with the others technologies.

As expected, the synthesis results for ASIC performances when using new CMOS standard cell libraries are more significant in terms of frequency and dynamic power consumption compared to the FPGA prototyping results. In fact, the ASIC circuit is three times faster and consumes much less than the FPGA solution. However, the ASIC encoder circuit when implemented with 0.6 μm integration technology has less attractive characteristics in power consumption, frequency and required area when compared both to ASIC, with new integration technologies, and FPGA studied solutions.

The use of the compression chip based-ASIC implementation with 45 nm CMOS standard cell looks to be the most adequate for WSN constraints. In fact this circuit is the best in terms of operating frequency, area and power consumption.

For placement and routing step, the environment Cadence (Encounter Tool) was used to perform the circuit layout. It has a circuit area of 0.955 mm². The SRAM memory [32] used for this design presents 20.57% of the chip size. We note that it has a cells placement density of 73%. Fig. 6 shows the layout of the proposed circuit with 45 nm CMOS standard cell library.

Table 7
General characteristics of the circuit for different standard cell libraries.

		F max (MHz)	Total cell area (mm ²)	Power (mW)
CL-DCT-based encoder	45 nm	266	0.23	13.35
	90 nm	120	0.89	11.23
	0.13 μm	160	1.56	28.12
	0.18 μm	190.66	2.76	14
	0.6 μm	37	38.25	1.02 W
JPEG-like encoder	45 nm	207	0.27	10.5
	90 nm	110	1.18	15.66
	0.13 μm	142	1.96	36.56
	0.18 μm	158.33	2.91	42.98
	0.6 μm	33.25	43	970

Table 6
Encoder performances on FPGA circuit.

Image size	Estimated processing time (ms)	Total power (mW)	Estimated energy (mJ)
128×128	2.86	89.05	254.68
256×256	5.23	83.66	437.54
512×512	8.03	67.0	538.01

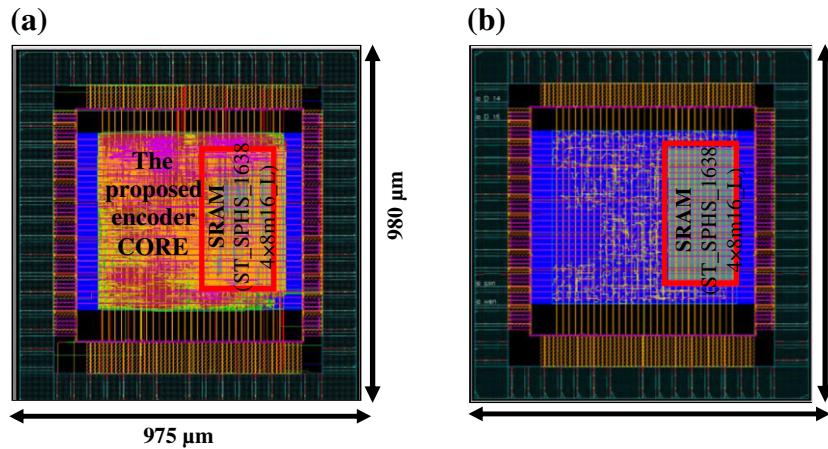


Fig. 6. Layout of the proposed hardware processor: (a) placement and routing of the compression chip and (b) clock tree routing.

6. Comparison with other proposed solutions

There is few number of completely designed encoders reported in the literature, some of them are ASIC designs and others are FPGA-based designs. In this section, we have dressed a comparison of the proposed solution regarding these implementations for image compression on chip.

6.1. Comparison with FPGA implementations

We have dressed a comparison of the proposed solution with other implementations of the image compression on chip [19,20,41]. These solutions have been proposed mainly for embedded system processing. In the following section, we provide brief descriptions of these solutions:

- In Ref. [40], an implementation of a JPEG compressor, using low-cost and low power FPGAs, has been presented. The JPEG compressor was partitioned into four main blocks. These four blocks are: 2-D DCT, quantization, zigzag buffer and entropy coder. The authors have reported an operating frequency equal to 37.6 MHz and a maximum rate of 122.4 fps for 640×480 gray scale images.
- An FPGA solution has been studied in Ref. [19] for the implementation of JPEG encoder with an optimal resources use as a scope. The compression circuit has five main units: DCT, quantizer, RLE_encoder, Huffman codec, and generate_output that are responsible of handling the constraints set by the JPEG standard. The maximum operating frequency of the JPEG circuit is 48.5 MHz.
- The JPEG compressor proposed in Ref. [20] was partitioned into four main blocks: 2-D DCT, quantization, zigzag buffer and entropy coder. These blocks were described in VHDL and synthesized to

Altera and Xilinx FPGAs circuits. The maximum reached frequency is 39.84 MHz. Synthesis results show that the 2-D DCT block will clock slowly compared to the four main blocks.

- Paper [22] proposes the hardware implementation of lossless JPEG-LS encoding. The proposed algorithm consists of two main units, a context modeler and a Golomb–Rice code. The JPEG-LS encoder was implemented on several FPGA devices. Particularly, when mapped to Virtex-E XCV1600E, the JPEG-LS encoding reached an operating frequency of 66.9 MHz.

Table 8 sums up the performances of our proposed solution and dresses a comparison with the previously described circuits.

This table reveals that the proposed implementation is the most efficient in terms of throughput, logic area and operating frequency. This was expected due to two main reasons:

- The use of the fast multiplierless CL-DCT algorithm compared to the most of previous implementations using LLM DCT.
- The entropy coder (RLE + Huffman) basically used in these implementations is not adopted in our compression scheme to reduce complexity, which in turn will decrease area and increase the frequency of the circuit.

6.2. Comparison with ASIC implementation

Some related works on image compression using ASIC circuits have been reported in literature [22,33,41]. These solutions have been implemented using 0.6 μ m and 0.18 μ m integration technologies. The details of these circuits are the following:

- The authors in Ref. [33] proposed a JPEG chip for image compression. The compression circuit is partitioned into four modules: DCT/IDCT, quantizer/dequantizer, zigzag, and Huffman codec. The circuit is implemented in CMOS standard cell using 0.6 μ m. It occupies 45.54 mm² and consumes 1000 mW.
- In Ref. [22], Marcos et al. proposes the hardware implementation of lossless JPEG encoder (JPEG-LS). The proposed algorithm consists of two main units, a context modeler and a Golomb–Rice code. The JPEG-LS encoder was implemented on ASIC CMOS standard cell libraries using 0.18 μ m technology. The maximum reached frequency is 183 MHz. The occupied area is 27,681 equivalents Gates.
- In Ref. [41], M. Lin et al. have studied the implementation of an image compressor for wireless capsule endoscopy application. The proposed compression algorithm has been modified to comply with the JPEG standard. Their architecture is based on Lempel–Ziv (LZ) coding and implemented using the CMOS 0.18 μ m technology. The maximum reached frequency is 12.58 MHz, and consumes 14.92 mW MHz. The occupied area is around 390 k gates.

Table 8
Comparison to other available FPGA implementations.

Design	FPGAs technology	Logic cells	Frequency MHz	Throughput (Msamples/s)
Solution in Ref. [41]	Altera Flex10KE	4568	37.6	37.6
Our solution	Altera Flex10KE	2632	60.77	60.77
Solution in Ref. [19]	Xilinx Spartan-3 XC3S200	2711	48.5	48.5
Our solution	Xilinx Spartan-3 XC3S200	2385	66.4	66.4
Solution in Ref. [20]	Altera FLEX 10KE	4844	39.84	39.84
Our solution	Altera FLEX 10KE	2632	60.77	60.77
Solution in Ref. [22]	Xilinx Virtex-E XCV1600E	4929	66.9	66.9
Our solution	Xilinx Virtex-E XCV1600E	3938	73.5	73.5

Table 9

Comparison with others ASIC designs for image compression.

Design	ASIC integration technology	Area (mm ²)	Frequency MHz	Power (mW)
Our solution	45 nm	0.23	266	13.35
Solution in Ref. [33]	0.6 μ m	45.54	27	1000
Our solution	0.6 μ m	38.25	37	1002
Solution in Ref. [22]	0.18 μ m	27,681 (Gates)	183	–
Our solution	0.18 μ m	2.76	190.66	14
Solution in Ref. [42]	0.18 μ m	390 (K gates)	12.58	14.92

Table 9 provides a comparison of our ASIC solution, designed using different integration technologies, with some related works described in the above section.

From the following table, it is easily observed that the proposed CL-DCT based encoder implementation presents significantly higher processing frequency with less area then the other presented solutions.

6.3. Comparison with TinyOS solutions

As this solution is intended to replace software implementation with TinyOS systems, we have compared its performances to others works developed with software approach. C. Duran-Faundez et al. studied in Ref. [42] a JPEG implementation in a Mica2 node connected to a Cyclops image sensor. The authors studied the power consumption and the processing time related to the image capture, the compression and the transmission processes. They have measured, for these tasks; the energy cost of a (128 × 128) image as about 860 mJ and the time processing is around 13.5 s.

For the same image size, the hardware solution that we propose in this paper can achieve lower energy consumption (254.68 mJ) and shorter processing time (2.86 ms) with the use of the FPGA-based implementation. The ASIC-based solution proposed in this paper can offer low energy consumption. Particularly, when we use the ASIC-based implementation with 45 nm, the energy consumption is highly reduced (38.18 mJ). The gains in energy are about 70% for the FPGA-based solution and in the range of 95.5% for the ASIC-based solution. Moreover, this study shows that the gain in term of processing time is about of 99% when we use the hardware-based solution.

In Ref. [16], Makkaoui et al. have studied a JPEG-like compression implementation based on a zonal approach in a Telos node connected to a Cyclops camera [43]. The authors studied the power consumption and the processing time related to the image capture, the compression and the transmission processes. They have measured, for these tasks, the energy cost of a (128 × 128) image as about 35.21 mJ and the processing time is 12.77 s. For the same image size, the hardware solution proposed in this paper can achieve shorter processing time (2.86 ms) (around 99.9% of saving in process time). The energy consumption is reduced to 32.11 mJ with a gain of approximately 8.8%.

Based on these comparisons, we think that the CL-DCT based encoder circuit proposed in this work for image compression in WCSNs offers the best tradeoff between energy consumption and processing time. It could be adopted to perform efficiently multimedia applications built on WSN.

7. Conclusion

In this paper, we have presented a hardware solution for low power image compression based on CL-DCT transform, quantization, and Golomb and MQ coders. This solution is intended to be embedded as a coprocessor for the sensor node's main CPU. The performance of this circuit has been studied for both FPGA and ASIC prototyping solution.

This solution provides dynamic compression settings at run time according to the camera node state and the user requirements. It saves energy and grants short time image compression compared to a software

implementation of the image compression scheme. It enhances the capability of the receiver to extract high quality image and to avoid packet retransmission with the introduction of image-block interleaving scheme. In addition, the proposed circuit will contribute to unload the node-microcontroller from image processing tasks giving it more computational bandwidth for network interactions.

As a future work, we think that this solution can be enhanced with the integration of a CMOS image sensor that will reduce the energy and the time acquisition of the image. With this approach, it seems possible to tune on the physical sensing process in order to compress the image and then to avoid considerable energy loss and lengthy time processing.

References

- [1] E. Culurciello, A.G. Andreou, CMOS image sensors for sensor networks, *Analogue Integrated Circuits and Signal Processing* 49 (1) (2006) 39–51.
- [2] I.F. Akyildiz, T. Melodia, K.R. Chowdhury, A survey on wireless multimedia sensor networks, *Computer Networks*, vol. 51, no. 4, Elsevier, 2007, pp. 921–960, 14.
- [3] Y. charfi, N. Wakamiya, M. Murata, Challenging issues in visual sensor networks, *IEEE Wireless Communications* 16 (2) (2009) 44–49.
- [4] Q. Lu, W. Luo, J. Wang, B. Chen, Low complexity and energy efficient image compression scheme for wireless sensor networks, *Elsevier, Computer Networks* 52 (13) (2008) 2594–2603.
- [5] S. Soro, W. Heinzelman, A survey of visual sensor networks, *Advances in Multimedia*, Article ID 640386, 2009, 21 pages.
- [6] M. Rahimi, R. Baer, O.I. Iroezi, J.C. Garcia, J. Warrior, D. Estrin, M. Srivastava, Cyclops: in situ image sensing and interpretation in wireless sensor networks, *ACM Conference on Embedded Networked Sensor Systems (SenSys 2005)*, ACM, San Diego (CA), USA, 2005, pp. 192–204.
- [7] A. Rowe, A. Goode, D. Goel, Nourbakhsh, et al., CMUcam3: an open programmable embedded vision sensor, *Technical Report RI-TR-07-13*, Carnegie Mellon Robotics Institute, Pittsburgh (PA), USA, May 2007.
- [8] C. Duran-Faundez, V. Lecuire, Error resilient image communication with chaotic pixel interleaving for wireless camera sensors, *Proceedings of the 2008 Workshop on Real World Wireless Sensor Networks (REALWSN'08)*, Glasgow, Scotland, 2008.
- [9] Sadler, M. Christopher, M. Martonosi, Data compression algorithms for energy-constrained devices in delay tolerant networks, *Proceedings of the ACM Conference on Embedded Networked Sensor Systems (SenSys)*, 2006.
- [10] L. Ferrigno, S. Marano, V. Paciello, A. Pietrosanto, Balancing computational and transmission power consumption in wireless image sensor networks, *IEEE International Conference on Virtual Environments, Human-Computer Interfaces and Measurement Systems (VCIMS 2005)*, Giardini Naxos, Italy, 2005.
- [11] L. Benini, D. Bruni, A. Macii, E. Macii, Hardware-assisted data compression for energy minimization in systems with embedded processors, *Proc. of the Conference on Design, Automation and Test in Europe, DATE 2002*, Nice, France, 2002.
- [12] G. Chrysos, I. Papaefstathiou, Heavily reducing WSNs' energy consumption by employing hardware-based compression, *ADHOC-NOW 2009, LNCS 5793*, Springer, 2009, pp. 312–326.
- [13] W. Yu, Z. Sahinoglu, A. Vetro, Energy efficient JPEG 2000 image transmission over wireless sensor networks, *Proc. IEEE GLOBECOM '04*, 2004.
- [14] V. Lecuire, C. Duran-Faundez, N. Krommenacker, Energy-efficient transmission of wavelet-based images in wireless sensor networks, *EURASIP J. Image Video Process* 2007 (2007), Article ID 47345, 11 pages.
- [15] H. Wu, A. Abouzeid, Error resilient image transport in wireless sensor networks, *Computer Network* 50 (2006) 2873–2887.
- [16] L. Makkaoui, V. Lecuire, J.-M. Moureaux, Fast zonal DCT-based image compression for wireless camera sensor networks, *IEEE International Conference on Image Processing Theory, Tools and Applications (IPTA 2010)*, Paris, France, July 2010.
- [17] D. Panigrahi, Clark N. Taylor, S. Dey, A hardware/software reconfigurable architecture for adaptative wireless image communication, *IEEE International Conference On VLCI, Design VLCID*, 2002.
- [18] F. Akyildiz, et al., A survey on sensor networks, *IEEE Commun. Mag.* 40 (Aug. 2002) 102–114.
- [19] H. Osman, W. Mahjoub, A. Nabih, G.M. Aly, JPEG encoder for low-cost FPGAs, *Computer Engineering & Systems, ICCECS '07*, 2007.
- [20] L. Volcan Agostini, I. Saraiva Silva, S. Bampi, Multiplierless and fully pipelined JPEG compression soft IP targeting FPGAs, *Microprocessors and Microsystems*, 31, Elsevier, 2007, pp. 487–497.
- [21] S.R. Mallireddy, S. Commuri, Run time compression of image data in wireless sensor networks, *IEEE International Conference Networking, Sensing and Control (ICNSC)*, 2010.
- [22] M.E. Papadonikolakis, A.P. Kakarountas, C.E. Goutis, Efficient high-performance implementation of JPEG-LS encoder, *J Real-Time Image Proc* 3 (2008) 303–310.
- [23] R. Zhou, L. Lu, S. Yin, A. Luo, X. Chen, S. Wei, A VLSI design of sensor node for wireless image sensor network, *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2010.
- [24] C. Loeffler, A. Ligtenberg, G. Moschytz, Practical fast 1-D DCT algorithms with 11 multiplications, *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 1989)*, Glasgow, UK, May 1989.
- [25] J. Liang, T.D. Tran, Fast multiplierless approximations of the DCT with the lifting scheme, *IEEE Trans. on Signal Processing* 49 (12) (2001) 3032–3044.

- [26] H. Jeong, J. Kim, W.-K. Cho, Correlation low-power multiplierless DCT architecture using image, *IEEE Trans. on Consumer Electronics* 50 (1) (2004) 262–267.
- [27] B. Heyne, C.C. Sun, J. Goetze, S.J. Ruan, A computationally efficient high-quality cordic based DCT, *European Signal Processing Conference (EUSIPCO 2006)*, Proceedings, Florence, Italy, September 2006.
- [28] B. Heyne, J. Gotze, A low-power and high-quality implementation of the discrete cosine, *Advances in Radio Science* 5 (2007) 305–311.
- [29] International Organization for Standardization, IUT-T Recommendation T.81, in: ISO/IEC IS 10918-1, <http://www.jpeg.org/jpeg>.
- [30] G.H. H'ng, M.F.M. Salleh, Z.A. Halim, Golomb coding implementation in FPGA, *ELEKTRIKA* 10 (2) (2008) 36–40.
- [31] P.G. Howard, J.S. Vitter, Practical implementations of arithmetic coding, *Image and Text Compression*, Norwell, MA, 1992, pp. 85–112.
- [32] http://category.alldatasheet.com/index.jsp?sSearchword=SRAM%20ST_SPHS_16384X8M16_L.
- [33] S. Hsien, S.J. Lee, A JPEG chip for image compression and decompression, *Journal of VLSI Signal Processing* 35 (2003) 43–60.
- [34] G. Seroussi, M.J. Weinberg, On adaptive strategies for an extended family of Golomb-type codes, *Proceedings of the Data Compression Conference*, 1997, pp. 131–140.
- [35] S.W. Golomb, Run-length encodings, *IEEE Trans. Inform. Theory* IT-12 (July 1966) 399–401.
- [36] <http://www.binaryessence.com/dct/en000119.htm>.
- [37] G. Voyatzis, I. Pitas, Chaotic mixing of digital images and applications to watermarking, *Proceedings of European Conference on Multimedia Applications, Services and Techniques (ECMAST'96)*, volume 2, Louvain-la-Neuve, Belgium, May 1996, pp. 687–695.
- [38] A. Amara, F. Amiel, T. Ea, FPGA Vs ASIC for low power applications, Elsevier, *Microelectronics Journal* 37 (8) (2006) 669–677.
- [39] www.xilinx.com.
- [40] L.V. Agostini, R.C. Porto, S. Bampi, I.S. Silva, A FPGA based design of a multiplierless and fully pipelined JPEG compressor, *Proceedings of Euromicro conference on digital system design (DSD'05)*, Porto, Portugal, August–September 2005, pp. 210–213.
- [41] M. Lin, L. Dung, P. Weng, An ultra low power image compressor for capsule endoscope, *Biomedical Engineering Online* 5 (2006) 14.
- [42] C. Duran-Faundez, "Transmission d'images sur les réseaux de capteurs sans fil sous la contrainte de l'énergie," *thèse, Université Henri Poincaré, Centre de Recherche en Automatique de Nancy*, (juin 2009) 129–132.
- [43] Crossbow Technology Inc., <http://www.xbow.com/>.



Vincent Lecuire earned his Ph.D. in Computer Science from the Université Henri Poincaré (Nancy, France) in 1994. He is currently an Associate Professor at the Université de Lorraine (France) and is a member of the Centre de Recherche en Automatique de Nancy (CRAN, CNRS). His research interests are in the area of image communication over large-scale IP networks, mobile ad hoc networks and wireless sensor networks, with an emphasis on packet-level forward error correction, delay and energy-aware protocols, robust image coding, and cross-layer design.



Khaldoun Torki received in 1990 his Ph.D. degree in Microelectronics from the Institut National Polytechnique de Grenoble, France. Since then, he joined CMP as a senior engineer, later on joining CNRS in 1994. He is currently the Technical Director at CMP since 2002. His research interest includes Deep Submicron design methodologies, Thermal simulation at IC level, Magnetic/CMOS integration (MRAM), and 3D-IC integration. He authored and co-authored more than 80 scientific papers, designed more than 30 ASIC circuits, and participated or coordinated 15 European and National projects.



Leila Makkaoui received her M.S. degree in Computer Science from the Ecole Polytechnique of TOURS University, (France) in 2007. She is currently completing her Ph.D. degree in automatic, signal processing and computer engineering with the Centre de Recherche en Automatique de Nancy (CRAN, CNRS) at the University de Lorraine. Her research interests include signal and image processing in wireless sensor networks.



Jean-Marie Moureaux received his Ph.D. degree in Electrical Engineering from the University of Nice-Sophia Antipolis, France in 1994 and the "Habilitation à Diriger des Recherches" from the University Henri Poincaré, Nancy 1 in December 2007. He is currently a Professor at the University Henri Poincaré, Nancy 1, France. Since 1995, he is with the Centre de Recherche en Automatique de Nancy (CRAN, Nancy Université, UMR CNRS 7039). His research interests are image and video coding, watermarking, lattice vector quantization, medical image compression and image transport over wireless sensor networks.



Med Lassaad Kaddachi received his M.S degree (2007) in Electronic and Microelectronics from the Monastir University. He is currently completing his Ph.D. degree at the Electronic and Micro-Electronic Laboratory (LAB-IT06), Sciences Faculty of Monastir, Tunisia. His research interests are QoS evaluation in Network on chip, design and performances evaluation of circuits for communication devices and robust image coding.



Adel Soudani received his Ph.D. (2003) in Electronic and also Electrical Engineering respectively from the University of Monastir, Tunisia, and the University of Henri Poincaré Nancy I, France. He is currently an Assistant Professor at the Institute of Applied Sciences and Technology of Sousse. His research activity includes QoS management in real time embedded system and multimedia applications. He focuses mainly on the design and performances evaluation of hardware solutions for communication systems with multiple constraints.