International Scholarly Research Network
ISRN Sensor Networks
Volume 2012, Article ID 760320, 19 pages
doi:10.5402/2012/760320

Review Article

A Survey of Image Compression Algorithms for Visual Sensor Networks

Abdelhamid Mammeri, Brahim Hadjou, and Ahmed Khoumsi

Department of Electrical and Computer Engineering, University of Sherbrooke, Sherbrooke, QC, Canada J1K 2R1

Correspondence should be addressed to Abdelhamid Mammeri, abdelhamid.mammeri@usherbrooke.ca

Received 3 September 2012; Accepted 22 October 2012

Academic Editors: A. Rezgui and A. Song

With the advent of visual sensor networks (VSNs), energy-aware compression algorithms have gained wide attention. That is, new strategies and mechanisms for power-efficient image compression algorithms are developed, since the application of the conventional methods is not always energy beneficial. In this paper, we provide a survey of image compression algorithms for visual sensor networks, ranging from the conventional standards such as JPEG and JPEG2000 to a new compression method, for example, compressive sensing. We provide the advantages and shortcomings of the application of these algorithms in VSN, a literature review of their application in VSN, as well as an open research issue for each compression standard/method. Moreover, factors influencing the design of compression algorithms in the context of VSN are presented. We conclude by some guidelines which concern the design of a compression method for VSN.

1. Introduction

Recent advances in microelectromechanical systems, wireless communication technology together with low-cost digital imaging cameras, have made it conceivable to build in an ad hoc way a wireless network of visual sensors (VSs), called visual sensor network (VSN). Inside a VSN, each VS node has the ability to acquire, compress, and transmit relevant frames to the base station, also called sink, through the path between the source and the sink; see Figure 1. Generally, the base station is defined as a powerful collecting information node located far away from the other (nonpowerful) nodes. Such networks have a myriad of potential applications, ranging from gathering visual information from harsh environment to monitoring and assisting elderly peoples [1].

Unlike classical wired networks and scalar data wireless sensor networks (WSNs), VSN faces new additional challenges. Compared to conventional wired networks, VSNs encounter more problems due to their inherent wireless nature and the resource constrained of VS. VSNs differ from their predecessor's scalar WSN basically in the following points. (1) The nature and the volume of visual flows, which are pixel based, are quite different from simple scalar data manipulated by WSN, such as temperature or humidity. (2) VSN's cameras have a restricted directional sensing field of view, which is not the case for scalar data sensor. (3) Contrary to WSN, important resources in memory, processing, and communication power are required for VS nodes to manipulate visual flows. (4) Energy-aware compression algorithms are mandatory to handle images, compared to data scalar sensor where the compression is not required.

Typically, compression is performed by exploiting data correlation and redundancy. In VSN, three scenarios of data redundancy are observed. First, redundancy between successive frames captured by the same sensor within an interval of time, which is known as interimage redundancy or temporal redundancy. Second, redundancy between neighboring sensors monitoring the same scene which is also called interimage redundancy. Finally, redundancy between neighboring pixel values of an image, called spatial redundancy. In case of color image, we note the existence of a fourth type of redundancy, called spectral redundancy.

A few number of related review papers have been proposed in the literature [1–5]. An extensive survey of wireless multimedia sensor networks is provided in [1], where the state of the art in algorithms and protocols at the application,

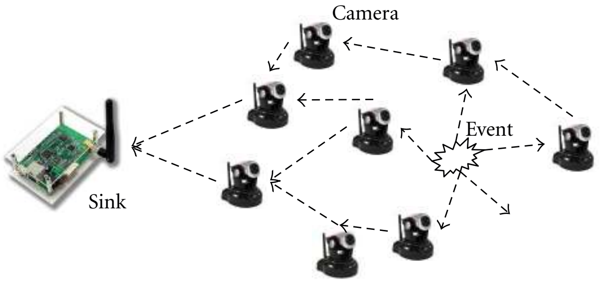


Figure 1: Visual sensor network.

transport, network, link, and physical layers of the communication protocol stack are investigated. Open research issues are discussed at each layer. Moreover, architecture and hardware for wireless multimedia sensor networks are supplied and classified. The authors concentrate only on recent advances on low complexity encoders based on Wyner-Ziv coding. In [2], the authors present a survey on multimedia communication in WSN with a main focus on the network layer, the application layer, and some considerations on the transport layer. The authors in [2] do not discuss deeply the compression algorithm, where they consider only the DSC paradigm. The authors in [3] complement their successors in [2] by categorizing the requirements of multimedia streams at each layer of the communication protocol stack and survey cross-layer mechanisms for multimedia streaming. Moreover, they outline some future research directions at each layer of the stack as well as for a cross-layer scheme. Their work is not compression oriented. They consider only some compression algorithms proposed in the literature. Another work is suggested in [5], where the authors present an overview on several challenging issues influencing the design of VSN, such as network architectures and energy-aware communication and processing scheme. In the same context, the authors in [4] provide an overview of the current state of the art in VSN and explore several relevant research directions.

While the aforementioned studies have considered some VSN aspects including the requirements of multimedia streams at each layer of the communication protocol stack and cross-layer synergies and optimizations, only few of them (e.g., [1, 3]) have considered some aspects around image compression, and none of them have discussed the compressive sensing-based algorithms for VSN or Fractals imaging for VSN. In this survey paper, we focus on the state of the art in image compression and point out different compression methods, ranging from the conventional standards (JPEG and JPEG2000), and their application in VSN, to a new compression methods including compressive sensing. More precisely, we focus on individual source coding (ISC) schemes, while the distributed source coding (DSC) methods are given little explanation (see [1, 3] for more details). Our survey complements the aforementioned surveys as follows:

(1) we survey and classify the ISC compression methods suggested in the literature,

(2) we introduce some notions behind the compressive sensing, and its possible application to VSN,

(3) we provide a brief overview of each compression method, the advantages and shortcomings of their application in VSN, a literature review of their application in VSN, as well as an open research issue for each compression method,

(4) we conclude by some guidelines which concern the design of a compression method for VSN.

This paper is structured as follows. In Section 2, we discuss some requirements and characteristics of VSN, then we study the relationship between compression and transmission costs, and after that we suggest the classification of the main individual compression algorithms. In Section 3, we present the main idea behind DCT and some related compression algorithms in the context of VSN. The explanations of DWT, DWT-based schemes such as EZW, SPIHT, EBCOT, or SPECK, and their applications in VSN are presented in Section 4. The non-transform-based algorithms including vector quantization, Fractal compression, and their introduction in VSN is explained in Sections 5.1 and 5.2, respectively. The distributed source coding paradigm as well as some research works incorporating this paradigm in VSN is presented in Section 7. In Section 8, another paradigm called compressive sensing is presented, along with some applications in VSN context. Some guidelines for designing a compression scheme for VSN are presented in Section 9. Finally, we conclude this paper by Section 10.

2. Overview of Image Compression for VSN

This section provides some background information to follow this paper. Recall that VSNs are spatially distributed networks consisting of small sensing devices equipped with low-power CMOS imaging sensors such as Cyclops. Ideally, VSNs are deployed in the region of interest to collect and transmit data in multi-hop way. VSNs are involved in many domains such as environmental monitoring, video surveillance, and object detection and tracking.

VSNs differ from their predecessor's scalar data WSN mainly in the following.

(i) Information volume and nature of VSN, which is in general pixel based, is quite different from simple scalar data manipulated by WSN, such as temperature.

(ii) Lost information in VSNs is tolerated due to the redundancy nature of visual flows. Whereas in WSN, the loss of some packets may affect seriously the value of collected data (e.g., temperature value).

(iii) VSN's camera has a restricted directional sensing field of view, which is not the case for scalar data sensors.

(iv) VS neighbors monitoring the same small local region of interest have multiple and different views of this scene, compared to scalar data sensor where a unique value (e.g., temperature) is collected by neighbor's nodes (situated in the same region).

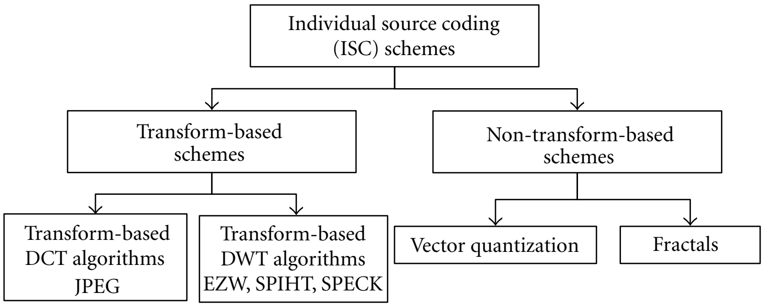

Individual source coding (ISC) schemes
Transform-based schemes
Non-transform-based schemes
Transform-based DCT algorithms JPEG
Transform-based DWT algorithms EZW, SPIHT, SPECK
Vector quantization
Fractals

Figure 2: Classification of ISC algorithms.

(v) Important resources in memory, processing, and communication power are required for VS node to manipulate the huge amount of visual flows.

(vi) Efficient compression algorithms, in terms of power dissipation, are mandatory to handle information flows, compared to scalar data sensor where the compression is not very required.

Most significant studies in scalar data WSN have typically assumed the computational costs, including acquisition and compression, insignificant compared to the related communication costs (e.g., [6]). This assumption may be suitable for scalar data sensors, where the computation cost of data compression, if performed, is negligible compared to the communication cost.

In case of WSN handling images (or video), this assumption may not hold, since visual flows always necessitate compression. In this section, we show the relationship between the compression cost and the transmission cost in the context of VSN. Deciding to precede transmission by compression or not depends mainly on the specific compression algorithm, and possibly on the processor and the radio transceiver if we include the time factor.

Usually, image transmission preceded by compression is the ideal choice to gain in time and energy. It is well known that some compression algorithms are more time and energy consuming than others. Those algorithms are, generally, used for storage purpose, or also used when no power or time restrictions are required. For instance, compression using Fractals or JPEG2000 is very time and energy consuming [7], and their applications to VSN seem less efficient. However, when applied to traditional wired networks, JPEG2000 gives the highest compression ratio regardless of the consumed energy. Another example is described in [8], where the authors have shown that compressing, using JPEG, and transmitting an image is more energy inefficient than transmitting the uncompressed image at higher quality level. In such a case, compression is not justified, since the transmission of the uncompressed image consumes less energy.

Different image compression classifications are found in the literature. In general, they are categorized in terms of data loss, or whether they use a transform coding or predictive coding [9]. Our goal is not to survey all of them, but rather we review those ISC algorithms that their applications in VSN domain seem practical. In particular, basic algorithms for coding images are still considered. Based on the requirements of reconstruction, image compression schemes are commonly provided into two categories: lossless and lossy scheme. Lossless image compression algorithms refer to the perfect reconstruction of the original image from the compressed one. On the other hand, with lossy image compression scheme, merely an approximation of the original image is achieved. The main benefit of lossy image compression algorithm over lossless one is to gain in encoding/decoding time, compression ratio [9], or also in case of power-constrained applications, in energy. That is, we believe that lossy schemes are highly encouraged in VSN, compared to lossless techniques. However, if lossy and lossless compression algorithms yield the same results in terms of power dissipation, lossless algorithms are encouraged.

We regroup the ISC algorithms discussed in this paper into two categories: transform-based algorithms, such as discrete cosine transform- (DCT-) and discrete wavelet transform- (DWT-) based algorithms, and non transform-based algorithms, such as vector quantization or fractals; see Figure 2. We note that the typical design of a transform-based algorithm is based on three stages: spatial decorrelation (also called source encoder), followed by quantizer, and entropy encoder. Other schemes (non-transform-based algorithms) such as vector quantization or fractals do not follow this design.

3. Transform-Based DCT Algorithms

Before reviewing the main DCT-based algorithms found in the literature, we briefly describe the principal idea behind DCT. The DCT is a technique for converting a signal into elementary frequency components. The image is decomposed into several blocks, and for each block, DCT is mathematically expressed as a sum of cosine functions oscillating at different frequencies. Since we concentrate on images, we consider only the two dimensional representation of DCT (2D DCT), which can be obtained from the cascade of two 1D DCTs.

The well-known compression scheme based on DCT is the standard JPEG [10]. In this survey paper, JPEG is analyzed in the context of power-constrained application. Other variants of the compression scheme based on DCT are proposed in the literature to enhance JPEG features, such as minimizing the blocking artifacts, minimizing the complexity at the encoder and/or the decoder, and increasing the compression ratio.

Since the DCT transform consumes the most power within a DCT-based compression scheme (more than 60% of the computation cost of the JPEG algorithm [11]), many attempts to decrease its computational complexity have been suggested in the literature. Some of them, which are helpful for VSN designers, are cited as follows.

(1) Parallel and pipelined implementation of multidimensional DCT: the authors in [12] use a parallel and pipelined row-column decomposition method based on two 1D DCT processors and an intermediate buffer. The proposed architecture allows the main processing elements and arithmetic units to operate in parallel, which reduce both the computational complexity and the internal storage, and allows a high throughput [12]. The same idea is explored in [13] with the integer cosine transform (a reduced computational complexity version of DCT) to further reduce the computational complexity of the whole system. To the best of our knowledge, the exploration of parallel and pipelined implementation of 2D DCT has not yet been investigated in VSN.

(2) Working with fixed-point instead of the more complicated floating-point DCT: compared to fixed-point DCT, working with the floating-point DCT exhibits high energy consumption. For illustration purpose, let us consider the following example from [14]. Encoding a grayscale QCIF image at 1 bit-per-pixel using the processor StrongARM SA1110 with JPEG-integer-point DCT requires 2.87 mJ. The same operation using floating-point DCT necessitates more than 22 mJ. This justifies the possible choice of fixed-point DCT over floating-point DCT in case of VSN.

(3) Converting the greedy operations such as multiplications into light operations: indeed, DCT can be implemented using light operations such as shifts and additions only. For instance in [15], a multiplierless version of DCT based only on shift and addition operations is suggested. This scheme enables low-cost and fast implementation compared to the original DCT, due to the elimination of the multiplication operations [15].

In the following section, we introduce JPEG, the well-known DCT-based scheme, its advantages and shortcoming, as well as a discussion about its possible application in VSN.

3.1. JPEG Background. The process of baseline JPEG compression consists of the following stages. First, the input image is divided into several blocks of fixed size 8 × 8 pixels, and, then, the DCT is applied to each block to separate the high and low frequency information. In order to compress an image, the DCT blocks are quantized uniformly. The quantization result is then reordered in zigzag way from lower to higher frequencies. After that, the run-length encoding (RLE) is applied to reduce the length of the generated sequences. Finally, the reversible entropy-coding process (such as Huffman or arithmetic coding) is performed on the quantized data to generate fixed or variable length codewords [10] (Figure 3).

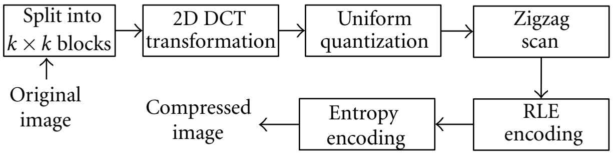


Figure 3: Lossy JPEG compression scheme.

DCT-based image compression provides acceptable compression results, and it gives a low memory implementation, since the encoding is done on small individual blocks of size 8 × 8 pixels. However, blocks tiling (which is the process of splitting the original image into several blocks) causes blocking artifacts which lead to a degradation in performance especially at very low bit rates.

3.2. DCT-Based Methods for VSN. The adoption of JPEG as a compression tool in VSN is not very beneficial in terms of power consumption [16]. This is due to the relatively complex coder, and precisely to the DCT stage which consumes at least 60% of the whole power encoder. Our preliminary studies on JPEG show the possibility of its application as a compression tool for VSN images at the cost of decreasing the network lifetime [16]. Another confirmation comes from [17], where the authors show the possibility to integrate successfully JPEG as a compression scheme for VSN.

In what follows, we briefly present the main DCT-based schemes for VSN. We begin this section by the work presented in [18], where the authors study the problem of compression of video-surveillance frames collected by WSN. They use an algorithm to build a map of active regions within one frame, and, then, they encode these regions. In particular, each input frame is divided into blocks of 8 × 8 pixels. In order to decrease the complexity, only a subset of blocks in the frame are considered, and only a subset of the pixels in each block are classified in the order of their importance. These pixels are then examined for changes in comparison to the corresponding pixels in the reference frame. Only the difference is encoded using JPEG-like scheme. In particular, a fast integer DCT and Golomb-Rice codes are used, since they exhibit low complexity and less power dissipation.

The authors in [8] suggest an energy-optimized approach ensuring that the JPEG computations utilize the minimum precision needed to obtain optimized DCT and quantization. To accomplish this, they develop a method that determines the optimum integer and fractional bit widths in the compression process which guarantees the required precision. Moreover, to be fast in computations, the authors use the LLM algorithm developed in [19]. We mention that the authors in [8] implement the JPEG computations in fixed-point arithmetic instead of floating-point representation for energy considerations. Several experimentations are performed using various processors to measure the energy savings resulting from the precision optimization process. These processors are Atmel ATmega128, TI MSP430, TI TMS320C64x, and Analog Devices Blackfin ADSPBF533. The suggested method outperforms JPEG in terms of speed and energy. The authors observe that the Atmel ATmega128 consumes the highest energy compared to the processors under evaluation.

The work in [14] investigates the trade-off between image quality and power consumption in wireless video-surveillance networks. To reduce computation energy dissipation, they use JPEG with integer DCT kernel instead of the commonly used floating-point DCT. Moreover, they discuss lightly the impact of compression on image delay using ARQ scheme implemented at the link layer [14]. In [20], the same authors in [14] investigate the interactions between energy consumption, quality, and delay and analyze the system performance when ARQ and FEC-based error-control techniques are applied. As in [14], they use JPEG with integer DCT to reduce computation energy dissipation. Precisely, they investigate ARQ and FEC as error-recovery techniques and propose an adaptive scheme, which selects the most appropriate error-control technique depending on the channel propagation conditions.

An image sensor network platform was developed in [17] to show the feasibility of transmitting the compressed images over multi-hop sensor networks supporting ZigBee technology. The compression was performed using the standards JPEG and JPEG2000. The comparison between the two standards was performed only in terms of tolerance to bit errors and packet losses. The authors observed that JPEG2000 is more error-resilient than JPEG, while having the highest PSNR for the same compression ratio. Hence, they conclude that JPEG2000 is a more suitable standard for image compression in VSN in terms of packet losses. We highlight that the predominant design factor in VSN, that is, energy, was not considered, and their evaluation seems to be not practical for all VSN applications, especially outdoor applications.

The authors in [21] study the trade-off between energy consumption and image quality when different routing paths are used. Particularly, they study the effect of the proximity of the sink on the energy overhead. For compression purpose, the authors use the standard JPEG. To control the compression rate, the quality level parameter is used. The higher the quality level, the better the image quality, but with a larger file size. To reduce the image quality the number of quantization levels is reduced.

Contrary to [21], where the authors study the trade-off between energy consumption and image quality, in [22] they deal with the trade-off between energy consumption and the covered viewing directions in VSN. A selective transmission protocol is developed to select and transmit images to the mobile sink in an energy efficient way. To do that, similarity score between images is computed and compared. To perform similarity between images, nodes perform feature extraction procedure on the captured images. To save transmission energy, only the image having the most number of feature points among the similar images will be transmitted. For compression purpose, the authors in [22] use the standard JPEG. As in [21], the authors study the effect of the proximity of the sink on the energy overhead. The simulation results show that the protocol can achieve a significant reduction in energy consumption while preserving most of the views. Despite the saving in energy transmission, feature extraction and comparison seems to be energy consuming. Moreover, the authors do not compare the transmission with and without feature extraction.

The aim of [23] is the reduction of the transmission energy through the selection of appropriate paths and appropriate compression of images. They use the standard JPEG in the compression stage. First, they demonstrate that the amounts of energy required in different forwarding paths are different. Then, they develop an algorithm to select a path that requires the least energy.

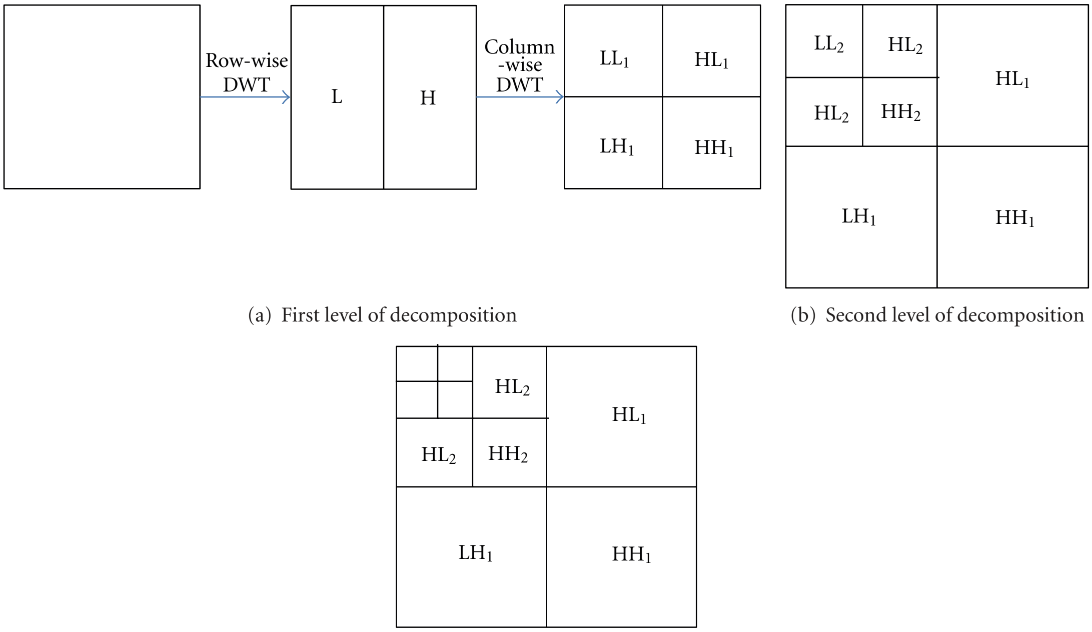
The authors in [24] present an analysis of both power requirement and execution time of the basic tasks that compose a typical duty cycle of a camera node within a real VSN. For that reason, a Crossbow Stargate platform is used along with Logitech QuickCam Pro 4000 webcam. Among tasks considered in [24], we cite acquisition and compression. The authors use JPEG as a compression standard to compress images or subimages. Each considered task has an associated power dissipation cost and execution time. Numerous interesting results are observed. For instance, the time needed to acquire and compress an image is 2.5 times larger than that of the transmission of the compressed image. The authors also show that the transmission and reception consume about the same amount of energy. Moreover, the power cost of analyzing an image, and compressing a subimage, is about the same as compressing the whole image.

Another interesting work is presented in [25], where the authors address the problem of reducing energy consumption. The authors' aim is to find the most efficient compression algorithm achieving the best compromise between the quality of the reconstructed image and the energy consumption. Their analysis is conducted from the measurements results of the current consumption for each state: standby, sensing, processing, connection, and communication. For that reason, several compression methods are considered, namely, JPEG, JPEG2000, SPIHT, and subsampling. They realize that the most appropriate compression methods are SPIHT, which gives the highest compression rate, and subsampling, which requires the smallest execution time.

In the following section, we present the alternative solution to DCT, that is, DWT, which represents a promising technique for image compression.

4. Transform-Based DWT Methods

We start this section by a short introduction on wavelets. Basically, the wavelet was developed to overcome the weakness of the short time Fourier transform and to enhance DCT features, such as localization in time and frequency. We consider in this paper the 2D DWT representation, as we work with images. Since, in general, the 2D wavelets used in image compression are separable functions, their implementation can be obtained by first applying the 1D-DWT row wise to



(a) First level of decomposition

(b) Second level of decomposition

(c) Third level of decomposition

Figure 4: Three levels of decomposition of 2D-DWT.

produce L and H subbands, and then column wise to produce four subbands LL, LH, HL, and HH. Then, in a second level, each of these four subbands is itself decomposed into four subbands, and so on we can decompose into 3, 4,... levels. Figure 4 illustrates the decomposition of the LL subband.

The DWT is widely considered to yield the best performance for image compression for the following reasons. It is a non-block-based transform, and, thus, it allows avoiding the annoying blocking artifacts introduced by the DCT transform within the reconstructed image. Moreover, it has a good localization in both time (space) and frequency domains [26].

A variety of wavelet-based image compression schemes have been developed due to their usefulness for signal energy compaction. In this paper, we discuss some well-known algorithms such as EZW, SPIHT, EBCOT, and SPECK and their advantages and shortcomings, as well as their applications in VSN.

4.1. EZW-Based Image Compression

4.1.1. EZW Background. In this section, we roughly present the main idea of EZW, more details can be found in [27]. EZW algorithm starts by performing the wavelet decomposition on the input image, which allows its decomposition into a series of wavelets coefficients. The EZW algorithm assumes that if a coefficient magnitude at a certain level of decomposition is less than a threshold T, then all the coefficients of the same orientation in the same spatial location at lower scales of decomposition are not significant compared to T. A wavelet coefficient is said to be significant with respect to T if its absolute value is higher than or equal to T.

The EZW algorithm is a multiple-pass procedure, where each pass involves two steps: the dominant pass (or significance map encoding) and the subordinate pass (or refinement pass). In the dominant pass, the initial value of the threshold is chosen, against which all the wavelet magnitudes are compared. The coefficients are then encoded according to their values with respect to the fixed threshold. A wavelet coefficient (or its descendant) is encoded if its magnitude is greater than or equal to the threshold T, otherwise, it is processed as in [27]. Once a determination of significance is achieved, the subordinate pass is started. In this pass, the significant coefficients found in the dominant pass are quantized using successive approximation quantization approach. When all the wavelet coefficients have been scanned, the threshold is halved and the scanning process is repeated again, to add more detail to the already encoded image, until some rate is met.

The EZW method is a simple efficient compression algorithm. This is achieved through a combination of a hierarchical multiresolution wavelet transform and progressive zerotree encoding of wavelet coefficients, along with successive approximation quantization. The intrinsic progressive processing behavior lets the encoding process end at any point in time, which may help, in case of VSN, savings in power processing and communication. However, EZW presents some disadvantages. In fact, the number of passes required to compress an input image affects considerably the image quality and the VS power supporting EZW. That is, if the number of passes increases, the precision of the coefficients increases the full reconstructed image quality at the base station. Another shortcoming of EZW is related to the memory required to store the significant wavelet coefficients found at each pass. One solution to remove the need for this memory is to decrease the number of passes. Moreover, EZW is susceptible to transmission errors and packet losses, which require the introduction of an error correction models [28]. Another major drawback of EZW is that it does not present multiresolution scalability. It is well known that, in subband coders, the coefficients are transmitted progressively from low to high frequency, while with EZW, wavelet coefficients prioritization is performed according to their magnitudes [27].

4.1.2. EZW-Based Scheme for VSN. The unique research work adopting EZW as a compression tool for VSN is that one suggested in [29]. The authors in [29] suggest a multimodal sensor network architecture using acoustic, electromagnetic, and visual sensors, along with a satellite communication backbone. Based on the collaborative effort of this architecture, the target position is recognized, and its fine details are acquired using visual sensors. For this purpose, EZW coding algorithm is adapted to VSN requirements. This is performed by introducing spatial information about target activity. The adapted EZW provides high-resolution data for the regions where one or more intrusions have been detected and low-resolution data for the remaining regions. This scheme allows saving in bandwidth, power, and storage resources.

The adoption of the EZW as a compression tool in VSN can be beneficial in terms of power consumption. This is due to the relatively simple complexity of its encoder and its progressive paradigm. An open research work should be the adaptation of the EZW algorithm to the power-constrained VSN. This is can be performed by minimizing the number of passes to minimize the memory required to store the significant wavelet coefficients found at each pass.

4.2. SPIHT-Based Image Compression

4.2.1. SPIHT Background. SPIHT introduced in [30] is an improvement of EZW algorithm. By adopting set partitioning algorithm and exploring self-similarity across different scales in an image wavelet transform, SPIHT algorithm reaches high compression performance. Unlike EZW, SPIHT maintains three linked lists and four sets of wavelet coordinates, which are deeply explained in [30]. With SPIHT, the image is first wavelet decomposed into a series of wavelet coefficients. Those coefficients are then grouped into sets known as spatial orientation trees. After that, the coefficients in each spatial orientation tree are encoded progressively from the most significant bit planes to the least significant bit planes, starting with the coefficients with the highest magnitude. As with EZW, the SPIHT algorithm involves two coding passes: the sorting pass and the refinement pass. The sorting pass looks for zerotrees and sorts significant and insignificant coefficients with respect to a given threshold. And the refinement pass sends the precision bits of the significant coefficients. After one sorting pass and one refinement pass, which can be considered as one scan pass, the threshold T is halved, and the coding process is repeated until the expected bit rate is achieved.

SPIHT achieves very compact output bitstream and low bit rate than that of its predecessor's EZW without adding an entropy encoder, which allows its efficiency in terms of computational complexity [30]. Moreover, it uses a subset partitioning scheme in the sorting pass to reduce the number of magnitude comparisons, which also decrease the computational complexity of the algorithm. Finally, the progressive mode of SPIHT allows the interruption of coding/decoding process at any stage of the compression [30]. Despite these advantages, SPIHT presents the following shortcomings, particularly in power-constrained applications. It requires important memory storage and sorting/list procedures, which increases the complexity and the computational complexity. Precisely, SPIHT uses three lists to store coding information which needs large memory storage. In general, those lists grow up with the encoding process, which requires additional memory. Furthermore, the wavelet filter used in SPIHT is Mallat algorithm based, which incurs large convolution computations compared to lifting scheme version of wavelet transforms. As with EZW, over unreliable networks, SPIHT suffers from the network state and, thus, is vulnerable against packets loss, which requires the use of an appropriate error correction scheme.

Many attempts to enhance SPIHT features and reduce its limitations have been suggested in the literature, for instance [31–33]. In [31], the authors apply the concept of network-conscious image compression to the SPIHT algorithm to improve its performance under lossy conditions. Hence, SPIHT-NC (a network-conscious version of SPIHT) is suggested to enhance its performance over unreliable networks. A real-time implementation of SPIHT is presented in [32]. The authors try to speed up the SPIHT process and reduce the internal memory usage by optimizing the program structure and presenting two concept numbers of error bits and absolute zerotree. An improved zerotree structure and a new coding procedure are adopted in [32] to improve the quality of the reconstructed image by SPIHT. To further reduce the internal memory usage, the authors suggest a listless version of SPIHT, where lists are replaced successfully by flag maps. Moreover, a wavelet lifting scheme is adopted to speed up the coding process. A modified SPIHT algorithm for real-time image compression, which requires less execution time and less memory usage than SPIHT, is presented in [33]. Instead of three lists, the authors use merely one list to store the coordinates of wavelet coefficients, and they merge the sorting pass and the refinement pass together as one scan pass.

4.2.2. SPIHT-Based Schemes for VSN. We start by the compression method proposed in [34], where the basic design idea is drawn from the following observation. It is more efficient to send a very long bitstream in small decomposed fragments or bursts than their transmission as one entire block. That is, the suggested scheme in [34] uses wavelet-based decomposition strategy to create multiple bitstream image encodings which are sent in small bursts. The wavelet coefficients are grouped into multiple trees and encoded separately using SPIHT algorithm. The unequal error protection method is also adopted in order to combat time-varying

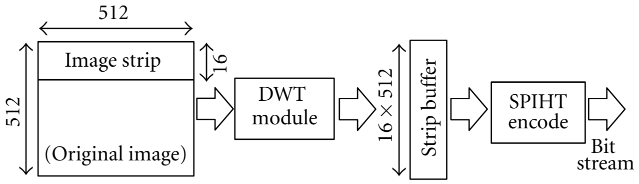


Figure 5: Block diagram of proposed strip-based image compression [35].

channel errors. Experimental results show that the proposed scheme has a good energy efficiency in transmission.

Another work incorporating SPIHT as a compression tool is presented in [35]. The authors use a strip-based processing technique where an image is divided into strips which are encoded separately. Figure 5 shows the block diagram of this suggested method. First, a few lines of image data are wavelet decomposed by DWT module. The lifting-based 5/3 DWT is used for this purpose. After that, the wavelet coefficients are computed and then buffered in a strip buffer. Finally, the bitstream generated is transmitted. The proposed SPIHT coding eliminates the use of lists in its set-partitioning approach.

The idea behind the framework developed in [36] is the use of image stitching in conjunction with SPIHT coding to remove the overlap and spatial redundancy. Image stitching can be defined as the process of combining multiple images with overlapping fields of view to create a segmented panorama or high-resolution image. Thus, the images taken by neighboring sensors are stitched together by certain intermediate nodes with an image stitching technique to remove the overlap redundancy. For compression purpose, a modified version of the SPIHT compression tool is used, which leads to the reduction in the amount of the transmitted data [36].

Implementing SPHIT on power-constrained devices, such as visual sensors, is an excellent idea. Its advantages over JPEG and EZW in terms of high compression ratio, less computational complexity, and low powerconsumption, as well as less complex implementation make it possible to play an interesting role in image compression for power-limited applications. An open research work should be the adaptation of the SPIHT algorithm to the power-constrained VSN. This can be performed by exploiting some ideas, like the substitution of lists by flags [32] to reduce the memory usage. An alternative idea is the use of wavelet lifting scheme instead of the convolutional based wavelet used by the original SPIHT [35].

4.3. EBCOT-Based Image Compression

4.3.1. EBCOT Background. EBCOT is a block-based encoding algorithm, where each subband (or block) is divided into nonoverlapping blocks of DWT coefficients called code blocks. Every code block is coded independently, which allows to generate a separate highly scalable embedded bitstream, rather than producing a single bitstream representing the whole image. As reported in [37], EBCOT, which represents the core functioning of the standard JPEG2000, is divided into two processes called Tier-1 and Tier-2, as shown in Figure 6. The data inputs of the Tier-1 process are code blocks while the outputs are bitstreams. Tier-1 is responsible for context formation and arithmetic encoding of the bitplane data and generates embedded block bitstreams. Context formation scans all code block pixels in a specific way as explained in [37]. The context formation requires three passes: significance propagation pass, magnitude refinement pass, and clean-up pass. Arithmetic encoding module encodes the code block data according to their contexts generated during context formation. Tier-2 operates on the bitstreams generated from Tier-1 to arrange their contributions in different quality layers. This is performed according to rate-distortion optimized property and features specified by the user. At the end of the second tier, a compressed bitstream is generated for transmission purpose.

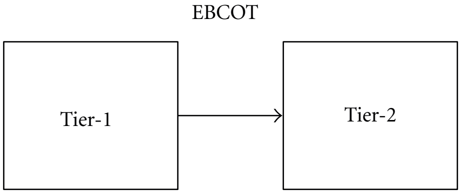


Figure 6: Illustration of EBCOT.

Table 1: Run time (%) profile for JPEG-2000 (gray scale image 1792 × 1200), 5 level wavelet decomposition, 1 layer, profile at PIII-7333 128 M RAM. (N.A.: not attributed.)

Operation	Lossless compression	Lossy compression
DWT	10.81	26.38
Quantization	N.A	6.42
EBCOT Tier-1	71.63	52.26
EBCOT Tier-2	17.56	14.95

EBCOT is a scalable and efficient compression algorithm, robust against error transmission, and has a flexible organization and arrangement of bitstreams [37]. Nevertheless, the EBCOT algorithm requires additional memory requirement, which increases the power dissipation and the computational complexity. Precisely, EBCOT uses two tiers (Tier-1 and Tier-2) to code information, which needs long time processing and high power consumption. In particular, context formation phase which includes three passes takes a long time to encode samples for a code block [38]. It is observed in [39] that Tier-1 is the most computational intensive part, due to the fact that it requires significant bit-level processing and three separate passes through the code blocks. It is reported in [40] that Tier-1 accounts for more than 70% of encoding time, due to the extensive bit-level processing, followed by the DWT transformation stage (see Table 1 for an example of both lossless and lossy compressions).

Recently, efficient techniques have been suggested to improve the coding speed and to minimize the memory usage of EBCOT, for example, [38, 41, 42]. Almost all of them focus on the enhancement of the context formation phase by different ways. As our goal is not to survey all the techniques suggested in the general domain of digital imaging, we provide some research works that can be used to minimize the EBCOT power consumption in VSN. For instance, two speed-up methods called sample skipping and group-of-column skipping were proposed to accelerate the encoding process of EBCOT [41]. Another interesting architecture is proposed in [42], where the authors merge the three coding passes into a single pass in order to improve the overall system performance as well as to reduce memory requirement. Further details on this subject can be found in [38, 41, 42].

4.3.2. EBCOT-Based Schemes for VSN. In this section, we review the main schemes adopting EBCOT (or JPEG2000) for compression purpose in VSN. We start by the architecture suggested in [43] which releases the visual sensors from the burden compression process, to prolong the network lifetime. Except the camera sensor, all data sensors are organized into clusters. The visual sensor does not join the cluster directly. Rather, it forms its own cluster and sends the target image to the cluster members. These members in the VS cluster, which belong to the data sensor clusters, share the task of image compression and transmission to the cluster head. Both computational and communication energy consumptions are considered in this architecture. For compression purposes, the authors in [43] use the standard JPEG2000, which increases rapidly the energy dissipation. By simulation, the authors show that this architecture can prolong the lifetime of the network.

The authors in [44] propose an energy efficient JPEG2000 scheme for image processing and transmission, given the expected end-to-end distortion constraint. In the suggested scheme, called joint source channel coding and power control (JSCCPC), the input image is firstly encoded as a scalable bitstream in an optimal number of layers. Based on the three following factors: the estimated channel condition, the characteristics of the image content, and the end-to-end distortion constraint, the suggested scheme determines adaptively the number of transmitted layers. Moreover, the JSCCPC unit adjusts the source coding rate, the source level error resilience scheme, the channel coding rate, and the transmitter power level for each layer. This approach extensively explores the multiresolution nature of bitstreams; however, the unequal importance between structure information and magnitude information is not fully identified. The authors show by simulations that up to 45% less energy consumption could be achieved under relatively severe channel conditions.

Another energy-aware scheme for efficient image compression for VSN is that one suggested in [45], where the authors formulate this challenging task as an optimization problem. They use JPEG2000 standard on a StrongArm SA-1000 processor. For a given image quality requirement and network conditions, the authors investigate a heuristic algorithm to select the optimal parameters of a wavelet-based coder, while minimizing the total energy dissipation. Results indicate that large fractions of the total energy are spent on computation due to the high complexity of JPEG2000. From [45], we can conclude that maximal compression before transmission may not always entail minimal energy consumption. However, their approaches mainly focus on power efficient techniques for individual components and cannot provide a favorable energy performance trade-off in the case of WSN.

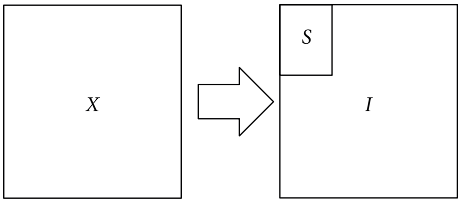


Figure 7: Partitioning of the image X into sets S and I.

Carrying out EBCOT or JPEG2000 in camera sensors may not always be the smart choice, since its implementation complexity induces high power consumption, where it is implemented (e.g., in VS), and possibly shrinks the network connectivity. Moreover, when combined with DWT stage (as with JPEG2000), more power will be dissipated due to the fact that DWT phase power consumption is significant and represents the second source consumption of an EBCOT-DWT compression scheme after Tier-1's EBCOT. An eventual open research work should be the adaptation of EBCOT to VSN constraints, taking advantage of some potential solutions to alleviate the workload and the complexity of the EBCOT algorithm.

4.4. SPECK-Based Image Compression

4.4.1. SPECK Background. SPECK is introduced in [46], where the authors suggest a compression algorithm that makes use of sets of pixels in the form of block when spanning wavelet subbands, instead of using trees as with EZW or SPIHT. SPECK algorithm starts by performing an appropriate subband transformation (usually, the DWT) on the input image, which allows its decomposition into a series of coefficients. After that, two phases are repeated recursively until the expected bit rate is achieved: sorting pass and refinement pass phase. Recall that SPECK necessitates three phases: initialization, sorting pass, and refinement pass phase. Unlike EZW, SPECK maintains two linked lists: list of insignificant sets (LISs) and list of significant pixels (LSPs).

During the initialization phase, a starting threshold T is chosen and the input image X is partitioned into two types of sets: S and I; see Figure 7. The set S, which represents the root of the pyramid, is added to LIS. The set I represents the rest of the image, that is, $I = X - S$. In the second phase called sorting pass, a significance test against the current threshold is performed to sort each block of type S in LIS. If an S block is significant, it is divided by a quadtree partitioning process into four subsets as shown in Figure 8. In turn, each of these four subsets is treated in the same way as a set of type S and processed recursively until the pixel level is reached. The insignificant sets are moved to LIS for further processing.

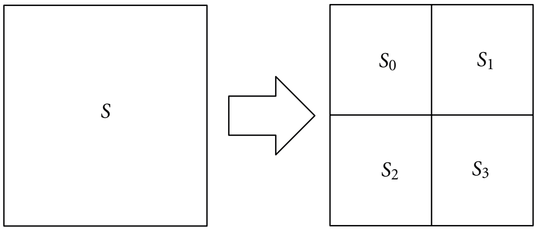


Figure 8: Partitioning scheme of set S.

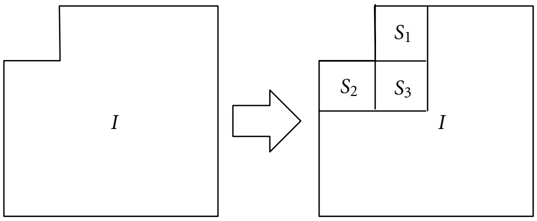


Figure 9: Partitioning scheme of set I.

Once the processing of sets S is achieved, a significance test against the same threshold is performed for I blocks. Thus, if an I block is significant, it is divided by the octave band partitioning scheme into four sets, one set having the same type I and three sets of type S; see Figure 9. This new set I formed by this partitioning process is reduced in size.

At the last phase, the refinement pass is started for LSP pixels, where the nth most significant bit (MSB) of pixels is output, at the exception of pixels which have been added during the last sorting pass. Finally, the threshold is halved, and the coding process (sorting and refinement passes) is repeated until the expected bit rate is achieved, or the set I will be empty.

Many advantages of SPECK are observed. It has efficient performance compared to the other low complexity algorithms available today. In fact, it gives higher compression ratio, has relatively low dynamic memory requirements; employs progressive transmission, and has low computational complexity and fast encoding/decoding process, due to the inherent characteristics of the quadtree partitioning scheme.

However, SPECK presents some minor disadvantages related mainly to the use of lists LIS and LSP, which require efficient memory management plan. In general, those lists grow up with the encoding process, which requires additional memory. This may be unattractive in hardware implementations. As with EZW and SPIHT, SPECK suffers from the unreliable network state and, thus, is vulnerable against packets loss which requires the use of an appropriate error correction scheme. Another shortcoming of SPECK is that it does not support resolution scalability [47].

In the last few years, some attempts to overcome SPECK shortcomings have been suggested in the literature, for instance [47–49]. In what follows, we list only some works whose applications seem useful in case of VSN. More complex SPECK-based algorithms such as Vector SPECK [49] are not reported. A listless variant of SPECK image compression called LSK is suggested in [48]. LSK uses the block-partitioning policies of SPECK and does an explicit breadth first search, without the need for lists as in [46] or [50]. State information is kept in an array of fixed size that corresponds to the array of coefficient values, with two bits per coefficient to enable fast scanning of the bit planes. The authors in [47] suggest another variant of SPECK called Scalable SPECK (S-SPECK), which extends the original SPECK to a highly scalable low complexity scheme.

Adopting SPECK as a compression tool in power-constrained devices, such as visual sensors, might be a promising technique, due to its high compression ratio and low computational complexity. Its advantages over JPEG and EZW in terms of high compression ratio are less computational complexity and low power consumption, as well as less complex implementation which make it possible to play an interesting task in image compression for power-limited applications. Low-power image compression SPECK encoders are highly encouraged in VSN application. To the best of our knowledge, the integration of SPECK within a compression chain of a VSN has not yet been investigated. An open research work may be the implementation of SPECK-based coders dedicated to the power-constrained VSN. A listless version of SPECK, as in [48], could be an efficient scheme to be implemented in visual sensors.

4.5. Other Wavelet-Based Compression Schemes for VSN. Herein, we consider another category of compression schemes, where authors do not use or modify an existing scheme, but rather they develop their own DWT-based method which fits their circumstances. Several research works have dealt with low-memory DWT schemes. Our goal is not to survey all DWT implementations suggested in the literature, but rather we review algorithms applicable to VSN. The line-based version of the image wavelet transform proposed in [51, 52] employs a buffer system where we store only a subset of the wavelets coefficients. That is, a considerable reduction in memory is observed, compared to the traditional transform approach.

The authors in [53] introduce the fractional wavelet filter as a computation scheme to calculate fractional values of each wavelet subband. This allows the image wavelet transform to be implemented with very low RAM requirements. More precisely, the authors show that their schemes permit to a camera sensor having less than 2 kByte RAM to perform a multilevel 9/7 image wavelet transform. The picture dimension can be 256×256 using fixed-point arithmetic and 128×128 using floating-point arithmetic. Compared to [51, 52], the line-based method cannot run on a sensor with very small memory. The fractional wavelet filter method reduces the memory requirements compared to the line-based approach. The authors do not show the impact of their scheme on energy consumption.

Based on the fact that an image is generally constituted by a set of components (or regions) with unequal importance, the authors in [54] explore this idea to build a semireliable scheme for VSN called image component transmission (ICT). ICT scheme can be performed in two phases. In the first phase, the identification of the important components within the target image is performed after DWT process. After that, in the second phase, unequally important levels of transmissions are applied to different components in the compressed image. Important parts within an image, such as the information for the positions of significant wavelet coefficients, are transmitted reliably. While relatively less important components (such as the information for the values of pixels) are transmitted with lower importance, leading to energy efficiency. In fact, the suggested methodology transmission is generic and independent of specific wavelet image compression algorithms.

In [55], the authors propose an adaptive energy-aware protocol for image transmission over VSN. It is based on wavelet image decomposition using the Le Gall 5-tap/3-tap wavelet filter and semireliable transmission using priority-based mechanism. The compression is achieved through the combination of the Le Gall 5-tap/3-tap wavelet filter with Lempel-Ziv-Welch (LZW) technique [9]. The target image is firstly decomposed using wavelet filter, which provides multiple levels of resolution of the input image having different priorities. After that, the semireliable policies are applied to the wavelet coefficients by intermediate nodes. Based on their remaining energies, intermediate nodes decide whether they drop or forward packets. As explained in [55], packet priority is defined either based on the wavelet resolution level of the image or based on the wavelet coefficients magnitude. This transmission scheme offers a trade-off in consumed energy versus reconstructed image quality, and it shows the advantage of the magnitude-based prioritization method over the resolution level method. However, this mechanism sacrifices a certain amount of image quality to prolong the VSN's lifetime.

The authors in [56] consider the slow activity scenario in clustered VSN. For that reason they suggest an adaptive and distributed wavelet compression algorithm. The key features of the proposed scheme are described as follows. This algorithm exploits the spatial inherent correlations between sensor readings using a position estimation and compensation method. For that purpose, a compression method based on 5/3 wavelet filter is used (the authors also mention the possibility to use EZW or SPIHT as a compression tool). They also propose a change detection algorithm to mark active blocks within a target image, and they only encode these blocks, which permits to reduce computation complexity without sacrificing the quality of the image reconstruction.

After the survey of the main transform-based schemes, we review in Section 5 another category of compression schemes, which is non-transform-based such as vector quantization and fractals.

5. Non-Transform-Based Algorithms

5.1. Vector Quantization Compression. Vector quantization (VQ) is a conventional method for performing data compression [57, 58]. VQ can be viewed as a mapping of a large set of vectors into a small subset of code vectors called the codebook. Formally, a vector quantizer Q is a mapping from a k-dimensional Euclidean space R^k into a finite subset C of R^k, called codebook. Thus, $Q : R^k \rightarrow C$. We highlight that the most important step is the codebook construction. The well-known algorithm used to design the codebook is LBG [59].

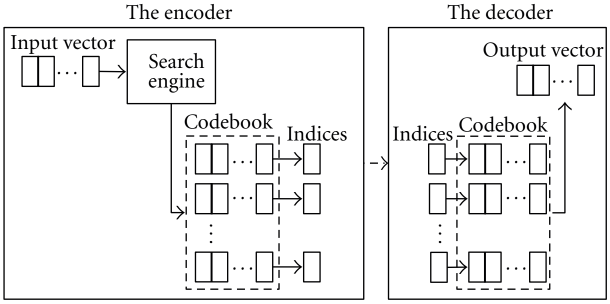


Figure 10: Vector quantization: Encoder and Decoder.

The encoder assigns to each input vector x from R^k an index which corresponds to a vector in C, that in turn is mapped to a codeword c in the set C by a decoder. If a distortion measure $d(x, c)$ which represents the cost associated with reproducing vectors x by c is defined, then the best mapping Q is the one which minimizes $d(x, c)$.

In image compression, basic vector quantization consists in dividing the input image into blocks of size $l \times m$ pixels, where each block is considered as a k-dimensional vector represented by a data vector x in the set R^k. Each vector is, then, compared with the entries of an appropriate codebook C, and the index i of the codebook entry c_i (most similar to the source data vector) is sent to a destination. At the destination, the index accesses to the corresponding entry from an identical codebook and permits to reconstruct approximately the original image (Figure 10). For more detail, the reader is referred to [9, 57, 60].

In this kind of compression (and in fractal compression presented in Section 5.2), one should note the absence of transformation block, such as DCT or DWT, and entropy encoding block, which may reduce the computation complexity. The remaining task is to compare between the gain, in terms of power dissipation, of VQ (without transformation block), and a usual framework encoding scheme incorporating transformation block (such as DCT or DWT) and entropy encoding block.

The advantage of image VQ over other types of quantizers is the simplicity of its decoder, since it only consists of table lookups. However, the basic disadvantage of VQ is its complexity, which increases with the increase of vector dimensionality. This complexity may decrease the coding speed and increase the power dissipation of the decoder especially in power-constrained applications such as VSN. Another disadvantage of image VQ is related to the design of a universal codebook for a large database of images, which requires an important memory and huge number of memory accesses.

Several image coding schemes with vector quantization have been proposed in the imaging literature. However, no VQ scheme has been proposed in VSN context. We find appealing to supply this section by some attractive works which may help for the conception and design of a new

VQ-based compression method dedicated to VSN. Particularly, we roughly present works which provide a VQ-based scheme exceeding the state of the art compression standards such as JPEG and JPEG2000 in terms of energy efficiency. The authors in [61] have considered a method for the reduction of the power consumption of vector quantization image processing, by truncating the least significant bits of the image pixels and the codeword elements during the nearest neighbor computation. In the same way, in [62], an algorithm for low-power image coding and decoding is presented. The suggested algorithm reduces the memory requirements of vector quantization; that is, the size of memory required for the codebook and the number of memory accesses by using small codebooks, which reduces the power consumption. The authors in [63] suggest a low-power pyramid vector quantization, which on average outperforms JPEG sometimes in excess of 2 dB. Another work showing that the possibility of designing an efficient image VQ encoder that exceeds the performance of JPEG is that one suggested in [64]. The authors in [64] use the pyramidal VQ, a variant of VQ, combined with some indexing techniques which require roughly the same encoding and decoding hardware complexity. This scheme outperforms JPEG implementations. The paper [65] evaluates and compares JPEG2000 with a new variant of VQ called successive approximation multistage vector quantization (SAMVQ) compression algorithms for hyperspectral imagery. It is observed in [65] that the SAMVQ outperforms JPEG2000 by 17 dB of PSNR at the same compression ratios. Unfortunately, since SAMVQ was patented by CSA, its main idea and its degree of complexity are not clearly presented. The work in [66] combines two kinds of VQ, predictive VQ (PVQ) and discrete cosine transform domain VQ (DCTVQ), to yield an efficient hybrid image compression scheme. Moreover, this scheme uses a simple classifier which employs only three DCT coefficients within each block of 8 × 8 pixels. For each image block, the classifier switches to the DCTVQ coder if the block is not complex, and to the PVQ coder if the block is relatively complex. The suggested algorithm can achieve higher PSNR values than VQ, PVQ, JPEG, and JPEG2000 at the same bit rate. This scheme may be a good candidate for power-aware applications such as VSN.

Data compression using VQ could be an acceptable compression technique for VSN, due to their reasonably compression ratio and relatively simple structure. Since VQ-based compression scheme could be implemented without any transformation (i.e., DCT or DWT), which dissipates the highest percentage of energy within a compression scheme, it is interesting to think about the design of VQ schemes dedicated to VSN. The encoder within such scheme has to be light compared to DCT-based encoder or DWT-based encoder. Low-power image VQ encoders are encouraged in VSN application. To the best of our knowledge, the application of VQ compression method in VSN has not yet been investigated.

5.2. Fractal Compression. Fractal image compression is a lossy compression technique based on fractal theory, which basically states that an image can be described by a set of fractals. Therefore, a compressed image using fractals contains a set of parameters allowing the decoder side to yield approximately a mathematical representation of the input image. Like VQ, fractal image compression is significantly different from conventional compression techniques such as JPEG, as it is not based on frequency transformations such as DCT or DWT.

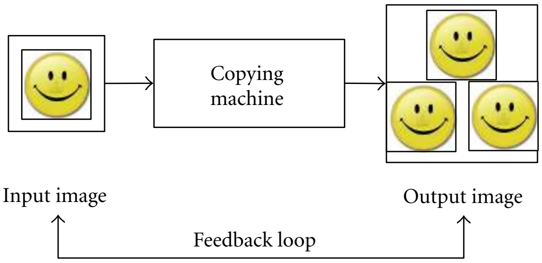


Figure 11: A photocopying machine which makes three reduced copies of the input image [67].

To the best understanding of the reader, let us introduce quickly fractal concept. Fractals are an iterative reproduction of a basic pattern, or geometric form, according to some mathematical transformations, including rotation, scaling, and translation. As explained in [67], let us imagine a copying machine which makes three reduced copies of the input image; see Figure 11. Imagine now that we fed the output of this machine back as input, the result will be an iteration of the input image. If we repeat this process many times on several input images, we will obtain Figure 12, where the process converges to the same final image Figure 12(c).

With fractal image compression, we exploit the self-similarity property between objects within natural images, which is expressed as similar repeating patterns, to reduce the image's file size. The well-known image coding scheme based on fractals is summarized in three steps as follows [68].

(1) Range block partition: partitioning the original image into nonoverlapped blocks R_i of size $s \times s$, called ranges.

(2) Domain block selection: for each R_i, search in the image to find a block D_i of size $2s \times 2s$ (double size of range block) that is very similar to R_i.

(3) Mapping: select the mapping functions, which map the domain to the range by an affine transformation $F_i(D_i) = R_i$ for each i. Usually, an affine transformation is applied when a domain block is mapped to a range block. Such affine transformation includes isometries (e.g., rotation and reflection), gray level scaling, and shift operation. In general, an affine transformation is given by: $D_i \rightarrow \alpha \cdot D_i + \Delta$, where α is the scale factor and Δ is the luminance shift factor. The best estimate can be obtained by minimizing the distance between R_i and D_i (usually the distance is represented by the Euclidean norm). The mapping relationships, which are called fractal codes, are recorded as compressed data.

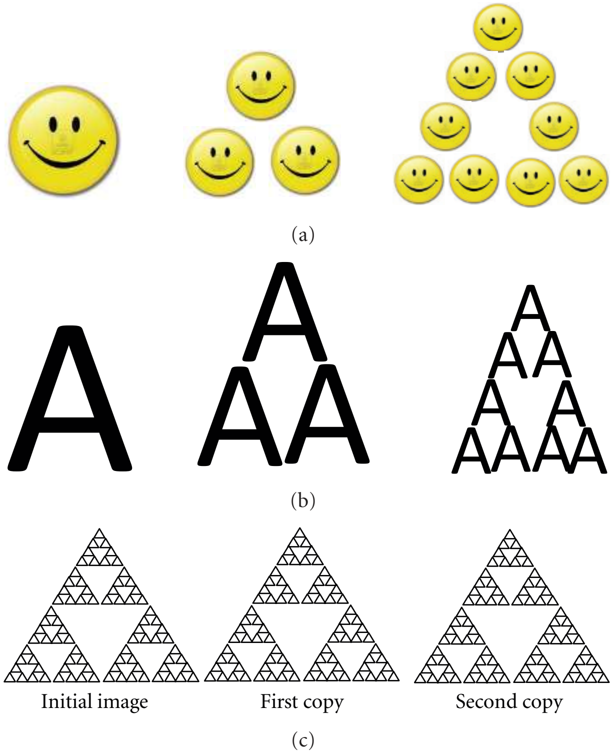


Figure 12: The first three copies generated by the photocopying machine of Figure 11 [67].

Fractal encoding is used to convert an input image to fractal codes, while fractal decoding is just the reverse, where a set of fractal codes are converted to reconstruct the input image.

The main noticeable advantages of fractal image compression can be summarized as follows: high achievable compression ratio; good quality of the reconstructed image; simple decoding process which is viewed as a simple interpretation of the fractal codes and their translation into bitmap image; fractal images are stored or sent as mathematical formulas instead of bitmaps, which minimize the sorting/sending cost; and the possibility of image scaling without distortion compared to JPEG. Nevertheless, fractal image compression presents a main drawback related to the encoding process which is extremely computationally intensive and time consuming. This is due to the hard tasks of finding all fractals during the partition step and the search for the best match of fractals.

After the first fractal-based image coder introduced by Jaquin in 1990, several variations of fractal coders have been proposed. Most of them focus on the improvement of the encoding process, especially on two main parts, partition [69, 70] and mapping [71]. Furthermore, some attempts to improve fractal compression encoding have tried to join fractal with some transforms, such as DCT and DWT. In [72], there is some early works trying to combine fractal with DCT and wavelets. In [73], the authors suggest a fast encoding algorithm for fractal image compression using the DCT inner product. One of the papers [74] is trying to join wavelet transform with fractal encoding. The main goal behind joining fractal with certain transforms is to take advantage of identifying more self-similarities within the frequency domain, in order to eliminate more redundant data and speed up the encoding process, which might reduce the computational complexity. Unfortunately, despite these improvements, the encoding process is still yet complex, and its application to VSN shortens the lifetime of the network.

To the best of our knowledge, no work has been suggested for the use of fractals within the compression chain of VSN. The main justification could be the high computational complexity of the encoding process. This complexity limits the usefulness of fractally compressed data to power-constrained applications such VSN.

An open research issue might be the adaptation and the integration of fractal compression within VSN codecs handling only natural images. This is due to the fact that fractal image compression has proven its efficiency especially on this kind of images and provides very high compression ratios [74]. Joining fractal with certain transform including DCT or DWT is another key issue permitting the reduction of the encoding process complexity. Another open research issue concerns the introduction of parallelism while using fractals in VSN. This technique allows circumventing the computational load of the encoding fractal compression within a VS node. Various parallel implementations of a fractal image compression are proposed in the literature [75–77]. A reader interested by this subject is invited to consult [78, 79].

6. ISC Summary

A brief summary is introduced in this section to show the best compression algorithms that possibly fit VSN requirements. Of the aforementioned discussed standards and algorithms, few of them could be a good candidate for VSN. The selection criterion is based mainly on the low power dissipated by a VS running one of compression algorithms in question, while having an adequate quality of the reconstructed image at the sink. The second criterion may be the low memory usage. It is difficult to say that one algorithm is less in power dissipation than another one without an evaluation on real testbed.

Let us start this discussion by the non-transform-based algorithms such as fractals and VQ. The main drawback of fractal image compression is related to the encoding process which is extremely computationally intensive and time consuming. This is due to the hard tasks of finding all fractals during the partition step and the search for the best match of fractals. The authors in [7] compare fractals with other schemes and their impact on fingerprint and face recognition. They found poorer PSNR results with fractals compared to other methods such as JPEG, JPEG2000, SPIHT, and VQ, specially with low bit rate. More details can be found in [7].

The basic disadvantage of VQ is its complexity, which increases with the increasing of vector dimension. This comlexity may decrease the coding speed and increase the power dissipation of the decoder especially in power-constrained applications such as VSN. Another disadvantage of VQ is related to the design of a universal codebook for a large database of images, which requires an important memory and huge number of memory accesses.

From the pravious discussion and some experiments [7], DCT- and DWT-based methods seem to be a relatively less energy dissipation than VQ and fractals. Depending on the compression ratio and the image quality, one shall select between DCT or DWT methods. DCT exhibits an annoying blocking artifacts in low bit rate. For DWT-based methods, SPECK has proven its efficiency in terms of both simplicity and image quality measure, followed by SPIHT and EZW [30]. However, the EBCOT algorithm requires additional memory requirement, which increases the dissipated energy and the computational complexity. Precisely, EBCOT uses two tiers: Tier-1 and Tier-2 to code information, which needs long time processing and high power consumption [39]. More precisely, it is reported in [40] that Tier-1 accounts for more than 70% of encoding time due to extensive bit-level processing, followed by the DWT transformation stage. From the viewpoint of hardware implementation, SPIHT is preferred over EBCOT coding [35].

After the examination of the main ISC compression schemes suggested in the literature, we present in the following section a small review on the distributed source coding (DSC) paradigm.

7. Distributed Source Coding Paradigm

To be self contained, we supply our paper by a short introduction on DSC paradigm and some related works. For more information on the subject, readers are advised to read [80] or [3]. DSC for VSN, refers to the compression of multiple statistically dependent sensor outputs that do not communicate with each other. Each sensor sends, independently, its compressed output to a base station for joint decoding. The well-known conventional one-to-many coding framework used in most codec's, such as MPEG, is reversed under DSC paradigm. In fact, within the one-to-many framework, the encoder usually behaves complex, compared to the relatively simple decoder complexity. On the other hand, the many-to-one coding paradigm, which is the intrinsic characteristic of DSC, moves the encoder complexity at the decoder side. Therefore, encoders can be designed simple, compared to the more complex decoders implemented at the sink. Under DSC paradigm applied in VSN, the complexity of the coder side is then shifted to the decoder at the sink, where enough power is available. Despite the inherent encoder simplicity characteristic of the DSC, their theoretical restrictions have not yet been closely achieved by practical applications. The theoretical aspects behind DSC schemes are outside the scope of this paper. We refer our reader to [81] for more details.

The lossless Slepian-Wolf and lossy Wyner-Ziv coding schemes are an encouraging conceptual basis for DSC. In practice, lossy DSC is usually implemented using a quantizer followed by lossless DSC, while the decoder consists of the joint entropy decoder followed by a joint dequantizer [80]. A brief description of Wyner-Ziv theorem is supplied, since it represents a promising solution for VSN and achieves a comparable performance to that of MPEG. The Wyner-Ziv theorem extends the Slepian-Wolf work for lossy coding with a distortion measure. Theoretically, a Wyner-Ziv encoder can be seen as a coarse quantizer of the original signal, followed by a Slepian-Wolf encoder stage, which performs lossless encoding of source data assuming that the decoder has access to some side information which is not known to the encoder [82]. To reconstruct the received signal at the decoder with minimum distortion, joint source-channel coding is performed using side information (complete sensed data sent by one of the sources). Figure 13 shows a schematic diagram of the Wyner-Ziv encoder/decoder. For more information about this subject, the reader is referred to [80].

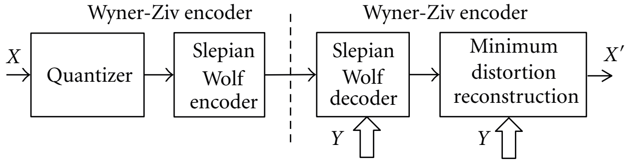


Figure 13: Practical Wyner-Ziv encoder obtained by cascading a quantizer and a Slepian-Wolf encoder [80].

Recall that our interest in this section is to review the interesting DSC schemes of still images in VSN context, including distributed JPEG2000 and distributed coding of overlapped images taken by different cameras.

We start by the work presented in [83], where the authors use JPEG as a compression method to reduce the size of images without any special considerations to energy factor during compression stage. Rather, they consider a scenario where sensors, sharing the same field of view, can process and combine overlapping regions to reduce the energy spent on image transmission. For that reason, a distributed protocol was proposed and evaluated. The simulations show that the distributed protocol, when compared to sending images individually, can achieve some reductions in energy consumption.

The authors in [84] present a method for distributed coding technique of images in VSN by exploring correlation between overlapped sensor field of views. To do that, overlapped images are first registered via a method involving the extraction of image feature points and feature points analysis. After that, the region of overlap is identified, and each sensor sends a low-resolution version of the overlapped area toward the receiver. At the reception, the base station uses the superresolution methods allowing the high-resolution version of the overlapped region.

The work in [85] is inspired by the concept of parallel distributed computing theory. A distributed lapped biorthogonal transform- (LBT-) based image compression scheme is proposed for VSN. It uses LBT transform, which is very suitable for distributed implementation in the sensor network, compared to DCT or DWT. Moreover, to further reduce the computational complexity, Golomb and multiple quantization coders are used in image compression instead of Huffman or arithmetic coding. For routing purposes, the proposed scheme is designed based on the well-known LEACH protocol, which is designed for clusters sensor networks [86]. This scheme prolongs the lifetime of the network under a specific image quality requirement. Compared to DCT, LBT improves coding efficiency by solving the problem of blocking artifacts and taking into consideration interblock spatial correlation. Weighted against DWT, LBT may lower considerably the complexity of computation and reduce the required memory.

In resource-constrained VSNs, the authors in [87] firstly notice the high energy consumption of JPEG2000. To make it light, they distribute the work load of wavelet transform to several groups of nodes along the path between the source and the destination using the concept of parallel distributed computing theory. That is, they propose two data exchange schemes with respect to image quality and energy consumption. In the first scheme, the target image is partitioned into a set of blocks along the rows to perform 1D wavelet. Similarly, the target image is divided into a set of blocks to perform 1D wavelet on columns. This data exchange scheme does not result in any image quality loss. In the next scheme, the image is partitioned into tiles, and each tile is sent to a node to do 2D wavelet transform independently. The authors in [87] show, by simulation, that the distributed scheme improves significantly the network lifetime compared to a centralized approach.

8. Other Scheme: Compressive Sensing

Compressed Sensing (CS), also called compressive sampling, is a new paradigm that combines both signal acquisition and compression. Originally, CS is based on the work of Candès et al. [88] and Donoho [89]. This section is by no means an exhaustive overview of the literature on the CS or an in depth mathematical description of the CS theory, but rather it presents basic definition related to CS and some works related to the integration of CS within a VSN. Issues, such as formulating the problem of sparse event detection in sensor networks as a CS problem [90], or the look for a suitable transformation that makes the signal sparse, are not considered. We refer our reader to [88, 89] for the theoretical concepts behind CS paradigm.

Any real-valued, finite length, and compressible signal $x \in \mathbb{R}^N$ can be represented in terms of basis matrix $\psi_{i}{}_{i=1}^{N}$, which is assumed to be orthogonal,

$$x = \sum_{i=1}^{N} s_i \psi_i \quad \text{or } x = \psi s, \tag{1}$$

where s is the $N \times 1$ column vector of weighting coefficients $s_i = \langle x, \psi_i \rangle = \psi^t x$. The signal x is called K-sparse if K coefficients of s_i coefficients of (1) are nonzero, and $(N - K)$ are zero. The case of interest is when $K \ll N$. In many applications, signals have only a few large coefficients. One of the most applications of sparse representation is in image compression, where a an image with dense (nonzero) pixel values can be encoded and compressed using a small fraction of the coefficients after a transformation, such as DCT or DWT. In fact, CS has been motivated by a striking observation: if the source signal s is K-sparse, s can be recovered from a small set of observations $y \in \mathbb{R}^M$ under a linear projection on x,

$$y = \phi x = \phi \psi s = \Theta s, \tag{2}$$

where $\Theta = \phi\psi$, and the measurements matrix $\phi \in \mathbb{R}^{M\times N}$ is typically full rank with $M < N$. There exist infinitely many solutions of s that give rise to y in (2). The CS theory states that, for most full rank matrices ϕ that are incoherent to ψ, if s is K-sparse, it is the unique solution of a regularized ℓ_0-minimization (ℓ_0-min) program [88]

$$\min \|\mathbf{s}\|_0 \quad \text{subject to } \mathbf{y} = \phi\psi\mathbf{s}. \tag{3}$$

Unfortunately, solving (3) is both numerically unstable and NP-complete, requiring an exhaustive enumeration of all $(N\ K)$ possible locations of the nonzero coefficients. Surprisingly, optimization based on the ℓ_1 norm,

$$\min \|\mathbf{s}\|_1 \quad \text{subject to } \mathbf{y} = \phi\psi\mathbf{s}, \tag{4}$$

can exactly recover K-sparse signals and closely approximate compressible signals with high probability using only $M \geq cK \log(N/K)$ iid Gaussian measurements [91].

The CS paradigm combines acquisition and compression in one step, which is totally different than conventional compression paradigms mentioned in this paper. This allows the reduction in power computation, which is very required in limited power applications such as VSN. The theory of CS seeks to recover a sparse signal from a small set of linear and nonadaptive measurements. The tremendous advantage of CS is to exhibit recovery methods that are computationally feasible, numerically stable, and robust against noise and packet loss over communication channels. Despite the aforementioned CS benefits, there still exists a huge gap between theory and imaging applications. In particular, it is unknown how to construct an efficient sensing operator and reduce the number of random measurements needed at the acquisition stage, particularly when the measurement is performed in spatial domain.

The authors in [92] study the performance of CS for VSN images in terms of complexity and quality of reconstruction. In order to assess the performance of CS, the authors implement the block diagram shown in Figure 14, where x is the input image of $N \times N$ pixels, and M is the number of measurements. The projection is performed onto a measurement matrix whose elements are generated by gathering 256 samples of the Fourier coefficients of the input image X along each of r radial lines in the frequency plane as explained in [92]. The authors show that it is possible to operate at very low data rates with reduced complexity and still achieving good image quality at the reception.

Based on CS, an image representation scheme for VSN is proposed in [93]. The target image is firstly divided into two components through a wavelet transform: dense and sparse components. The former is encoded using JPEG or JPEG2000, while the latter is encoded using CS. In order to improve the rate distortion performance, the authors suggest leveraging the strong correlation between dense and sparse components using a piecewise autoregressive model. Given the measurements and the prediction of the sparse component as initial guess, they use projection onto convex set algorithm to reconstruct the sparse component. In general, the proposed work reduces the number of random measurements needed for CS reconstruction and the

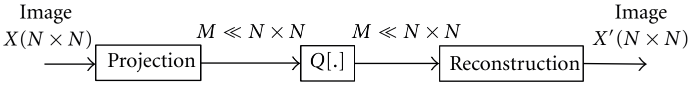


FIGURE 14: Simulated system block diagram [92].

decoding computational complexity, compared to some CS methods.

In [91], the authors suggest algorithms and hardware implementation to support CS. In fact, they use a camera architecture, called single-pixel camera (which is detailed in [94]), which employs a digital micromirror device to carry out optical calculations of linear projections of an image onto pseudorandom binary patterns. Its main characteristic is the ability to acquire an image with a single detection element. This can significantly reduce the computation and the power required for video acquisition and compression. In [95], the authors propose a sparse and fast sampling operator based on the block Hadamard transform. Despite its simplicity, the proposed measurement operator requires a near optimal number of samples for perfect reconstruction. From the practical standpoint, the block Hadamard transform is easily implemented in the optical domain (e.g., using the single-pixel camera [94]) and offers fast computation as well as small memory requirement. The suggested algorithm seems very efficient to be applied in power-constrained applications such as VSN. The unique work adopting CS paradigm in the context of VSN is that one developed in [96], where both CS and JPEG are used for compression purpose. No details about the CS scheme are furnished in [96].

9. Guidelines for Designing a Compression Scheme for VSN

In general, the design of a power efficient compression algorithm depends on all compression stages of the compression chain (Recall that the transformed-based algorithms are preferred over the non-transform-based algorithm (Section 6).) In other words, it depends on the selected transform, such as DCT, LT, or DWT; the selection of an appropriate quantization matrix; the entropy encoder, such Huffman or Golomb-Rice encoder; and the interconnection between those stages. Moreover, depending on the application domain, either lossy or lossless schemes have to be selected, knowing the fact that lossy scheme is generally preferred over lossless one, in terms of energy efficiency. We have to mention also that it is mandatory to deal with the acquisition phase before compression. In fact, at the exception of CS (Section 8), all compression methods do not consider image acquisition while its encoding. Joining acquisition phase while compressing the input image helps to reduce drastically the overall energy of a visual sensor. Another related point is to know whether or not intermediate nodes within the established path between the source and the destination are required to encode and decode images. Decoding and encoding images by intermediate nodes requires extra energy related to decoding process, compared to nodes relaying packets without further decoding stage. In such a case (encoding and decoding tasks), the decoding process has to be light in terms of energy computation.

In general, a dedicated compression algorithm for VSN has to exhibit the following properties.

(i) Acceptable compression rate.

(ii) Low-power consumption.

(iii) Low computational complexity.

(iv) Low dynamic memory usage.

(v) Embedded encoding.

10. Conclusion

In this survey paper, we provided an overview about the current state of the art in VSN compression algorithms and pointed out new classification of the currently proposed compression schemes along with their advantages, shortcomings, and open research issues. Two main coding paradigms for VSN are discussed: individual source coding (ISC), also known as one-to-many coding such as JPEG, and distributed source coding (DSC), which is related to the compression of multiple statistically dependent sensor outputs.

For ISC paradigm, we have considered two types of compression algorithms, transform-based (DCT and DWT) and non-transform-based algorithms (fractals and VQ). Throughout the literature review, we have observed that transform-based algorithms are generally preferred over non-transform-based ones. This is due the fact that the encoder is less complex, which justifies its usefulness for power-constrained applications. Moreover, for transform-based algorithms, we found that SPECK followed by EZW and SPIHT are excellent candidates for image compression for VSN. Light versions of these algorithms are requested to compress efficiently images over VSN.

Of the considered paradigms, DSC fits well the nature of the distributed VSN. Hence, distributed schemes are preferred over ISC algorithms, which may reduce in the consumed energy. Even the existence of a considerable number of distributed algorithms for VSN, most of them are theoretical (such as Wyner-Ziv), simulation based, or considered only for a small-scale VSN. That is, new DSC solutions are highly encouraged for VSN.

The compressive sensing is the last theory considered in this paper. It represents the unique paradigm that combines acquisition and compression, which allows a considerable reduction in energy consumption. That is, CS-based schemes for VSN are highly requested.

References

[1] I. F. Akyildiz, T. Melodia, and K. R. Chowdhury, “A survey on wireless multimedia sensor networks,” *Computer Networks*, vol. 51, no. 4, pp. 921–960, 2007.

[2] E. Gürses, G. B. Akar, and N. Akar, "A simple and effective mechanism for stored video streaming with TCP transport and server-side adaptive frame discard," *Computer Networks*, vol. 48, no. 4, pp. 489–501, 2005.

[3] S. Misra, M. Reisslein, and G. Xue, "A survey of multimedia streaming in wireless sensor networks," *IEEE Communications Surveys and Tutorials*, vol. 10, no. 1–4, pp. 18–39, 2008.

[4] S. Soro and W. Heinzelman, "A survey of visual sensor networks," *Advances in Multimedia*, vol. 2009, Article ID 640386, 21 pages, 2009.

[5] Y. Charfi, N. Wakamiya, and M. Murata, "Challenging issues in visual sensor networks," *IEEE Wireless Communications*, vol. 16, no. 2, pp. 44–49, 2009.

[6] B. Krishnamachari, D. Estrin, and S. B. Wicker, "The impact of data aggregation in wireless sensor networks," in *Proceedings of the 22nd International Conference on Distributed Computing Systems (ICDCSW '02)*, pp. 575–578, IEEE Computer Society, Washington, DC, USA, 2002.

[7] A. Mascher-Kampfer, H. Stögner, and A. Uhl, "Comparison of compression algorithms' impact on fingerprint and face recognition accuracy," in *Visual Communications and Image Processing*, Proceedings of SPIE, February 2007.

[8] D. U. Lee, H. Kim, S. Tu, M. Rahimi, D. Estrin, and J. D. Villasenor, "Energy-optimized image communication on resource-constrained sensor platforms," in *Proceedings of the 6th International Symposium on Information Processing in Sensor Networks (IPSN '07)*, pp. 216–225, April 2007.

[9] K. Sayood, *Introduction to Data Compression*, Morgan Kaufmann, San Francisco, Calif, USA, 3rd edition, 2005.

[10] G. K. Wallace, "The JPEG still picture compression standard," *Communications of the ACM*, vol. 34, no. 4, pp. 30–44, 1991.

[11] C. N. Taylor, D. Panigrahi, and S. Dey, "Design of an adaptive architecture for energy efficient wireless image communication," *Lecture Notes in Computer Science*, pp. 260–273, 2002.

[12] G. A. Ruiz, J. A. Michell, and A. Burón, "High throughput parallel-pipeline 2-D DCT/IDCT processor chip," *Journal of VLSI Signal Processing Systems for Signal, Image, and Video Technology*, vol. 45, no. 3, pp. 161–175, 2006.

[13] G. A. Ruiz, J. A. Michell, and A. M. Burón, "Parallel-pipeline 8×8 forward 2-D ICT processor chip for image coding," *IEEE Transactions on Signal Processing*, vol. 53, no. 2 I, pp. 714–723, 2005.

[14] C. F. Chiasserini and E. Magli, "Energy consumption and image quality in wireless video-surveillance networks," in *Proceedings of the 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC '02)*, pp. 2357–2361, September 2002.

[15] J. Liang and T. D. Tran, "Fast multiplierless approximations of the DCT with the lifting scheme," *IEEE Transactions on Signal Processing*, vol. 49, no. 12, pp. 3032–3044, 2001.

[16] A. Mammeri, A. Khoumsi, D. Ziou, and B. Hadjou, "Energy-aware jpeg for visual sensor networks," in *Proceedings of the MCSEAI Conference*, pp. 639–647, Oran, Algeria, 2008.

[17] G. Pekhteryev, Z. Sahinoglu, P. Orlik, and G. Bhatti, "Image Transmission over IEEE 802.15.4 and ZigBee networks," in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS '05)*, pp. 3539–3542, May 2005.

[18] E. Magli, M. Mancin, and L. Merello, "Low-complexity video compression for wireless sensor networks," in *Proceedings of the International Conference on Multimedia and Expo (ICME '03)*, pp. 585–588, IEEE Computer Society, Washington, DC, USA.

[19] C. Loeffer, A. Ligtenberg, and G. S. Moschytz, "Practical fast 1-D DCT algorithms with 11 multiplications," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, pp. 988–991, May 1989.

[20] C. F. Chiasserini and E. Magli, "Energy-efficient coding and error control for wireless video-surveillance networks," *Telecommunication Systems*, vol. 26, no. 2–4, pp. 369–387, 2004.

[21] K. Y. Chow, K. S. Lui, and E. Y. Lam, "Balancing image quality and energy consumption in visual sensor networks," in *Proceedings of the 1st International Symposium on Wireless Pervasive Computing*, pp. 1–5, January 2006.

[22] K.-Y. Chow, K.-S. Lui, and E. Y. Lam, "Efficient selective image transmission in visual sensor networks," in *Proceedings of the VTC*, pp. 1–5, 2007.

[23] K. S. Lui and E. Y. Lam, "Image transmission in sensor networks," in *Proceedings of the IEEE Workshop on Signal Processing Systems (SiPS '05)*, pp. 726–730, November 2005.

[24] C. B. Margi and K. Obraczka, "Energy consumption tradeoffs in visual sensor networks," in *Proceedings of the 24th Brazilian Symposium on Computer Networks*, Curitiba, Brazil, 2006.

[25] L. Ferrigno, S. Marano, V. Paciello, and A. Pietrosanto, "Balancing computational and transmission power consumption in wireless image sensor networks," in *Proceedings of the IEEE International Conference on Virtual Environments, Human-Computer Interfaces, and Measurement Systems (VECIMS '05)*, pp. 61–66, June 2005.

[26] S. G. Mallat, "Theory for multiresolution signal decomposition: the wavelet representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 674–693, 1989.

[27] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3445–3462, 1993.

[28] J. C. Liu, W. L. Hwang, and W. J. Hwang, "An ARQ-based diversity system for transmission of EZW compressed images over noisy channels," in *Proceedings of the International Conference on Image Processing (ICIP '02)*, pp. 221–224, September 2002.

[29] M. Hamdi, N. Boudriga, and M. S. Obaidat, "Bandwidth-effective design of a satellite-based hybrid wireless sensor network for mobile target detection and tracking," *IEEE Systems Journal*, vol. 2, no. 1, pp. 74–82, 2008.

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

[31] S. Iren and P. D. Amer, "Application level framing applied to image compression," *Annales des Telecommunications*, vol. 57, no. 5-6, pp. 502–519, 2002.

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

[33] M. Akter, M. B. I. Reaz, F. Mohd-Yasin, and F. Choong, "A modified-set partitioning in hierarchical trees algorithm for real-time image compression," *Journal of Communications Technology and Electronics*, vol. 53, no. 6, pp. 642–650, 2008.

[34] M. Wu and C. W. Chen, "Multiple bitstream Image Transmission over Wireless Sensor Networks," in *Proceedings of the 2nd IEEE International Conference on Sensors*, pp. 727–731, October 2003.

[35] L. W. Chew, W. C. Chia, L. M. Ang, and K. P. Seng, "Very low-memory wavelet compression architecture using strip-based processing for implementation in wireless sensor networks," *Eurasip Journal on Embedded Systems*, vol. 2009, Article ID 479281, 2009.

[36] W. C. Chia, L. M. Ang, and K. P. Seng, "Multiview image compression for wireless multimedia sensor network using image stitching and SPIHT coding with EZW tree structure," in *Proceedings of the International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC '09)*, pp. 298–301, August 2009.

[37] D. Taubman, "High performance scalable image compression with EBCOT," *IEEE Transactions on Image Processing*, vol. 9, no. 7, pp. 1158–1170, 2000.

[38] C. C Chang, S.-G. Chen, and J.-C. Chiang, "Efficient encoder design for JPEG2000 EBCOT context formation," in *Proceedings of the15th European Signal Processing Conference (EUSIPCO '07)*, Poznan, Poland, September 2007.

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

[40] T. W. Hsieh and Y. L. Lin, "A hardware accelerator IP for EBCOT Tier-1 coding in JPEG2000 standard," in *Proceedings of the 2nd Workshop on Embedded Systems for Real-Time Multimedia*, pp. 87–90, September 2004.

[41] C. J. Lian, K. F. Chen, H. H. Chen, and L. G. Chen, "Analysis and architecture design of block-coding engine for EBCOT in JPEG 2000," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 3, pp. 219–230, 2003.

[42] J. S. Chiang, C. H. Chang, Y. S. Lin, C. Y. Hsieh, and C. H. Hsia, "High-speed EBCOT with dual context-modeling coding architecture for JPEG2000," in *Proceedings of the IEEE International Symposium on Cirquits and Systems*, pp. III865–III868, May 2004.

[43] Q. Lu, X. Ye, and L. Du, "An architecture for energy efficient image transmission in WSNs," in *Proceedings of the International Conference on Networks Security, Wireless Communications and Trusted Computing (NSWCTC '09)*, pp. 296–299, April 2009.

[44] W. Yu, Z. Sahinoglu, and A. Vetro, "Energy efficient JPEG 2000 image transmission over wireless sensor networks," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '04)*, pp. 2738–2743, December 2004.

[45] H. Wu and A. A. Abouzeid, "Power aware image transmission in energy constrained wireless networks," in *Proceedings of the 9th International Symposium on Computers and Communications (ISCC '04)*, pp. 202–207, July 2004.

[46] W. A. Pearlman, A. Islam, N. Nagaraj, and A. Said, "Efficient, low-complexity image coding with a set-partitioning embedded block coder," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 11, pp. 1219–1235, 2004.

[47] G. Xie and H. Shen, "Highly scalable, low-complexity image coding using zeroblocks of wavelet coefficients," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 6, pp. 762–770, 2005.

[48] M. V. Latte, N. H. Ayachit, and D. K. Deshpande, "Reduced memory listless speck image compression," *Digital Signal Processing*, vol. 16, no. 6, pp. 817–824, 2006.

[49] C. C. Chao and R. M. Gray, "Image compression with a vector speck algorithm," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '06)*, pp. II445–II448, May 2006.

[50] A. Islam and W. A. Pearlman, "Embedded and efficient low-complexity hierarchical image coder," in *Proceedings of the Visual Communications and Image Processing*, pp. 294–305, January 1999.

[51] C. Chrysafis and A. Ortega, "Line-based, reduced memory, wavelet image compression," *IEEE Transactions on Image Processing*, vol. 9, no. 3, pp. 378–389, 2000.

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

[53] S. Rein and M. Reisslein, "Performance evaluation of the fractional wavelet filter: a low-memory image wavelet transform for multimedia sensor networks," *Ad Hoc Networks*, vol. 9, no. 4, pp. 482–496, 2011.

[54] W. Wang, D. Peng, H. Wang, and H. Sharif, "A novel image component transmission approach to improve image quality and energy efficiency in wireless sensor networks," *Journal of Computer Science*, vol. 3, no. 5, pp. 353–360, 2007.

[55] V. Lecuire, C. Duran-Faundez, and N. Krommenacker, "Energy-efficient transmission of wavelet-based images in wireless sensor networks," *International Journal of Sensor Networks*, vol. 4, no. 1-2, pp. 37–47, 2007.

[56] H. Dong, J. Lu, and Y. Sun, "A distributed wavelet-based image coding for wireless sensor networks," in *Intelligent Control and Automation*, vol. 344, pp. 72–82, Springer, Berlin, ,Germany, 2006.

[57] A. Gresho and R. M. Gray, *Vector Quantization and Signal Compression*, Kluwer Academic Publishers, Norwell, Mass, USA, 1995.

[58] B. Sastry and S. Kompella, *An optimized vector quantization for color image compression [M.S. thesis]*, Texas Tech university, 1998.

[59] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Transactions on Communications Systems*, vol. 28, no. 1, pp. 84–95, 1980.

[60] N. M. Nasrabadi and R. A. King, "Image coding using vector quantization: a review," *IEEE Transactions on Communications*, vol. 36, no. 8, pp. 957–971, 1988.

[61] K. Masselos, P. Merakos, and C. E. Goutis, "Power efficient vector quantization design using pixel truncation," in *Proceedings of the Proceedings of the 12th International Workshop on Integrated Circuit Design. Power and Timing Modeling, Optimization and Simulation (PATMOS '02)*, pp. 409–418, Springer, London, UK, 2002.

[62] K. Masselos, P. Merakos, T. Stouraitis, and C. E. Goutis, "Trade-off analysis of a low-power image coding algorithm," *Journal of VLSI Signal Processing Systems for Signal, Image, and Video Technology*, vol. 18, no. 1, pp. 65–80, 1998.

[63] W. Namgoong and T. H. Meng, "A low-power encoder for pyramid vector quantization of subband coefficients," *Journal of VLSI Signal Processing Systems for Signal, Image, and Video Technology*, vol. 16, no. 1, pp. 9–23, 1997.

[64] A. C. Hung, E. K. Tsern, and T. H. Meng, "Error-resilient pyramid vector quantization for image compression," *IEEE Transactions on Image Processing*, vol. 7, no. 10, pp. 1373–1386, 1998.

[65] S. E. Qian, M. Bergeron, C. Serele, I. Cunningham, and A. Hollinger, "Evaluation and comparison of JPEG 2000 and vector quantization based onboard data compression algorithm for hyperspectral imagery," in *Proceedings of the Learning From Earth's Shapes and Colours*, pp. 1820–1822, July 2003.

[66] Z. M. Lu and H. Pei, "Hybrid image compression scheme based on PVQ and DCTVQ," *IEICE Transactions on Information and Systems*, vol. 88, no. 10, pp. 2422–2426, 2005.

[67] Y. Fisher, M. Latapy, and D. Paris, "Compression fractale d'images," http://focus.ti.com/lit/an/bpra065/bpra065.pdf.

[68] C. M. Xu and Z. Y. Zhang, "A fast fractal image compression coding method," *Journal of Shanghai University*, vol. 5, no. 1, pp. 57–59, 2001.

[69] M. Kawamata, M. Nagahisa, and T. Higuchi, “Multi-resolution tree search for iterated transformation theory-based coding,” in *Proceedings of the ICIP*, pp. 137–141, 1994.

[70] D. Saupe and H. Hartenstein, “Lossless acceleration of fractal image compression by fast convolution,” in *Proceedings of the IEEE International Conference on Image Processing (ICIP ’96)*, pp. 185–188, September 1996.

[71] N. Zhang and H. Yan, “Hybrid image compression method based on fractal geometry,” *Electronics Letters*, vol. 27, no. 5, pp. 406–408, 1991.

[72] Y. Fisher, D. Rogovin, and T. P. Shen, “A comparison of fractal methods with dct and wavelets,” in *Neural and Stochastic Methods in Image and Signal Processing III*, Procedings of SPIE, pp. 2304–2316, 1994.

[73] T. K. Truong, J. H. Jeng, I. S. Reed, P. C. Lee, and A. Q. Li, “A fast encoding algorithm for fractal image compression using the DCT inner product,” *IEEE Transactions on Image Processing*, vol. 9, no. 4, pp. 529–535, 2000.

[74] A. V. D. Walle, “Merging fractal image compression and wavelet transform methods,” in *Proceedings of the NATO Advanced Study Institute*, pp. 8–17, Springer, 1995.

[75] C. Hufnagl, J. Hmmerle, A. Pommer, A. Uhl, and M. Vajtersic, “Fractal image compression on massively parallel arrays,” in *Proceedings of the International Picture Coding Symposium*, Berlin, Germany, 1997.

[76] K. P. Acken, H. N. Kim, M. J. Irwin, and R. M. Owens, “Architectural design for parallel fractal compression,” in *Proceedings of the International Conference on Application-Specific Systems, Architectures and Processors*, pp. 3–11, August 1996.

[77] W. A. Stapleton, W. Mahmoud, and D. J. Jackson, “Parallel implementation of a fractal image compression algorithm,” in *Proceedings of the 28th Southeastern Symposium on System Theory (SSST ’96)*, pp. 332–336, April 1996.

[78] D. Saupe and R. Hamzaoui, “A review of the fractal image compression literature,” *SIGGRAPH Computer Graphics*, vol. 28, no. 4, pp. 268–276, 1994.

[79] D. Liu and P. K. Jimack, “A survey of parallel algorithms for fractal image compression,” *Journal of Algorithms and Computational Technology*, vol. 1, no. 2, pp. 171–186, 2007.

[80] Z. Xiong, A. D. Liveris, and S. Cheng, “Distributed source coding for sensor networks,” *IEEE Signal Processing Magazine*, vol. 21, no. 5, pp. 80–94, 2004.

[81] P. L. Dragotti and M. Gastpar, *Dsitributed Source Coding: Theory, Algorithms and Applications*, Elsevier, 2009.

[82] A. D. Wyner and J. Ziv, “The rate-distortion function for source coding with side information at the decoder,” *IEEE Transactions on Information Theory*, vol. IT-22, no. 1, pp. 1–10, 1976.

[83] K. Y. Chow, K. S. Lui, and E. Y. Lam, “Efficient on-demand image transmission in visual sensor networks,” *Eurasip Journal on Advances in Signal Processing*, vol. 2007, Article ID 95076, 2007.

[84] R. Wagner, R. Nowak, and R. Baraniuk, “Distributed image compression for sensor networks using correspondence analysis and super-resolution,” in *Proceedings of the International Conference on Image Processing (ICIP ’03)*, pp. 597–600, September 2003.

[85] Q. Lu, W. Luo, J. Wang, and B. Chen, “Low-complexity and energy efficient image compression scheme for wireless sensor networks,” *Computer Networks*, vol. 52, no. 13, pp. 2594–2603, 2008.

[86] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, “Energy-efficient communication protocol for wireless microsensor networks,” in *Proceedings of the 33rd Annual Hawaii International Conference on System Siences (HICSS)*, p. 223, January 2000.

[87] H. Wu and A. A. Abouzeid, “Energy efficient distributed image compression in resource-constrained multihop wireless networks,” *Computer Communications*, vol. 28, no. 14, pp. 1658–1668, 2005.

[88] E. J. Candès, J. Romberg, and T. Tao, “Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information,” *IEEE Transactions on Information Theory*, vol. 52, no. 2, pp. 489–509, 2006.

[89] D. L. Donoho, “Compressed sensing,” *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[90] J. Meng, H. Li, and Z. Han, “Sparse event detection in wireless sensor networks using compressive sensing,” in *Proceedings of the 43rd Annual Conference on Information Sciences and Systems (CISS ’09)*, pp. 181–185, March 2009.

[91] M. B. Wakin, J. N. Laska, M. F. Duarte et al., “Compressive imaging for video representation and coding,” in *Proceedings of the 25th PCS: Picture Coding Symposium (PCS ’06)*, April 2006.

[92] W. Barakat and R. Saliba, “Compressive sensing for multimedia communications in wireless sensor networks,” Tech. Rep. MDDSP Project Final Report, 2008.

[93] B. Han, F. Wu, and D. Wu Volume, “Image representation by compressive sensing for visual sensor networks,” *Journal of Visual Communication and Image Representation*, vol. 21, no. 4, pp. 325–333, 2010.

[94] M. F. Duarte, M. A. Davenport, D. Takbar et al., “Single-pixel imaging via compressive sampling: building simpler, smaller, and less-expensive digital cameras,” *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 83–91, 2008.

[95] L. Gan, T. T. Do, and T. D. Tran, “Fast compressive imaging using scrambled block Hadamard ensemble,” in *Proceedings of the European Signal Processing Conference (EUSIPCO ’08)*, Lausanne, Switzerland, August 2008.

[96] P. Chen, P. Ahammad, C. Boyer et al., “CITRIC: a low-bandwidth wireless camera network platform,” in *Proceedings of the 2nd ACM/IEEE International Conference on Distributed Smart Cameras*, pp. 1–10, 2008.

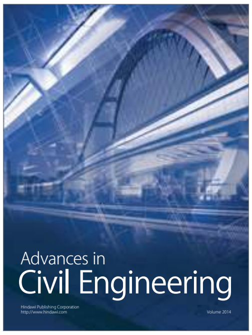
Journal of
Engineering

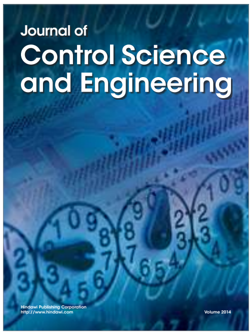
The Scientific
World Journal

International Journal of
Rotating
Machinery

Journal of
Sensors

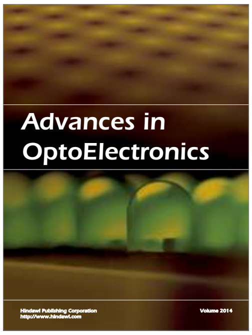
International Journal of
Distributed
Sensor Networks


Advances in
Civil Engineering


Journal of
Control Science
and Engineering

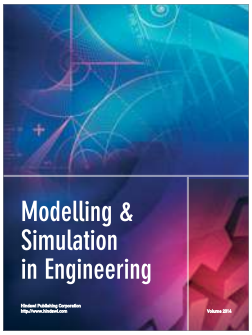
Journal of
Robotics

Journal of
Electrical and Computer
Engineering

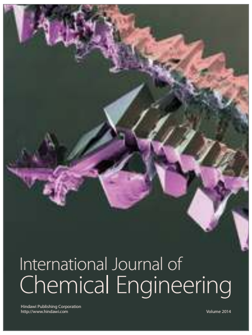

Advances in
OptoElectronics

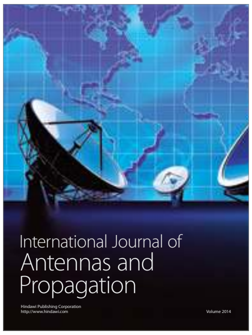
VLSI Design

International Journal of
Navigation and
Observation

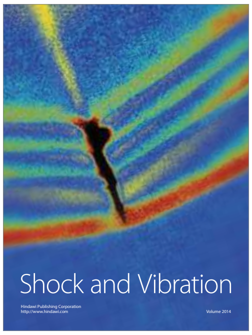

Modelling &
Simulation
in Engineering

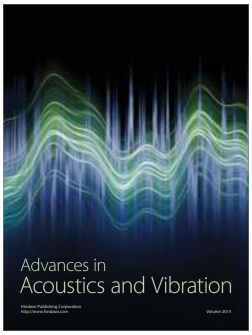
International Journal of
Aerospace
Engineering


International Journal of
Chemical Engineering


International Journal of
Antennas and
Propagation

Active and Passive
Electronic Components


Shock and Vibration


Advances in
Acoustics and Vibration