A Practical Study of Jointly Exploiting Multiple Image Compression Techniques for Wireless Multimedia Sensor Networks

Moad Y. Mowafi, Fahed H. Awad, Eyad S. Taqieddin, Omar Q. Banimelhem
Multimedia Networking Research Laboratory (MNRL)
Jordan University of Science and Technology/Department of Network Engineering and Security, Irbid, Jordan
Email: {mowafi, fhawad, eyadtaq, omelhem}@just.edu.jo

***Abstract*—In Wireless Multimedia Sensor Networks, the amount of multimedia traffic generated by the source nodes and relayed throughout the network towards the sink nodes is relatively vary large. As a result, the overall network performance and operational lifetime rely heavily on the efficiency and the complexity of the compression techniques used. However, known compression techniques have either high efficiency or low complexity but not both, which causes an imbalance in the distribution of the available network resources and hence shortens the network lifetime. In this work, a testbed to experimentally evaluate the performance and energy consumption of image compression and transmission was developed. Based on the testbed, the JPEG and JPEG2000 image compression algorithms were simultaneously embedded and evaluated with the purpose of jointly utilizing the features of both algorithms in order to facilitate a balanced tradeoff between the application performance requirements and the current network status. The analysis of the results verified that, in practical implementations of Wireless Multimedia Sensor Networks, using more than one compression technique by the same source node is feasible and has the potential of utilizing the capabilities of each technique based on the given situation.**

***Index Terms*—Imote2, JPEG, JPEG2000, TinyOS, experimental evaluation, WMSN**

I. INTRODUCTION

The rapid advances in microelectronics and wireless communications have enabled the manufacturing and deployment of tiny wireless sensor nodes in complex and remote environments. These nodes are usually equipped with self-discovery and self-organizing capabilities that allow them to autonomously form a Wireless Sensor Network (WSN). In some applications, these sensor nodes are integrated with cameras and microphones that generate multimedia data. As a result, researchers have shed the light on Wireless Multimedia Sensor Networks (WMSNs) that suit a wide range of vision-based audio-based applications, such as object tracking, environmental monitoring, localization, and video surveillance [1].

Manuscript received February 7, 2012; accepted March 27, 2012.
This work was funded by the Jordan University of Science and Technology, Research Project No. 69/2011.

The processing and transmission of multimedia data require relatively large amounts of energy and bandwidth. Therefore, the bandwidth-limited and energy-constrained nature of WMSNs mandates efficient processing and transmission of multimedia data.

In WMSN applications, there is a performance tradeoff between processing and transmitting of multimedia data. For example, in an image sensor node, the captured raw image may be either transmitted as is or it may be compressed at the source node before transmission. Because of the resource limitations of the sensor nodes, the second choice is usually preferred.

Image compression techniques use different coding algorithms in order to reduce the image size. The comparison between these techniques usually uses three performance metrics, which are the compression complexity, represented by the compression time and energy consumption, the compression efficiency, represented by the compression ratio (CR) and image quality, and the transmission cost, in terms of time and energy.

Several research projects examined the image compression and transmission for WMSN. Most of these projects focused on the JPEG [2] and JPEG2000 [3] compression algorithms and their variants. The common goal of most of these projects was to lower the compression complexity of the given algorithm in order to reduce the overall energy consumption. This was done by either optimizing the algorithm itself on the source node or by distributing the compression tasks on other nodes. However, the performance evaluation used in most these projects was simulation-based, while only a few of them were experimental.

The goal of this study is to investigate the possibility of utilizing the features of different compression algorithms, embedded simultaneously into the source node, in order to adapt the compression complexity and the compression efficiency to the application requirements and the network conditions. In order to achieve this goal, a testbed was developed in order to run the experiments and obtain the performance metrics. As part of the testbed, the JPEG and JPEG2000 algorithms were embedded into the TinyOS-based Imote2 platform [4,5,6].

doi:10.4304/jcm.7.4.309-320

The remainder of this paper is organized as follows. A discussion of the related work is presented in Section II. In Section III, the developed testbed is discussed in details. The experimental evaluation and the obtained results are presented in Section IV. The analysis of the results is discussed in Section V. The paper is concluded is Section VI.

II. Related Work

In this section, a review of the previous work related to this research is presented. In [7], a customized version of JPEG that targeted the compression complexity in resource-constrained platforms was introduced. The performance of the proposed algorithm was experimentally evaluated, where stored standard images were compressed and transmitted using the Mica2, MicaZ, and Telos WSN platforms. The results demonstrated the compression/transmission energy tradeoff for different processor/radio combinations.

A JPEG-like compression algorithm was proposed and evaluated by simulation using the instruction set of the Telos WSN platform [8]. The results showed a tradeoff between energy consumption and image distortion.

In [9], a performance analysis of JPEG2000 over the IEEE802.15.4 WSN standard was presented. The relationship among the compression parameters, image quality, and network throughput was studied through simulation. The results showed that many parameters and features of JPEG2000 must be tuned to adapt the compression to the IEEE802.15.4 constraints.

An adaptive algorithm based on JPEG2000 was proposed in [10]. The algorithm selects the optimal image compression parameters to minimize the total energy consumption in a multi-hop WSN, given the network conditions and image quality constraints. The performance of the proposed algorithm was evaluated by simulation that is based on measured data. These data represent the energy consumption of the used JPEG2000 coder running on a StrongARM processor. The results showed that the energy consumed in compression has a fairly significant portion of the total consumed energy.

The work in [11] proposed a distributed image compression algorithm based on JPEG2000 as a means to overcome the computation and/or energy limitation of individual nodes by sharing the processing of tasks. The proposed distributed algorithm was compared with the centralized algorithm by simulation with respect to two performance metrics: the total energy consumption and the system lifetime. As in [10], the energy consumption for the JPEG2000 coder was measured on a StrongARM processor, which was then used in the simulation. The results showed that the proposed scheme prolongs the system lifetime compared to the centralized image compression.

A distributed image compression algorithm based on JPEG was proposed in [12] in order to overcome the computation and energy limitation of individual nodes. This was accomplished by assigning image compression tasks to nodes other than the source node. The proposed distributed algorithm was compared with the centralized algorithm by simulation in terms of network lifetime. The results showed that the proposed algorithm extends the network lifetime compared to the centralized algorithm.

In [13], the use of Zigbee networks for image transmission was experimentally tested. Compressed images were pre-stored in the nodes, thus no actual image compression was performed. A comparison between JPEG and JPEG2000 was performed in terms of two performance metrics: the number of bytes received in error per image and the Peak Signal-to-Noise Ratio (PSNR). The results showed that JPEG2000 images encoded into multiple quality layers are more error-resilient while high PSNR is maintained and that JPEG2000 is more suitable in low rate sensor networks.

The work in [14] proposed a methodology to reduce the energy consumption of a visual sensor node based on the balance between compression and transmission tasks. In order to select the best compression algorithm, the compression time of five algorithms including JPEG and JPEG2000 was estimated by simulation on the PC. To validate the simulation results, one of the algorithms was implemented on a sensor node with a low-power microcontroller in order to estimate the compression and communication times. The scaling factor between the PC-based and the microcontroller-based execution time was used to estimate the compression time of the other compression algorithms.

In our previous work in [15], an experimental evaluation of JPEG on the TinyOS-based Imote2 platform was presented. The results showed a tradeoff between the compression and transmission costs.

The research presented in this paper surpasses the prior art mentioned above in several aspects. The evaluation was experimental; unlike the pure simulation of hardware platforms as in [8] and [9] or the use of measured data from a single node in simulation as in [10]. The testbed that was designed and used in the experiments consists of hardware (i.e.; the Imote2 platform) that is specifically designed for WMSN with enough resources to capture, store and compress images using more than one simultaneously embedded compression algorithms; unlike the resource-limited hardware platforms used by others (e.g.; MICAz and Telos) that use pre-stored images as in [7]. The goal of this work was to optimize the whole process of compressing and transferring images via a network by utilizing the benefits of more than one compression algorithm rather than focusing on a single algorithm, trying to optimize the compression parameters as in [10], distributing and delegating the compression tasks to neighbor nodes as in [11] and [12], or evaluating the performance of the compression algorithm itself as in [15]. Finally, while the work in [13] and [14] considered more than one algorithm, the performance of the compression algorithm itself was either not evaluated [13] or estimated by simulation on the PC [14].

III. Testbed Setup

As mentioned before, a testbed was developed in order to provide an experimental environment to evaluate the performance of the JPEG and JPEG2000 compression algorithms on the Imote2 platform and obtain the various performance metrics. The testbed presented in [15] was redesigned with new features and components following the guidelines proposed in [16]. It consists of four main hardware units:

1) The Camera Node (CN), which is the multimedia source node responsible of capturing, compressing, and transmitting the image data.
2) The Base Station (BS), which runs and manages the different tools for user interface, data storage and analysis, and the communication with the CN. The BS consists of the following components:
   a. The User Interface Module (UIM).
   b. The Communication Module (CM).
   c. The Storage Unit (SU).
   d. The Data Analysis Unit (DAU).
3) The Gateway (GW), which connects the CN to the PC and relays the information back and forth between them.
4) The Current Measurement Module (CMM), which is used to measure the energy consumption of the CN during the different phases of operation.

Fig. 1 shows the block diagram of the various testbed components and their corresponding connections. The following sections provide details about each component.

A. The Camera Node

The CN is a WSN source node with an integrated image sensor. It is responsible of capturing, compressing, and sending image data to the GW via the WSN. In this work, the CN is an Imote2 WSN platform connected to an Imote2 Multimedia Board (IMB400) [17], which supports image capturing via an integrated camera module.

In our testbed, the CN was setup to be used in one of two modes: a pre-stored image mode and image-capture mode. In the pre-stored image mode, a raw image is saved into the memory of the CN. The image is then compressed and sent by the CN. On the other hand, in the image-capture mode, the CN captures a new image before it is being compressed and sent. The choice between the two modes and the features of the image capturing and/or compression are determined through a command message sent to the CN node by the UIM via the GW.

For this work, existing implementations of the JPEG and JPEG2000 standards are embedded into the CN, simultaneously. The JPEG implementation uses the Independent JPEG Group's (IJG) library [18], which allows the user to specify the quality of the JPEG image using a quality-setting scale, called the quality factor (Q), which ranges between 1 and 100. A Q of 1 produces the smallest file size, but with the worst image quality, whereas a Q of 100 produces the largest file size with the best image quality. On the other hand, the JPEG2000 is implemented using the OpenJPEG library [19]. The user has the option to either specify the desired quality of a JPEG2000 image using a PSNR value (in dB) or to specify a target compression ratio using a factor of compression.

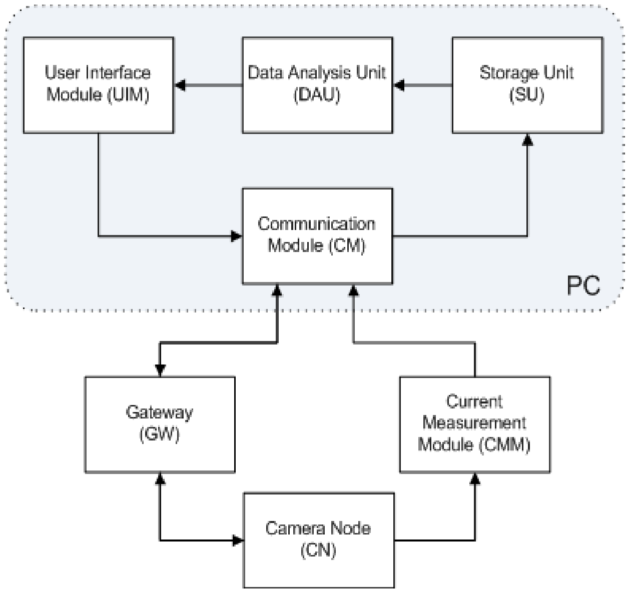


Figure 1. The block diagram of the testbed.

B. The Gateway

The GW is another Imote2 WSN platform connected to the BS through the CM. It is in charge of forwarding the command messages from the BS to the CN via the WSN. When an image is sent back from the CN to the BS, the GW relays it to the SU through the CM.

The communication between the GW and the CN is implemented using the Collection Tree Protocol (CTP) [20], which supports multi-hop routing via the WSN. However, in this work, only single-hop communication was experimentally tested.

C. The User Interface Module

The UIM is a JAVA-based GUI used by the user to choose a stored- or captured-image mode and specify the features of the image capturing and compression to be performed by the CN. Through the UIM, image requests are initiated with the desired image resolution (i.e.; VGA, QVGA, CIF, or QCIF), image format (i.e.; raw, JPEG, or JPEG2000), and/or the targeted image quality or compression ratio. Each request is sent through the GW to the CN in the form of a command message that includes the desired configuration parameters. Upon the receipt of the command message, the CN retrieves the raw image stored in its memory or captures a new image with the required resolution, then processes it based on the required format, quality, and/or compression ratio. After being sent back by the CN to the BS, the image information and analysis are passed to UIM via the DAU in order to be displayed to the user and/or to be used for constructing the next command message to the CN.

D. The Communication Module

The CM is responsible of connecting the different testbed components together. It passes the command messages from the UIM to the CN via the GW and both the image data from the CN via the GW and the energy consumption data from the CMM to the SU. Different ports on the BS are used for each stream of data and the CM listens to each of them separately.

E. The Current Measurement Module

The CMM is directly connected to the load (i.e.; the CN and its associated hardware) to accurately measure the operational current draw. The readings of the CMM are stored in the SU along with timestamps to indicate the time, at which each measurement was taken. These timestamps help in determining the various phases, through which the CN is going (i.e.; compression, radio on/off, transmission, etc.) and to identify the current requirements of each phase.

One approach for current measurement is to place a resistor in series with the load. Using Ohm's law, the value of the voltage on the resistor will yield the value of the current. This approach, however, results in power dissipation through the resistor, which will affect the overall performance of the system. Moreover, a small current will result in a small voltage across the resistor and hence, inaccurate results are obtained. To overcome these problems, the CMM, shown in Fig. 2, was developed to accurately measure the current draw (I_{Load}).

The idea here is that I_{Load} is translated to a voltage, V_{sense}, when it passes through the 1Ω resistor, R_{sense}. This voltage is too small to be accurately measured as it falls within the error range of an Analog-to-Digital Converter (ADC). Therefore, the AD620 instrumentation amplifier [21] is used to amplify V_{sense} in order to be accurately measured. The gain of the amplifier, G, is determined using the resistor R_G as follows:

$$G = \frac{49.4\ k\Omega}{R_G} + 1 \quad (1)$$

In this work, a gain of 25.7 was achieved using $R_G = 2.0\ k\Omega$. The resulting output voltage (V_{out}) is:

$$V_{out} = G \times V_{sense} \quad (2)$$

The load in Fig. 2 consists of an Imote2 platform, including: the microcontroller, the radio board, and the IMB400 board. The power modes of these components (i.e.; ON, OFF, active, sleep, etc.) determine the instantaneous current draw, which is usually in the millimeter range.

The output of the 12-bit ADC on the MDA300CA sensor board [22] is passed by the MicaZ mote [23], which in turn sends the readings to the SU through the CM. Note that the MicaZ may be connected to the BS wirelessly or through a cable.

F. The Storage Unit

The SU is used to store the received images, to keep records of the events taking place in the system, and to log the CMM readings. All these data are passed through the CM.

The CMM readings are stored along with timestamps that indicate when each reading was taken. Additionally, a log file with information about the image request time, capture time, compression time, and transmission start/end time, is also stored. By using this information, it is possible to map these times to the logged information from the CMM and thus, the corresponding current draw can be accurately determined.

G. The Data Analysis Unit

The DAU is used in the computation of the CR and the PSNR image parameters during the experiments. These values are based on the stored raw and JPEG-compressed images. The values are then fed to the UIM as input parameters of the requested JPEG2000 images that are sent to the CN via a command message, as will be shown in Section IV.

Moreover, the DAU contains scripts for extracting data from the stored log files. In the energy computation, it is essential to know the duration of time, through which a node is in any of the possible states (e.g.; transmitting vs. processing) as well as to determine the current draw. These data are readily available and need only to be parsed.

By knowing the period through which a certain value of current is drawn, the energy may be calculated using

$$E = (V_{bat} - V_{sense}) \times I_{Load} \times \Delta t \quad (3)$$

where Δt is the time spent in performing an operation.

IV. EXPERIMENTAL EVALUATION AND RESULTS

In this section, the performance of the JPEG and JPEG2000 image compression and transmission is experimentally evaluated using the testbed presented in Section III.

The performance metrics used are: the compression complexity, in terms of compression time and energy, the compression efficiency, in terms of compression rate and distortion, and the transmission cost, in terms of transmission time and energy.

The following sections describe the conducted experiments and the corresponding results.

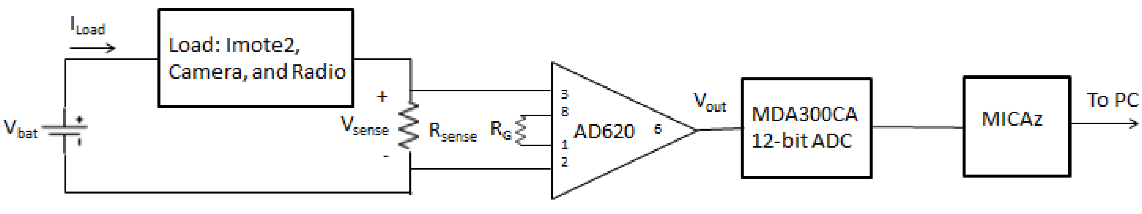


Figure 2. Current Measurement Module

A. Experiment 1: Pre-Stored Standard Image Compression

In this experiment, a standard raw image was pre-stored in the CN for the purpose of examining the testbed and verifying the relative performance of the embedded JPEG and JPEG2000 algorithms in terms of rate-distortion and compression time; as compared to the well-known results reported in the literature. Hence, the IMB400 is turned off during this experiment.

This experiment was performed based on the following procedure:

1) The standard gray scale Baboon image (see Fig. 3) with 256×256 pixels and 8 bits per pixel (bpp) was pre-stored inside the flash memory of the CN. The UIM sends a request message to the CN to compress the stored image using the JPEG algorithm for specific values of the quality factor (Q). In this experiment, Q values from 2 to 96 with increments of 2 were used. For each case, the CN measures the compression time and sends it to the BS along with the compressed image for storage and further analysis and statistics.
2) The DAU determines the CR and the PSNR for each obtained JPEG-compressed image in order to be used as inputs to the JPEG2000 algorithm.
3) The obtained CR and PSNR values are used as follows:
   a. For each CR value, the UIM sends a request message to the CN, including the CR value, to be used along with the raw image, as inputs to the JPEG2000 algorithm in order to obtain a JPEG2000-compressed image with a size (i.e.; a CR) similar to that of the JPEG image. The purpose of this step was to compare the relative rate-distortion behavior of the two algorithms and verify the correctness of embedding the compression libraries into the CN as compared to the results reported in the literature.
   b. Similarly, for each obtained PSNR value, the UIM sends a request message to the CN, including the PSNR value, to be used along with the raw image, as inputs to the JPEG2000 algorithm in order to obtain another JPEG2000-compressed image with a quality (i.e.; a PSNR) similar to that of the JPEG image. The purpose of this step was to measure and compare the relative compressed image size and the compression time of the two algorithms for the same image quality.

   For each case, the CN measures the compression time and sends it to the BS along with the compressed image for storage and further analysis and statistics.
4) The received performance data are passed to the DAU to analyze and draw the performance curves.

Fig. 4 shows the obtained rate-distortion curves for JPEG and JPEG2000. The JPEG2000 achieves an average of 2.25dB more than JPEG (about 8-10% on average). Similar behavior was reported in [24,25,26].

Fig. 5 shows the relative compression efficiency in terms of compressed image file size for JPEG and JPEG2000 with similar PSNR values. The results show that the file size of the JPEG2000 image can be less than that of a JPEG image, with similar quality, by 25% to 75%.

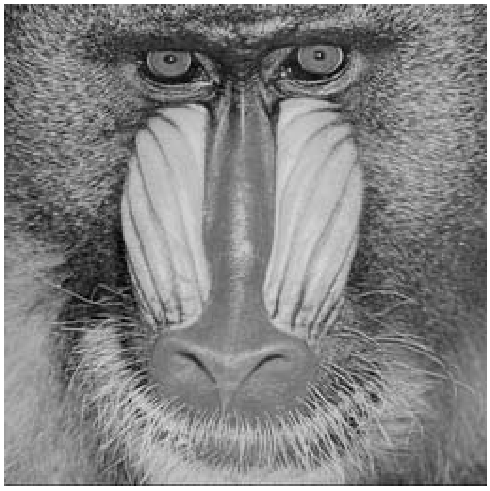


Figure 3. The standard baboon raw image used in Experiment 1

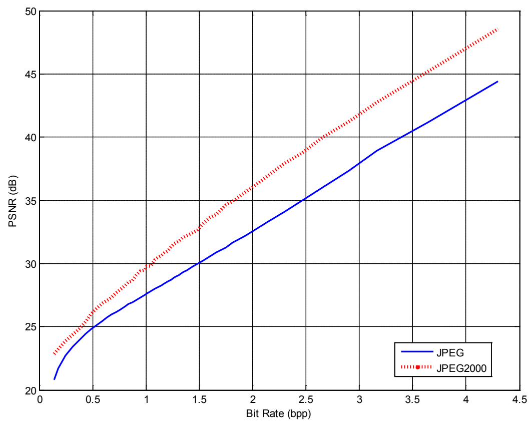


Figure 4. Rate-distortion comparison for the stored image

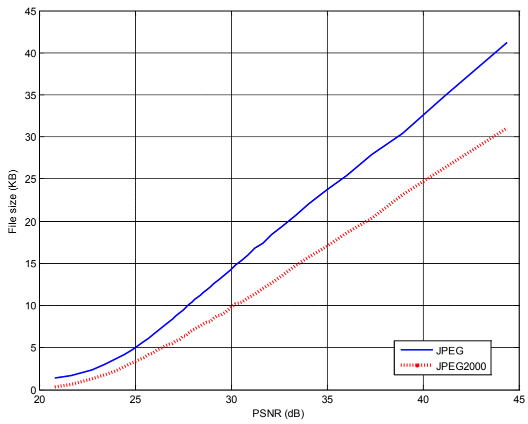
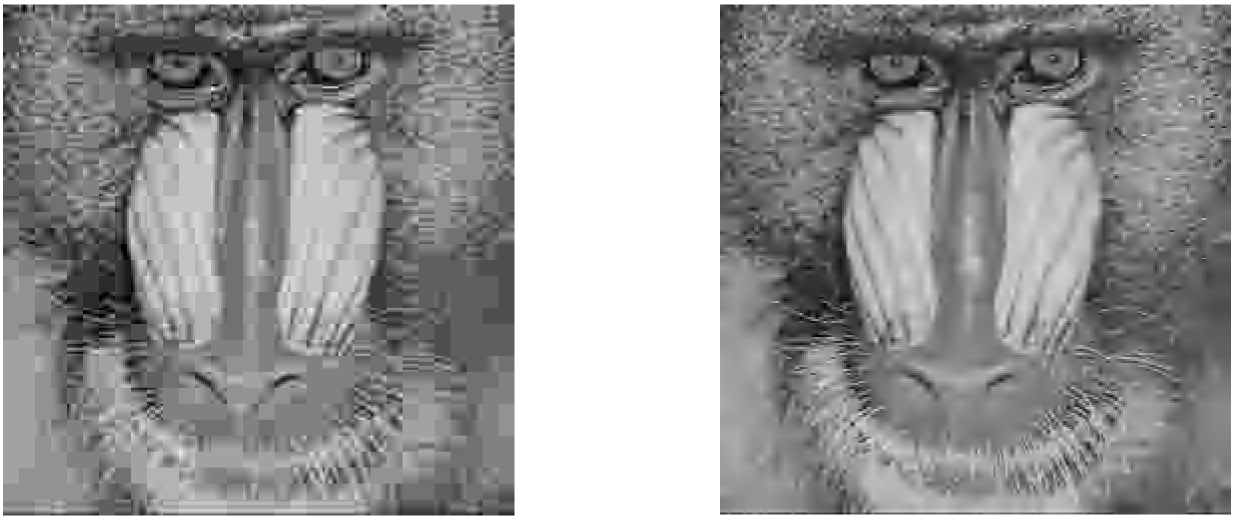


Figure 5. Comparison of the compressed image file size for the stored image



(a) (b)

Figure 6. Sample received images with similar compression ratio from Experiment 1 for (a) JPEG with Q=6, PSNR=22.7 dB, File Size =2239 B and (b) JPEG2000 with CR=28, PSNR=23.9 dB, File Size =2291 B

(a) (b)

Figure 7. Sample received images with similar quality from Experiment 1 for (a) JPEG with Q=40, PSNR=27.6 dB, File Size=9727 B and (b) JPEG2000 with PSNR=27.3dB, File Size=6327 B

The compression times for JPEG and JPEG2000, as measured during this experiment, demonstrated that the compression time of JPEG2000 is almost constant at around 5.4 seconds regardless of the compression ratio, whereas for JPEG, it increases very slowly (ranges between 0.33 to 0.43 seconds) with respect to bit rate. Hence, JPEG has shown to be around an order of ten times faster than JPEG2000. Note that this result is implementation-dependent. For example, in [26], the relative compression time of three different implementations of JPEG2000 ranged from three to ten times more than that of JPEG, depending on the level of optimization. It is worth noting here that the standard JPEG2000 implementation used in this work was used as is without any optimization.

Fig. 6 and Fig. 7 show sample compressed images and their corresponding performance measures representing the cases discussed in Experiment 1.

Fig. 6 shows the received JPEG image (Fig. 6a) and the corresponding JPEG2000 image (Fig. 6b) with similar file size (i.e.; similar CR). At CR=28 (i.e.; Q=6), the PSNR values as well as the visual perception indicate that the JPEG2000 provides better image quality than JPEG.

Fig. 7 shows the received JPEG image (Fig. 7a) and the corresponding JPEG2000 image (Fig. 7b) with similar quality (i.e.; similar PSNR). For PSNR=27.3 (i.e.; Q≈40), the file size of the JPEG2000 image is less by 35%.

B. Experiment 2: Captured Image Compression and Transmission

In this experiment, the CN was used to capture a new image then measure and compare the performance of image compression and transmission using the embedded JPEG and JPEG2000 algorithms in terms of compression complexity and energy efficiency. The purpose of this experiment was to examine the applicability of jointly utilizing both compression algorithms for practical use in WMSN. Therefore, the focus of this experiment was on compressed images with low bit rate (less than 1 bpp).

This experiment was very similar in concept to that of Experiment 1. However, they differed in the following:

1) The CN is attached to the CMM in order to log the current draw and the timestamps during the experiment.
2) The UIM sends a request to the CN to capture an image using the IMB400 multimedia sensor board before performing the compression.
3) One single image is captured at the beginning of the experiment and is used throughout the stages of the experiment in order to have a fair comparison of the algorithms. This raw image is sent to the BS as the reference image for quality analysis and comparison. Fig. 8 shows the captured raw image used in this experiment, of which the size was 320×240 with 8 bpp.

Fig. 9 shows the rate-distortion curves of the captured image for low bit rates. This figure confirms the superiority of JPEG2000 over JPEG at low bit rates. For example, JPEG2000 achieves an average of 4.25dB more than JPEG for bit rates up to 0.5 bpp. In other words, it achieves a much smaller image size than JPEG given the same image quality as shown in Fig. 10. For example, for PSNR of 30dB, the JPEG2000-compressed image is about 56% smaller than the JPEG-compressed image. This reduction in image size has a significant impact on the transmission time and energy.

Figure 8. The captured raw image used for Experiment 2

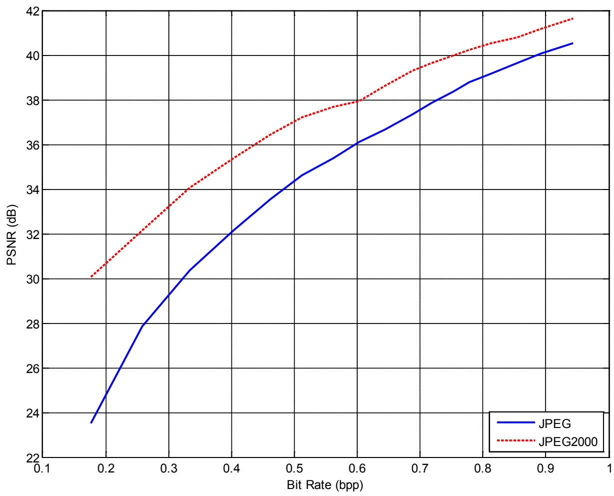


Figure 9. Rate-distortion comparison for the captured image

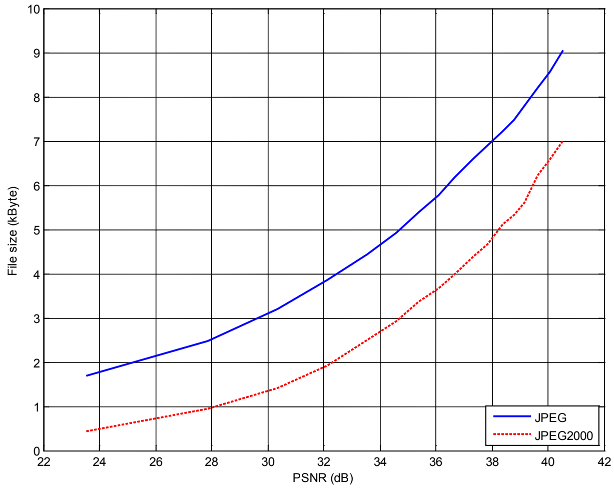


Figure 10. Comparison of the compressed image file size for the captured image

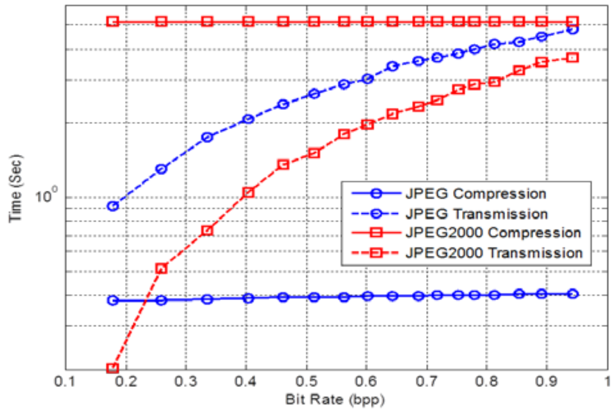


Figure 11. JPEG and JPEG2000 compression and transmission time

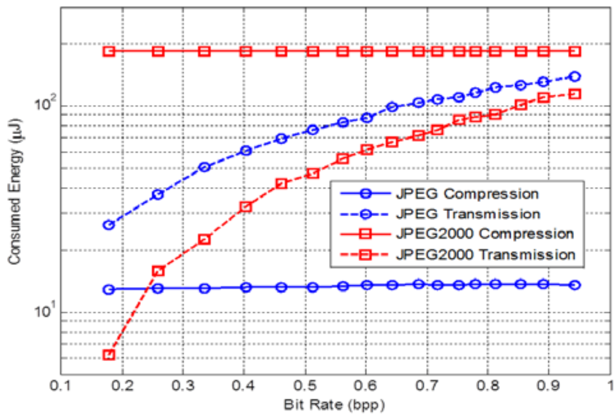


Figure 12. JPEG and JPEG2000 energy consumption

Fig. 11 shows the compression and transmission times for JPEG2000 and JPEG. The compression time of the two algorithms is almost independent of the bit rate. However, the transmission time is highly dependent on the bit rate. As a result, the compression and transmission energy consumption follow the same behavior, as seen in Fig. 12. Note here that the transmission time of JPEG2000 at low bit rates is much smaller than JPEG due to the corresponding difference in the yielded file sizes.

V. Joint Utilization of JPEG and JPEG2000

In this section, the experimental results obtained in Section IV are discussed and analyzed. Based on that, the potential of jointly utilizing JPEG and JPEG2000 in WMSN is presented.

In order to maximize the network lifetime, a tradeoff between the performance requirements of the WMSN application and the instantaneous overall distribution of the available network resources such as the nodes' remaining energy; has to be continuously maintained.

We show here that the JPEG2000 still has the potential to be practically utilized in WMSN even with the straightforward JPEG2000 implementation (i.e.; with no optimization) used in this work. Without loss of generality, having both algorithms implemented in the same source node may facilitate the possibility of adaptively utilizing the features of either algorithm based on a number of factors. These factors include, but are not limited to, the network status (such as the level of congestion and the length of the routing path), the application performance requirements (such as the end-to-end delay and the image quality), and the status and distribution of the remaining energy of the nodes.

Even with its relatively high compression complexity, JPEG2000 has an important feature that makes it very attractive, which is its relatively high compression efficiency, especially at low bit rates, that are highly suitable in most WMSN applications [13].

On the other hand, even though the compression complexity of JPEG is relatively low (about 93% less compression time and energy than JPEG2000, as shown in Fig. 11 and Fig. 12, respectively), its corresponding compression efficiency is low too, especially at low bit rates. That is, the obtained image quality and file size are worse than those of JPEG2000. This can be clearly observed in Fig. 9 and Fig. 10, where JPEG2000 can improve the image quality by 2.5% to 27.5% and reduce the image file size by 25% to 75%, respectively. As a result, this reduction in file size is directly reflected on the transmission cost as observed in Fig. 11 and Fig. 12, where the transmission delay of the, even higher-quality, JPEG2000 image is reduced by 17% to 77% and the corresponding transmission energy is reduced by 12% to 77%, respectively.

Thus, it is obvious from the analysis of the results that the compression complexity gain offered by JPEG is, to some extent, comparable to the compression efficiency gains offered by JPEG2000, especially at very low bit rates, which are the most of interest to WMSN applications [13]. This analysis has only been discussed, so far, in the context of a static single-hop star network topology, where the source node captures and compresses the image and transmits it directly to the BS. However, this topology is not practical in real life WMSN applications due to the relatively high cost of transmission power incurred and the congestion at the star hub caused by the heavy multimedia traffic. The same applies to 2-hop clustered topologies, where the cluster-heads would be quickly exhausted. For that reason, multi-hop with disjoint multi-path routing is mostly preferred for WMSN's [27], especially for those applications that rely on remote sensing. In this topology, the energy-delay tradeoff is an important factor in extending the network lifetime while retaining reasonable performance. This can be achieved by distributing the traffic relaying task on as many parallel disjoint paths as possible, while maintaining balanced remaining energy among the nodes. However, the availability of such parallel disjoint paths relies heavily on the distribution of the nodes and hence the number of neighbor nodes the BS and each source node has [28]. Therefore, the utilization of the existing paths depends on the path forming strategy used by the routing algorithm.

To clarify the relative impact of multi-hop routing on the performance of transferring a compressed image from a source node to the BS, we investigate two extreme multi-hop routing scenarios: pipelined and non-pipelined routing paths.

Scenario 1. Pipelined routing path with the following characteristics:

1. A dedicated node-disjoint path.
2. No packet collision or error, hence no retransmissions, occur.
3. No other nearby active nodes that may contend for channel access. This allows for a perfect pipeline.

Note, however, that for two nodes on the path to transmit in parallel, they must be at least two hops apart. Otherwise, a collision would occur at the middle node. For that reason, no pipelining would be at all possible with less than four hops. To illustrate, consider a 5-hop routing path with four relay nodes connecting the source node and the BS; as shown in Fig. 13.

Assuming that the compressed image file is divided into p equal-size packets and that the transmission delay of each packet is t_p, the routing delay, T_r, of the image over a 4-hop path can be defined as:

$$T_r = (3p + 1)t_p \tag{4}$$

In general, for an h-hop routing path, the routing delay can be defined as:

$$T_r = [3p + (h - 3)]t_p \tag{5}$$

where $h \geq 3$. Hence, the total delay needed to compress and deliver the image to the BS via an h-hop routing path is defined as:

$$T = T_{Comp} + [3p + (h - 3)]t_p \tag{6}$$

where T_{Comp} is the image compression time.

Fig. 14 shows the total delay needed to compress and transfer the image used in Experiments 2 based on Scenario 1, as a function of the number of hops; using the measured values of p, t_p, and T_{comp}. It is worth noting here that the total time values for the single-hop case in Fig. 14 were taken from the experimentally measured values in Experiment 2.

Thus, in Scenario 1, the number of hops has a negligible impact of the relative performance of the two algorithms and hence the compression complexity is the most dominant factor.

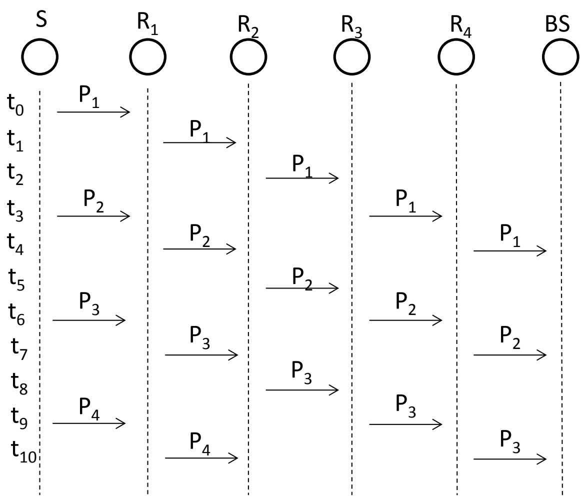


Figure 13. Pipelined multi-hop routing path

Scenario 2. Non-pipelined routing path with the following characteristics:

1. A dedicated node-disjoint path.
2. No packet collision or error, hence no retransmissions, occur.
3. Many other nearby active nodes along the path that contend for channel access and disable the pipelining. This is equivalent to a store-and-forward of the complete image.

In this case, the total delay is defined as:

$$T = T_{comp} + h \times p \times t_p \tag{7}$$

Fig. 15 shows the total delay needed to compress and transfer the image used in Experiment 2 based on Scenario 2. The most importance observation in Fig. 15(a) is that the total delay of JPE2000 surpasses that of JPEG after a certain path length; depending on the bitrate used. That is, after a number of hops (call it a cross-over path length), the compression complexity gain of JPEG was exceeded by the compression efficiency gain of JPEG2000. Note that the cross-over path length increases as the bitrate decreases since the transmission cost the compressed image is inversely proportional to the compression ratio.

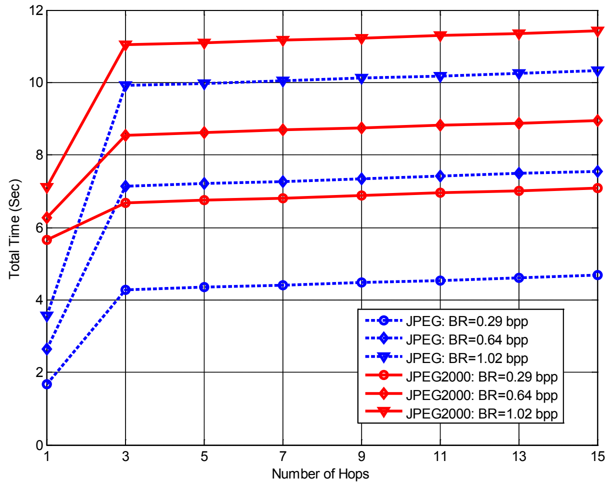


Figure 14. Total delay for Scenario 1

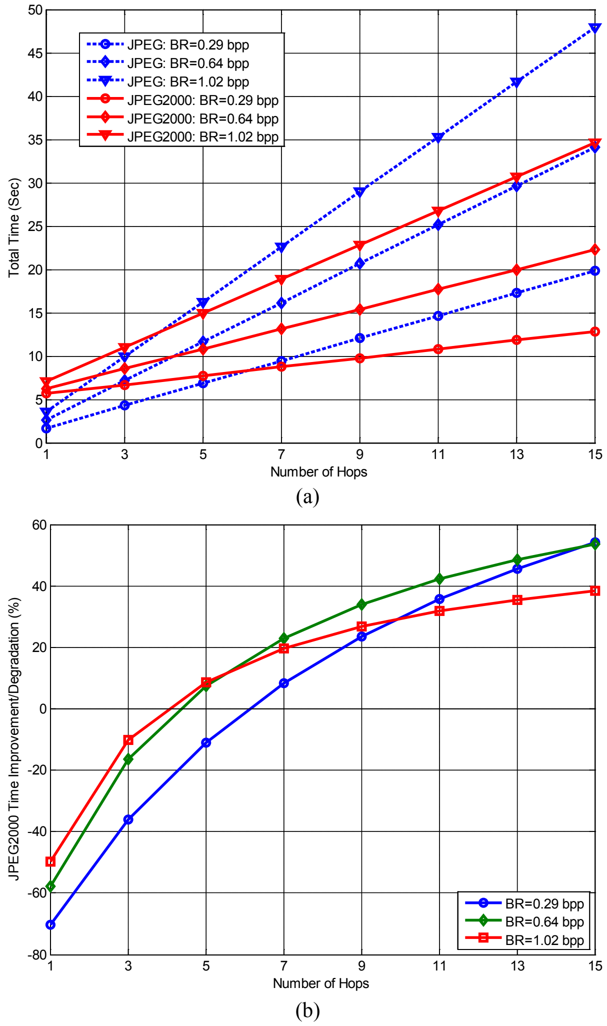


Figure 15. Total delay for Scenario 2

Fig. 15(b) shows the percent improvement/degradation of JPEG2000 on the total delay, compared to JPEG, as a function of the path length. Even though JPEG2000 is 70% worse than JPEG with a single-hop, it becomes better than JPEG beyond 4 to 6 hops and to achieve up to 55% improvement at a 15-hop path. Note here that, even though the cross-over path length of the lower bitrates is higher, the rate of its performance improvement is much faster.

On the other hand, in practical multi-hop routing and due to the broadcast and error-prone nature of the wireless link, there are several factors that influence the routing performance such as the channel access contention, packet collision, acknowledgements, and retransmissions. Even if a dedicated disjoint path is used, the surrounding in-range active nodes would still slow down the packet flow, especially with intense multimedia traffic and would increases the chances of packet collision and retransmission. Therefore, in-transit packets would get accumulated and queued in the relay nodes' buffers waiting for an opportunity to be transmitted along the path. This in turn restricts the pipelining functionality of the routing path. Thus, the practical multi-hop routing delay would be in-between the two scenarios used; rather closer to Scenario 2.

As discussed above, even though the relative delay of compressing and transferring an image is highly dependent on several network parameters such as the congestion level and the node distribution, the length of the routing path may or may not have a significant impact on it. On the other hand, the corresponding relative overall energy cost is more critical and decisive than the delay. The reason is that, regardless of how fast the packets are delivered through the network, the number of hops traversed has a direct and tenacious impact on the overall transmission energy consumed. That is, even for non-congested routing path of Scenario 1, the JPEG2000 high energy cost of compression will be shortly surpassed by the JPEG high energy cost of transmission; as seen in Fig. 16(a). In this case, the cross-over path length ranges between 3 and 5, which is shorter than that caused by the total delay even for congested routing path of Scenario 2.

Finally, Fig. 16(b) shows that the corresponding percent improvement on consumed energy that JPEG2000 achieves can be up to 70% with 15-hop path; compared to JPEG.

Based on the discussion above, in practical WMSN's with multi-hop routing, the use of a single compression algorithm at the source node may not allow for optimal compromise among the different performance metrics. For example, if JPEG is only used, then the advantage of the low-cost compression gained by the source node will shortly vanish as the number of hops along the routing path exceeds a few; since the transmission cost of the relatively large image file will be multiplied by the number of hops. Consequently, the relay nodes would be exhausted much faster than the source node. On the other hand, if JPEG2000 is only used, then the high-cost compression paid by the source node will shortly be compensated for by the saving on transmitting the relatively small file. In this case, the source nodes would die much faster than the relay nodes.

In conclusion, if both algorithms are made available at the same source node, then the two can be used adaptively and cooperatively by the BS and the source node based on different network parameters such as the hop count, the nodes' remaining energy, and/or the application performance requirements such as the end-to-end delay and the image quality.

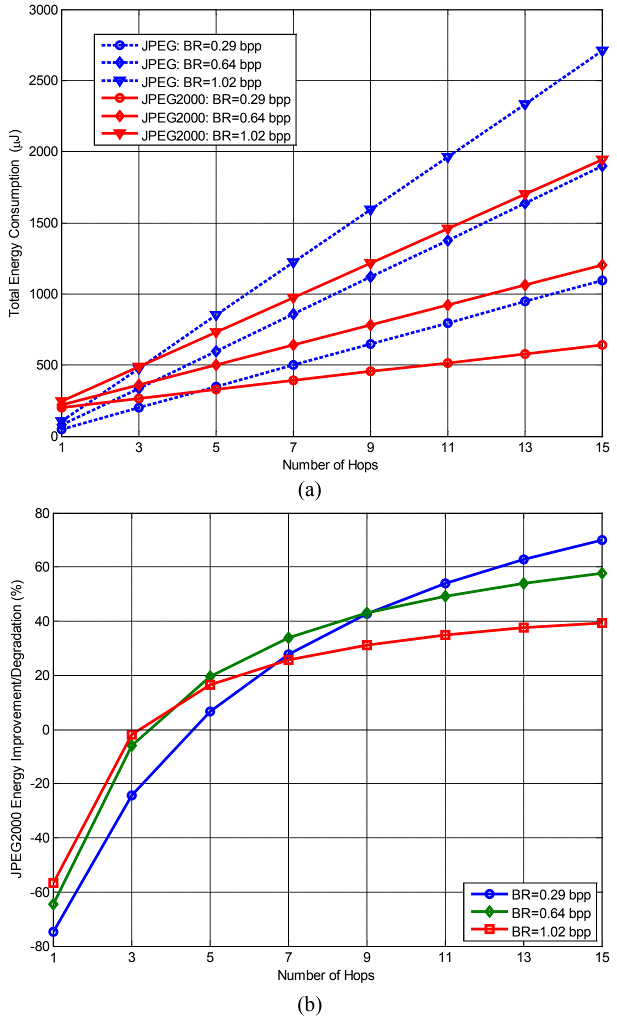


Figure 16. Total energy consumption for both Scenario 1 and Scenario 2

VI. Conclusion

In this work, the applicability of jointly utilizing the advantages of different image compression techniques in wireless multimedia sensor networks is investigated. The purpose of this study is to adapt the cost of image compression and transmission to the required image quality and network status. To that end, a general-purpose testbed was developed in order to easily implement and test different image compression techniques and compare their relative complexities and performances. Through this testbed, standard performance metrics, such as the compression complexity and efficiency and the transmission cost, can be measured, stored, and analyzed.

As part of the testbed, the JPEG and JPEG2000 algorithms were embedded simultaneously into the TinyOS-based Imote2 platform, where their performances and energy consumption were concurrently measured and compared. The results confirmed that jointly utilizing more than one compression algorithm is feasible.

As a future work, the practical scenarios, including the different routing and channel access parameters such as path setup and maintenance processes, packet collisions, traffic intensity, and node distribution, will be modeled, simulated, and then verified using the testbed.

References

[1] I. F. Akyildiz, T. Melodia, and K. R. Chowdhury, "A Survey on Wireless Multimedia Sensor Networks," *Computer Netwroks*, vol. 51:4, 2007, pp. 921–960.

[2] G.K. Wallace, "The JPEG Still Picture Compression Standard," *Commun. of the ACM*, 34:4, 1991, pp. 30-44.

[3] C. Christopoulos, A. Skodras, and T. Ebrahimi, "The JPEG2000 Still Image Coding System: An Overview," *IEEE Trans. on Consum. Electron*. 46:4, 2000, 1103-1127.

[4] L. Nachman, J. Huang, J. Shahabdeen, R. Adler, and R. Kling, "IMOTE2: Serious Computation at The Edge," in *Proceedings of IEEE International Wireless*

Communications and Mobile Computing Conference (IWCMC), Crete Island, Greece, 2008, pp. 1118–1123.

[5] Imote2 Datasheet. [Accessed: 20 October 2011] <http://www.memsic.com/support/documentation/wireless-sensor-networks/category/7-datasheets.html?download=134%3Aimote2>.

[6] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister, "System Architecture Directions for Networked Sensors," *ACM SIGPLAN Notices*, 35:11, 2000, 93–104.

[7] D. Lee, H, Kim, M. Rahimi, D. Estrin and J.D. Villasenor, "Energy-Efficient Image Compression for Resource-Constrained Platforms," *IEEE Trans. on Image Process.* 18:9, 2009, pp. 2100–2113.

[8] L. Makkaoui, V. Lecuire, J-M. Moureaux, "Fast Zonal DCT-Based Image Compression for Wireless Camera Sensor Networks," in *IEEE International Conference on Image Processing Theory, Tools and Applications (IPTA '10)*, Paris, France, 2010, pp. 126-129.

[9] P. Andrea, C. Scavongelli, S. Orcioni, and M. Conti, "Performance Analysis of JPEG 2000 over 802.15.4 Wireless Image Sensor Network," in *Proceedings of the Workshop on Intelligent Solutions in Embedded Systems*, Crete, Greece, 2010, pp. 55-60.

[10] H. Wu and A. A. Abouzeid, "Power Aware Image Transmission in Energy Constrained Wireless Networks," in *Proceedings of the International Symposium on Computers and Communications (ISCC '04)*, Alexandria, Egypt, 2004, pp. 202–207.

[11] H. Wu and A. A. Abouzeid, "Energy Efficient Distributed Image Compression in Resource-Constrained Multihop Wireless Networks", *Comput. Commun*, 28:14, 2005, pp. 1658–1668.

[12] Y. Xiaobo, S. Lijuan, and W. Ruchuan, "Distributed Image Compression Algorithm in Wireless Multimedia Sensor Networks," *ZTE Communications*, 8:1, 2010, pp. 50-54.

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

[14] 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)*, Giardini Naxos, Italy, 2005, pp. 61–66.

[15] M. Mowafi, F. Awad, E. Taqieddin, O. Banimelhem, "Experimental Evaluation of Image Compression and Transmission for TinyOS-Based Imote2 Platform," in *Proceedings of the 7th International Conference on Innovations in Information Technology (Innovations '11)*, Abu Dhabi, United Arab Emirates, 2011, pp. 173–178.

[16] I. F. Akyildiz, T. Melodia, and K. R. Chowdhury, "Wireless Multimedia Sensor Networks: Applications and Testbeds," in *Proceedings of the IEEE*, 96:10, 2008, pp. 1588-1605.

[17] IMB400 Datasheet. [Accessed: 20 October 2011] <http://www.memsic.com/support/documentation/wireless-sensor-networks/category/7-datasheets.html?download=139%3Aimote2-multimedia>.

[18] The Independent JPEG Group's JPEG Software. [Accessed: 20 October 2011] <http://www.ijg.org>.

[19] The OpenJPEG Library. [Accessed: 20 October 2011] <http://www.openjpeg.org >.

[20] R. Fonseca, O. Gnawali, K. Jamieson, S. Kim, P. Levis, and A. Woo. "TEP 123: The Collection Tree Protocol," Aug. 2006.

[21] The AD620 Datasheet. [Accessed: 20 October 2011] <http://www.analog.com/static/imported-files/data_sheets/AD620.pdf>.

[22] The MDA300 Datasheet. [Accessed: 20 October 2011] <http://www.memsic.com/support/documentation/wireless-sensor-networks/category/7-datasheets.html?download=141%3Amda300>.

[23] The MicaZ Datasheet. [Accessed: 20 October 2011] <http://www.memsic.com/support/documentation/wireless-sensor-networks/category/7-datasheets.html?download=148%3Amicaz>.

[24] S. Grgic, M. Mrak, and M. Grgic, "Comparison of JPEG Image Coders," in *Proceedings of the 3rd International Symposium on Video Processing and Multimedia Communications*, Zadar, Croatia, 2001, pp. 79-85.

[25] M. Rabbani and R Joshi, "An Overview of the JPEG2000 Still Image Compression Standard," *Signal Process: Image Commun*., vol. 17, 2002, pp. 3-48.

[26] H. Man, A. Docef, and F. Kossentini, "Performance Analysis of the JPEG2000 Image Coding Standard," *Multimed. Tools and Appl.*, vol. 26, 2005, pp. 27-57.

[27] E. Gurses, O. B. Akan, "Multimedia Communication in Wireless Sensor Networks", *Ann. of Telecommun. 60,* 2005, pp. 799-827.

[28] F. Awad, O. Banimelhem, and N. Al-Rousan. "The Potential of Using Network Coding with Geographical Forwarding Routing for Wireless Multimedia Sensor Networks," in *Proceedings of the 11th IEEE International Conference on Scalable Computing and Communications*, Pafos, Cyprus, 2011, pp. 9 - 14.

Moad Y. Mowafi received the B.Sc. and M.Sc. degrees in Electrical Engineering from Jordan University of Science and Technology, Irbid, Jordan, and Ph.D. degree in Computer Engineering from The University of New Mexico, Albuquerque NM, USA, in 1996, 1998, and 2003, respectively. Presently he is an Assistant Professor with the Department of Network Engineering and Security, Jordan University of Science and Technology, Irbid, Jordan. His research interests include Multimedia Networking, Wireless Networks, and Wireless Multimedia Sensor Networks.

Fahed H. Awad received his B.Sc. degree in Electrical Engineering from the Jordan University of Science and Technology (JUST), Irbid, Jordan in 1990 and his M.Sc. and Ph.D. degrees in Electrical Engineering from the University of Wisconsin Milwaukee (UWM) in 1994, and 2002, respectively. He worked as a Teaching and Research Associate at UWM (1992-1996), a Senior Network Design Engineer in the cellular service provider industry (1996-2001), a Senior Research Scientist at Rockwell Automation HQ, Milwaukee, Wisconsin, USA (2001-2004), an Assistant Professor at the Department of Computer Engineering at JUST (2004-2009), and he is currently an Assistant Professor and a Chairman at the Department of Network Engineering and Security at JUST (since 2009). His research interests are in the areas of wireless networking and robotics applications.

Eyad S. Taqieddin received his BSc in Electrical Engineering from Jordan University of Science and Technology (JUST) in 1999. He received his M.Sc. and Ph.D. in Computer Engineering from the University of Missouri-Rolla in 2002 and 2007, respectively. Currently, he is an Assistant Professor at the Department of Network Engineering and Security at JUST. His research interests include Wireless Multimedia Sensor Networks, Network Security, and Embedded Systems.

Omar Q. Banimelhem received his BSc and M.Sc. in Electrical Engineering from Jordan University of Science and Technology (JUST), Irbid, Jordan in 1995 and 1998 respectively. He received his Ph.D. in Electrical and Computer Engineering from Concordia University, Montreal, Canada in 2006. From 2006 to 2009, he worked as an Assistant Professor at the department of Computer Engineering at JUST. Currently he is working as an Assistant Professor, Department of Network Engineering and Security, Irbid, JUST. His research interests focus on Computer Networks, Wireless Multimedia Sensor Networks and Artificial Intelligence Applications.