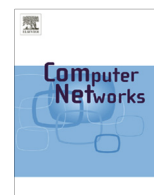




ELSEVIER

Contents lists available at [SciVerse ScienceDirect](http://SciVerse.ScienceDirect.com)

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Object extraction scheme and protocol for energy efficient image communication over wireless sensor networks



Duc Minh Pham*, Syed Mahfuzul Aziz

School of Engineering, University of South Australia, Mawson Lakes, SA 5095, Australia

ARTICLE INFO

Article history:

Received 13 June 2012

Received in revised form 2 July 2013

Accepted 3 July 2013

Available online 11 July 2013

Keywords:

Wireless sensor networks

Image communication

Energy efficiency

Object extraction

Field Programmable Gate Array (FPGA)

ABSTRACT

To date, wireless sensor networks lack the most powerful human sense – vision. This is largely due to two main problems: (1) available wireless sensor nodes lack the processing capability and energy resource required to efficiently process and communicate large volume of image data and (2) the available protocols do not provide the queue control and error detection capabilities required to reduce packet error rate and retransmissions to a level suitable for wireless sensor networks. This paper presents an innovative architecture for object extraction and a robust application-layer protocol for energy efficient image communication over wireless sensor networks. The protocol incorporates packet queue control mechanism with built-in CRC to reduce packet error rate and thereby increase data throughput. Unlike other image transmission protocols, the proposed protocol offers flexibility to adjust the image packet size based on link conditions. The proposed processing architecture achieves high speed object extraction with minimum hardware requirement and low power consumption. The system was successfully designed and implemented on FPGA. Experimental results obtained from a network of sensor nodes utilizing the proposed architecture and the application-layer protocol reveal that this novel approach is suitable for effectively communicating multimedia data over wireless sensor networks.

© 2013 The Author. Published by Elsevier B.V. Open access under [CC BY-NC-ND license](http://creativecommons.org/licenses/by-nc-nd/3.0/).

1. Introduction

In recent years, Wireless Sensor Networks (WSNs) have attracted significant research interests [1–3]. A WSN can be defined as a network of a large number of spatially distributed, small, low cost and low power nodes, which can sense the environment and wirelessly communicate the information gathered to other nodes. The collected information is forwarded, normally via multiple hops, to a sink (or controller or monitor) node that uses the information locally or transmits it to other networks (e.g., Internet)

through a gateway. WSNs are normally comprised of scalar sensors capable of measuring physical phenomenon such as temperature, pressure, light intensity, and humidity.

With the availability of cheap, small size and low-power CMOS cameras and microphones, there is a strong interest in deploying WSNs for multimedia communication. Such Wireless Multimedia Sensor Networks (WMSNs), with the ability to gather multimedia information from the surrounding environment, is providing the impetus for extending the capabilities WSNs for many new applications such as advanced environmental monitoring, advanced health care delivery, traffic avoidance, fire prevention and monitoring, and object tracking [4,5]. However, the challenge is how to handle the large volume of multimedia data using sensor nodes that are severely constrained in both processing capability and energy. In addition to developing energy aware multimedia processing algorithms and architectures, it equally important to develop efficient communication strategies [5,6] to maximize the network lifetime while

* Corresponding author. Tel.: +61 8 830 23241; fax: +61 8 830 23384.

E-mail addresses: minh.pham@unisa.edu.au (D.M. Pham), mahfuzul.aziz@unisa.edu.au (S.M. Aziz).

meeting the application specific QoS constraints such as latency, packet loss, bandwidth, and throughput.

Several proposals have been put forward to achieve image transmission over WSN [7–9]. Ref. [7] aims at providing a reliable, synchronous transport protocol (RSTP), with connection termination similar to TCP. Ref. [8] presents an energy-efficient and reliable transport protocol (ERTP) with hop-by-hop reliability control, which adjusts the maximum number of retransmission of a packet. Ref. [9] proposes another reliable asynchronous image transfer (RAIT) protocol. It applies a double sliding window method, whereby network layer packets are checked and stored in a queue, to prevent packet loss. With protocols providing reliability at the transport [8] or network layers [9], erroneous packets at the application layer can still be forwarded to the base station, requiring retransmission and associated energy cost [10]. In addition, the above protocols do not take into account the practical resource limitations (memory, processing capacity, energy) of the wireless sensor nodes. Consequently, in [11], the authors stated that multi-hop transmission of JPEG2000 images is not feasible due to interference and packet loss. This statement is also cited by other literature [10,12]. In this paper, image transmission over multi-hop WSN is proved to be feasible, using a reliable application layer protocol that reduces packet error rate and retransmissions. The proposed protocol uses an effective queue control strategy with built-in CRC, which helps achieve higher data throughput in error prone environments.

In terms of WMSN hardware, the few off-the-shelf motes reported to date include the Cyclops motes [13], WiSN motes [14], Panoptes motes [15] and SensEye motes [16]. Each of these motes consists of a CMOS camera, a Commercial Off-The-Shelf (COTS) processor, a wireless transceiver and a battery. The COTS processor, being a general purpose processor, often has redundant hardware blocks, which are not utilized by the WSN mote. This leads to higher energy consumption and less than optimum operation. The COTS processor runs at low frequency in order to keep the power consumption low. The limitations in processing speed and memory capacity of the traditional sensor nodes restrict the image processing and transmission capability of these WSN motes. Therefore, implementing complex image processing tasks on these WSN motes is almost impossible.

Instead of using COTS processor, the possibility of using FPGAs to implement WMSN nodes have been explored in recent literature [17–19]. In [17], the authors present a JPEG image compression system for a WMSN node on Altera EP2C35 FPGA using NIOS II soft-core microprocessor. Because the FPGA also incorporates networking functionalities, there is no mechanism to put the system on the FPGA to sleep mode to reduce energy consumption. In [18,19], a dedicated FPGA hardware is designed to handle image processing. These systems require another external (off-the-shelf) microprocessor to perform communication and system operations. This is problematic because the external processor is not optimized to work in a WSN node and therefore has hardware redundancies. There is also significant communication delay between the two processors. In this research, a novel architecture is proposed in which both image processing and networking functions

are handled by processors implemented on one FPGA. This approach helps to optimize the operation of the WMSN node, where the image processing block is designed for high performance (high speed), and can be turned on and off as required to minimize energy consumption.

The processing architecture presented in this paper provides high processing speed, and consumes much less energy for both processing and communication of images compared to COTS-based WMSN motes. A simple and efficient background subtraction [20] is applied to detect and extract the object area of interest. Only a portion of the image that contains the updated object is transmitted over the network. Experimental results demonstrate that the total energy consumed for processing and transmitting an image, using the combination of the proposed application-layer protocol and energy efficient processing architecture, is much less than that required for transmitting an entire image although the latter does not involve any image processing overhead.

2. System architecture

Fig. 1 presents the overall architecture of the proposed WMSN processing system. It consists of two major processing elements: a customized networking processor and an image processing block. The network processor performs some standard operations found in a typical processor as well as customized instructions to support the operations of the wireless transceiver. Since the network processor needs to run continuously to keep track of network traffic, it operates at a low clock frequency in order to keep the power consumption low.

On the other hand, due to the complexity associated with most image processing operations, the image processing block must run at a high frequency to process images at a high speed. This will lead to high power dissipation. However, if the processing time is short and if the image processing block is put to sleep mode when inactive then the overall energy consumption for image processing can be kept low. This is the philosophy behind the proposed separation of the network processor and the image

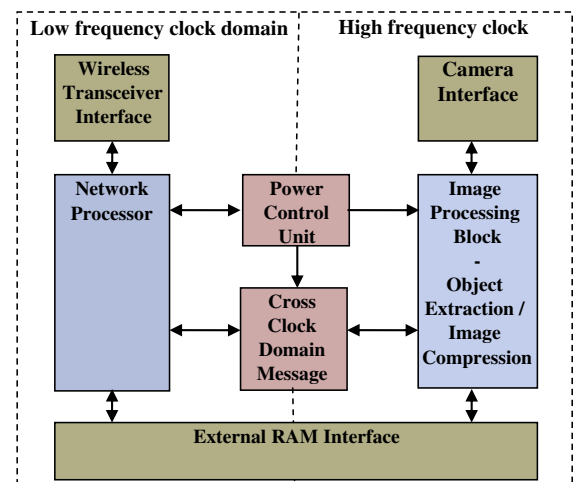


Fig. 1. Proposed WMSN processing system architecture.

processing block. By default, the image processing block is in the *inactive mode* (sleep mode with suppressed clock source), and can be quickly set into the *active mode* by the network processor whenever an image processing task is required. It receives specific hardware instructions from the network processor, quickly finishes the specified image processing function and goes back to the *inactive (sleep) state*.

To reduce the energy consumption further, the following section presents an innovative and resource efficient hardware architecture for extraction of areas of interest (updated objects) from the images captured by the camera nodes. Only the updated objects are transmitted as opposed to transmitting the whole image, thereby significantly reducing the image data to be communicated. The key aim of the hardware architecture is to reduce energy consumption for both processing and transmission of an image below that required to transmit the entire unprocessed image. The detailed design and optimization of the hardware blocks for object extraction and image compression are presented in the next two sections. The functions of these blocks are to provide the ability to process, detect, extract and compress the image captured by the camera sensor node in an energy efficient manner.

3. Object extraction architecture

In WMSN applications, the camera mote often has a fixed frame of view. In this case, to detect moving (updated) objects, background subtraction is a commonly used approach [20]. The basic concept of background subtraction is to detect the objects from the difference between the current frame and the background image. The background image represents a static scene of the camera view without any moving objects. An algorithm must be applied to keep the background image regularly updated to adapt to the changes in the camera view.

A number of background subtraction methods have been introduced in the literature. Some of them are: Running Gaussian Average, Temporal Median Filter, Mixture of Gaussians, Kernel Density Estimation and Sequential Kernel Density Approximation [21]. Out of these methods, the Running Gaussian Average appears to have simple calculation, fastest processing speed and lowest memory requirements [21]. For these reasons, the Running Gaussian Average method has been further optimized for FPGA implementation and incorporated into the proposed system for WMSN applications. The optimization of the subtraction algorithm and that of the memory system is presented in this section.

3.1. Memory design and address mapping

Image processing requires large amount of memory to store image data. Efficient memory design and address mapping are always vital tasks, because inefficient memory organization leads to low memory utilization, complex addressing and high processing time. These factors adversely affect the energy requirement of wireless sensor nodes. Therefore, compact memory systems with high

memory utilization and simpler address mapping are desirable for Wireless Multimedia Sensor Networks. In the proposed system, 1 MByte of asynchronous RAM is used. The memory array includes two 256Kx16-bits 10 ns SRAM device [22]. An external RAM interface, which provides a simple synchronous communication to the processing system, is designed as shown in Fig. 2. To the processing system, it would appear that there is only one RAM of size 512Kx16-bits. The memory data and address mapping is designed to achieve both good performance and high memory utilization.

The target image is defined as a gray scale image of 640×480 pixels. Each pixel in the image is represented by an 8-bit value of its intensity. The system design requires two full images to be stored in the external RAM: the background image and the current image. The detailed design of the external RAM data mapping and address mapping is illustrated in Fig. 3.

Each location in the external RAM is 16-bit wide, the 8 MS bits contain pixel data of the background image, and the 8 LS bits contain pixel data of the current image. Each pixel in an image is located by its row address and column address. The address to access the external RAM is 19-bit long, the 10 LS bits (LSB) form the column address and the 9 MS bits (MSB) form the row address. This can provide column and row addresses of up to 1023 and 511 respectively. This means that for an image of 640×480 pixels, there are some unassigned locations in the external RAM. These locations can be utilized to store other signals, which will be described in Section 3.3. All the information required by the image processing block for any pixel is read or updated in only one RAM access. This helps to simplify memory operation and enhance memory performance.

3.2. Background subtraction

The Running Gaussian Average model [23] is based on ideally fitting a Gaussian probability density function on the last n values of a pixel. The background pixel value at

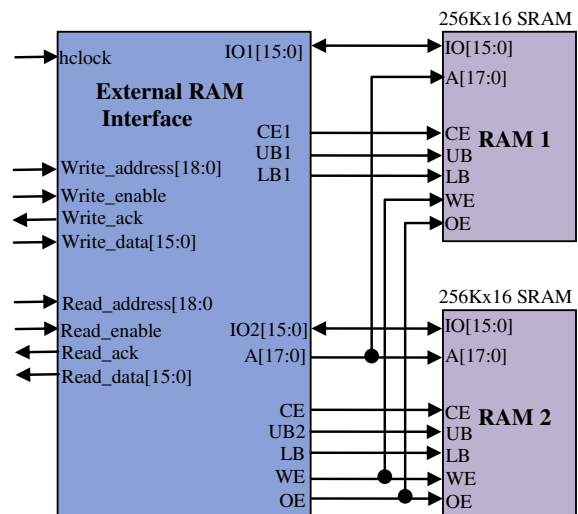


Fig. 2. External RAM interface.

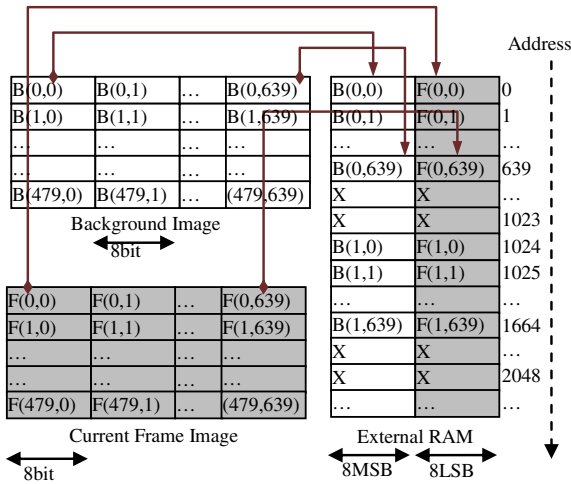


Fig. 3. External RAM data and address mapping.

frame n is updated by the running average calculation shown in the following equation:

$$B_n = (1 - \alpha)B_{n-1} + \alpha F_n \tag{1}$$

where B_n is the updated background average, F_n is the current frame intensity, B_{n-1} is the previous background average, α is an updating constant whose value ranges between 0 and 1 and represents a trade-off between stability and quick update. Eq. (1) can be rewritten as:

$$B_n = B_{n-1} + \alpha(F_n - B_{n-1}) \tag{2}$$

In FPGA design, to reduce hardware complexity and power consumption and to increase operating frequency, multiplication of real numbers is avoided. In this architecture, α value is chosen in the form of $1/2^k$. Thus multiplication by $1/2^k$ is required, which can be easily implemented by a simple bit shifting circuit. With this implementation approach, Eq. (2) can be rewritten as:

$$B_n = B_{n-1} + \frac{1}{2^k}(F_n - B_{n-1}) \tag{3}$$

At each new frame, the pixel is classified as foreground (i.e. belong to an updated object) if the condition below is true.

$$|F_n - B_{n-1}| > Thr \tag{4}$$

where Thr is the updating threshold.

The result of the calculation $(F_n - B_{n-1})$ in (3) can be reused in (4). This, along with a multiplier free implementation, leads to an optimized design. The results in (4) can be used to decide whether a background image needs to be updated or not. An optimized hardware implementation of (3) and (4), as described above, is shown in Fig. 4. Table 1 provides a description of the signals of the background subtraction hardware block and their specific functions. By default, B_n is calculated for every pixel in each frame. However, the decision to update the previous average (B_{n-1}) depends on the value of the *update* signal, which is a 1-bit signal. To aid in this decision making process it is

essential to save the values of the *update* signals for all pixels in a frame in an array in the external RAM.

3.3. Update memory organization

The *update* signal in Fig. 4 is saved in the external RAM to provide information required to calculate the location of the updated objects to be extracted. As stated in Section 3.1, for an image of 640×480 pixels, the highest row address for storing pixel intensity is 479 or “11101111” in binary. The remaining upper memory locations, addressed by the 9-bit row address starting from “11110000”, are used to store the *update* signals. Because each RAM location is 16-bit wide, so each of these locations store *update* information for 16 pixels. The scheme used to address the *update* signals is shown in Fig. 5, where a RAM location is accessed using an address composed of ‘1111’ in the 4 MSB bit positions followed by the 13 MS bits from the ‘original pixel address’. The least significant four bits of the ‘original pixel address’ identify the column where the corresponding *update* signal is stored. This mapping method helps maximize the utilization of the external RAM space. In the proposed system, the whole object extraction solution for a 640×480 image fits in 1 MByte of external RAM.

3.4. Object extraction

Any updated object in the camera view is extracted by detecting portions of the current frame that is significantly different from the background image. An efficient object detection algorithm has been implemented for this purpose. It is illustrated by an example shown in Fig. 6. The algorithm involves row and column scans to determine if the number of consecutive differences (1s) is greater than a pre-determined *difference threshold*. In the example of Fig. 6, the difference threshold is set to 3. During row scan, the first pixel location of the first 3 consecutive 1s in the 2nd row and the last pixel location of the last 3 consecutive 1s in the 7th row are recorded. Similarly, in column scan, the first pixel location of the first 3 consecutive 1s in the 5th column and the last pixel location of the last 3 consecutive 1s in the 12th column are recorded. The location of the updated object is then determined by the locations recorded in the row and column scans. In Fig. 6, the object extraction area is from the 2nd to the 7th row and from the 5th to the 12th column. Noise can cause some false *updates* in the background subtraction, but they are usually in a small group (1 or 2) of consecutive pixels. The row and column scanning functions implemented in the proposed object detection algorithm eliminate such noise.

3.5. Implementation results

A hardware model for the proposed object extraction scheme was developed using the Verilog hardware description language, and was successfully synthesized and tested on various Xilinx FPGAs. It consists of the background subtraction block shown in Fig. 4, and the row/column scanning and threshold comparison blocks. It does not require any internal memory. Table 2 summarizes the synthesis results for the proposed object extraction

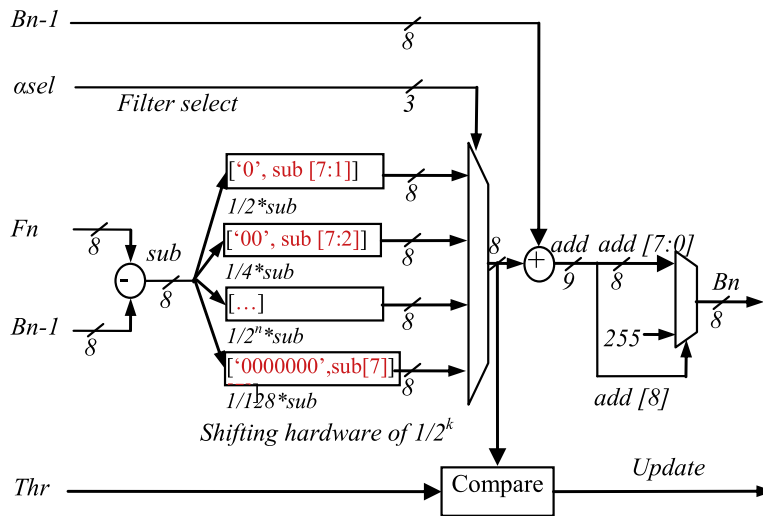


Fig. 4. Hardware architecture for background subtraction.

Table 1
Control signals for background subtraction.

Signal name	Description
B_{n-1}	8-Bit pixel value of the background image. This is extracted from the 8 MS bits (8 MSB) of the external RAM data as shown in Fig. 3
F_n	8-Bit pixel value of the current image frame coming from the camera interface
αsel	3-Bit updating constant. The value of α in Eq. (2) is defined by $1/2\alpha sel$
Thr	8-Bit updating threshold of Eq. (4)
B_n	Updated background pixel value. This will be used to update the background pixel if the update signal is 1, otherwise the old background pixel value will be retained
$Update$	This signal is used to determine whether the background pixel needs to be updated. It is also used to allocate the position of updated objects in the current image frame

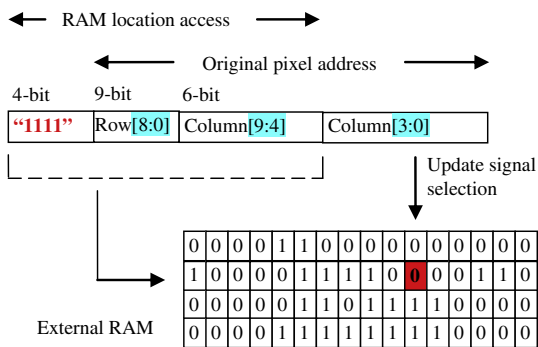


Fig. 5. Address mapping for Update information stored in external RAM.

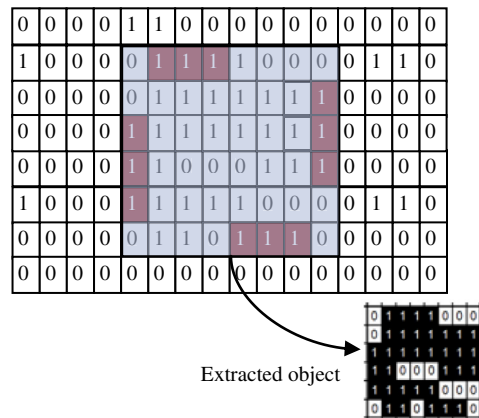


Fig. 6. Example of object extraction for a 16 × 8 image.

architecture and for other architectures reported in the literature that have synthesized similar object extraction functions [18,24]. Both [18,24] have synthesized their architectures for Altera FPGAs. Hence, for a fair comparison, we report synthesis results of the proposed architecture on the same Altera FPGAs. Clearly, the proposed architecture requires significantly less FPGA resources (Logic Elements) compared with the other designs [18,24]. The maximum frequency of operation (F_{max}) is 125.36 MHz on Altera Cyclone III, the highest reported so

far, due to its multiplier free and optimized hardware architecture.

3.6. Discrete wavelet transform

After an updated object has been extracted from the raw image captured by the camera, it can be compressed

Table 2
Synthesis results for the background subtraction architecture.

FPGA	Work	Image size	FPGA resource	Internal memory	External memory	F_{\max}
Altera Cyclone III EP3C25F324C6	[18]	640 × 480	2819 LE (11%)	122,880 bits	Used but size was not reported	25
	Proposed	640 × 480	174 LE (<1%)	0	1 MByte	125.4
Altera APEX20KE EP20K200EQC24	[24]	240 × 120	15142 LE (179%) ^a	106,496 bits	192 KByte	40
	Proposed	640 × 480	181 LE (<3%)	0	1 MByte	45.5

^a This design requires two APEX20KE FPGAs.

further by the image compression block (see Fig. 1). According to the JPEG2000 image compression scheme [25], a full image compression block typically includes a 2-D DWT processor and an encoder. The DWT processor allows separation of the low and high frequency components of the image into four sub-bands as shown in Fig. 7, where the first quadrant (LL_1) represents the low frequency components (smooth coefficients). Transmitting only the LL_1 sub-band reduces the time, bandwidth and energy required for transmission compared to the whole image (LL_0). Further reduction can be achieved using multiple levels of DWT processing on the low frequency sub-band (LL_1) as per the JPEG2000 standard [26]. Transmitting only the LL sub-band, where the majority of the important features of the image are retained, is adequate for promising WSN applications such as surveillance, object tracking and environmental monitoring. Using only the LL sub-band is an established technique and the full image can be reconstructed from it without much loss [25].

To demonstrate the energy saving concepts in a simple manner, the image compression block used in this work only incorporates a 2-D DWT processor, and not an encoder or other blocks typically associated with image compression. A JPEG2000 integer lossless 5–3 filter has been implemented for DWT transformation. The lifting scheme [26], a JPEG2000 compliant technique, is used to compute DWT, because it leads to less computational complexity and less memory requirements [27]. A highly parallel and pipelined DWT architecture has been developed and tested on various Xilinx FPGAs [28]. The details of this architecture can be found in [28].

Synthesis results for up to three levels of 2-D DWT processor implementation are given in Table 3. When synthesized on a Xilinx Spartan 3E FPGA, the required CLB slice count is 185 for 1-level transformation. This is the lowest CLB slice count reported to date. The maximum frequency of operation (F_{\max}) is same for all levels of transformation due to the parallel architecture of the multi-level DWT processor [28]. As can be seen from Table 3, the F_{\max} achieved by the DWT architecture is much higher compared with all other FPGA implementations reported to date [27,29–31]. At least 50% reduction in CLB slice count is achieved by

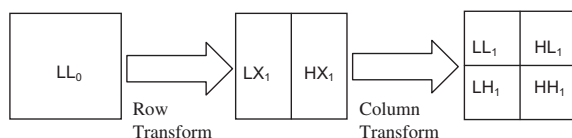


Fig. 7. Single level 2-D DWT transformation.

Table 3
Synthesis results for the DWT processor.

Work	FPGA	Level	CLB	Proposed CLB	F_{\max}	Proposed F_{\max}
[27]	XC2v500	1	602	185	54.9	110.9
[29]	XC2S50	1	391	185	24.3	90.4
[30]	XCV300e	3	2992	1766	20.0	93.6
[31]	XCV1000	1	5283	185	52.8	98.4

the design for 1-level transformation compared to the other implementations [27,29–31].

4. A robust image transmission protocol

The problem with transmission of images over WSN is that the image packets are sometimes lost or arrive in wrong order due to channel errors, congestion and limitation of memory in the intermediate (router) nodes. While end-to-end packet loss ratio can be reduced by applying reliability mechanisms such as automatic repeat request (ARQ) [32] at the network layer [9] or transport layer [7], multi-hop transmission of compressed images over WSN requires even higher level of reliability because compressed images cannot tolerate packet loss [33]. In multi-hop WSN transmissions, errors in compressed image packets at the application layer are multiplied due to channel errors, congestion and limitation of memory in the intermediate (router) nodes. Resolving this issue requires a suitable transmission protocol to prepare the network for burst transmission, to packetize the image to be transmitted, to assign frame number and to embed appropriate frame control parameters. One of the most effective ways to address this is to ensure that image packets that contain errors are not forwarded. This requires a mechanism to check for errors in the image packets at the intermediate nodes along the transmission route.

As was stated in Section 1, the available literature [10–12] has reported that multi-hop transmission of JPEG2000 images is not feasible due to interference and packet loss. From our practical experiment with a multi-hopping wireless network of the proposed FPGA-based nodes, we can confirm that a JPEG2000 image could not be reconstructed at the base station when the standard 802.15.4 protocol was used. In the remainder of this section, a robust yet simple and energy efficient image transmission protocol for WMSN is proposed, taking into account all the practical aspects mentioned above. The transmission protocol between the base station and the camera node is illustrated in Fig. 8. The basic idea behind this protocol is dividing

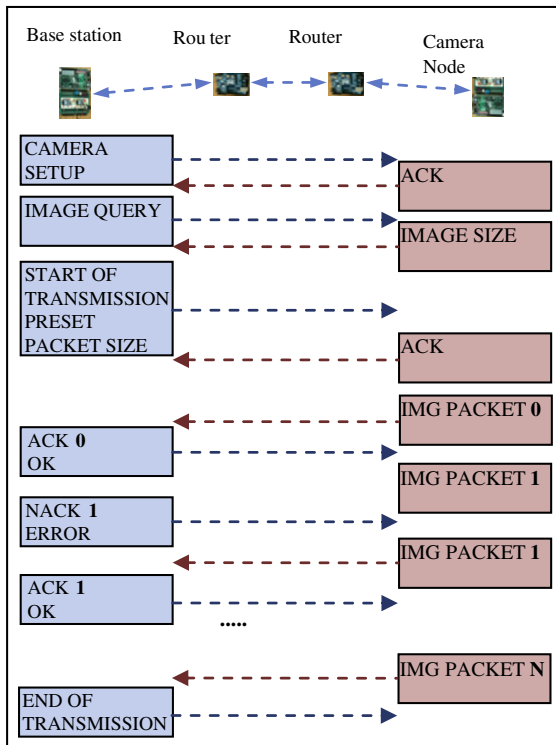


Fig. 8. Energy efficient image transmission protocol for WMSN.

the image into a number of packets, and embedding the following: packet ID, packet control and error detection. The structure of the protocol messages implemented at the application layer is shown in Table 4.

Even with the above application layer protocol incorporated within 802.15.4, practical experiments show that when transmitting image packets through multi-hop, the high packet error rate necessitates frequent retransmission, which is inefficient in terms of energy and bandwidth. The high packet error rate arises due to the following reasons.

- Unpredictable data throughput of the wireless channels due to variations in noise levels.
- Limitation of the size of packet queue in routers, because the sensor nodes have limited memory.
- Inability to adjust image packet size based on link conditions to minimize packet error rate.

In contrast with other image transmission protocols [7–9], where multiple nodes can start data transmission at the same time, the proposed protocol allows only one node to transmit packets to the base station. In Fig. 8, when the base station starts an image transmission, first it broadcasts a small `START_OF_TRANSMISSION` message to all nodes in the network. This message will also predefine the size of the image packet and forbid any nodes which are not part of the active image transmission link to send anything until they receive the `END_OF_TRANSMISSION` message. This mechanism ensures that only one image

Table 4
Application layer control messages.

Type of message	Message structure			
IMAGE PACKET	0xAAAA	Packet ID 2 bytes	Image data N bytes	CRC-8 1 byte
CAMERA SETUP	0xAA00	Camera parameters – 4 bytes		
IMAGE QUERY	0xAA01			
IMAGE SIZE	0xAA02	Image size – 2 bytes		
ACK	0xAA03	Packet ID – 2 bytes		
NACK	0xAA04	Packet ID – 2 bytes		
START OF TRANSMISSION	0xAA05	Packet size – 1 byte		
END OF TRANSMISSION	0xAA06			

packet is sent at a time to avoid collision and congestion, and therefore to reduce packet loss and the associated energy cost of retransmission. Image packet size can be adjusted (16–256 bytes) based on link conditions, i.e. noise levels, network topology, location of transmitting nodes. Examination of the `IMAGE_PACKET` in Table 4 reveals that the packet overhead (for protocol messages) is 5 bytes irrespective of the image data packet size.

The proposed protocol implements a packet queue control mechanism to reduce the *packet error rate* (*PER*) and increase data throughput. With queue control, every data packet is checked for correctness using Cyclic Redundancy Checks (CRC) before deciding whether to forward the packet or not. This reduces the *PER* at the base station. Without any queue control, the image data flow is transparent to the intermediate nodes. Therefore, the intermediate nodes have no means to identify erroneous packets and to stop such packets from being forwarded to the base station. This increases the *PER*, decreases *throughput* and increases the *energy cost* associated with retransmission. Table 5 shows that the proposed queue control strategy with built-in CRC significantly reduces *packet error rate* and increases data *throughput* for multi-hop communications. Clearly, in our tests, small sized packets show better performance (*throughput*) than large sized packets. However, this may not be the case in other environments, for example where the SNR is lower. Using small sized packets will lead to relatively larger packet overhead, but significantly reduce the *energy cost* of retransmission compared to that incurred for retransmission of very large packets. The throughput in Table 5 is calculated based on the actual image data received at the base station over time during practical tests. The tests have shown that the proposed protocol with queue control can provide reliable and energy-efficient image transmission in error-prone wireless environment, even though the protocol is much simpler in comparison with other image transmission protocols proposed for WSN, such as RSTP [7], ERTTP [8] and RAIT [9].

Among the three protocols mentioned above, only RAIT [9] implemented a cross layer optimization and packet queue control similar to the protocol proposed in this paper, although it [9] does not incorporate any error detection capability at the intermediate nodes. Consequently, a comparison of the overall performance of the proposed protocol against RAIT [9] is presented next. For this, simple

Table 5

Test results on packet error rate (PER) and throughput.

Hops	Packet size	Without queue control		With queue control	
		PER (%)	Throughput (kbps)	PER (%)	Throughput (kbps)
2	16	29.7	47	8.6	98
	64	44.7	27	11.7	67
	128	57.6	15	18.1	45
	256	74.5	9	34.4	22
3	16	32.1	44	22.1	44
	64	68.3	25	26.7	28
	128	68.3	6	38.3	15
	256	82.1	5	54.4	7

end-to-end Matlab simulation models have been developed. The topology of the network modeled in Fig. 9. According to the transmission strategy adopted in the proposed protocol, only one node is allowed to transmit once a link is established. Therefore, for a fair comparison, the simulation is set up to allow only one camera node to transmit image over a link for either protocols. For RAIT, we use the same queue size of 33 packets as reported in [9].

For transmitting a gray scale 128×128 image, simulation results on the queue size at intermediate node #2 and total transmission time are shown in Fig. 10. Clearly, the proposed protocol starts and finishes the transmission sooner because, as described earlier, it involves simpler synchronization by broadcasting a small START_OF_TRANSMISSION message to all nodes in the network. In case of the RAIT protocol [9], the queue of router 2 is always full (33 packets) whereas the queue for the proposed protocol is only 2 packets. Therefore, queue monitoring in RAIT [9] involves far more complexity and associated energy costs compared to the proposed protocol. As opposed to the simulation scenario presented above, where only one node was allowed to transmit over a single link, RAIT allows multiple nodes to transmit at the same time. With increasing number of nodes transmitting at the same time, competition and congestion will increase, and as a consequence, more and more packets will be lost. This will lead to further increase in retransmission and energy consumption. In the protocol proposed in this paper, this situation will not arise, because only one node is allowed to transmit at a time over a single link, thereby significantly reducing packet loss and the associated energy cost.

5. Network set up, testing and results

The WMSN processing architecture presented in this paper, including both the networking processor and the

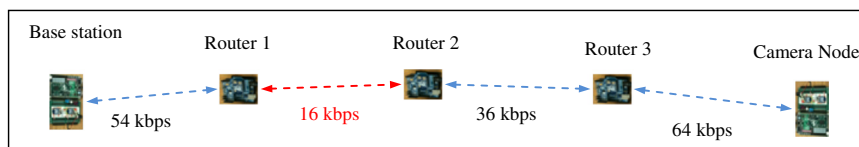
image processing block, was implemented on a FPGA platform containing a Xilinx Spartan-3 FPGA. Two such prototypes are shown in Fig. 11(a) and (b). These two prototypes are connected to Digi Xbee [34] and Microchip MRF24J40 [35] transceivers respectively. The two FPGA prototypes have been used in conjunction with the COTS based WSN nodes (uses Atmel ATmega328p microprocessor) shown in Fig. 11(c) to set up a network for conducting various tests. Each camera node includes a CMOS camera and a battery. The CMOS camera is a small size, low cost and low power OmniVision OV7640/8 VGA CMOS color sensor [36].

Fig. 12 shows the experimental set up of a WMSN with 10 nodes including camera nodes utilizing the proposed image processing prototype (CAM, CAM2, etc.). All multimedia data captured by these nodes could be successfully transferred and monitored on a base station connected to a PC. The background image and the extracted objects were received and reconstructed by software running on the PC. The graphical user interface (GUI) is shown in Fig. 13 along with the background image and the extracted objects.

5.1. Power analysis

As a result of the low CLB slice count (see Tables 2 and 3), the image processing block consumes very small amount of power. When synthesized on a Xilinx Spartan3 XC3s1000 FPGA, Table 6 shows the power consumption of the image processing block in the active state for both extraction and DWT, as reported by Xilinx Power Analyzer. These results are reported for an operating frequency of 50 MHz, at a supply voltage of 3.3 V and ambient temperature of 25 °C. In the inactive state, the power consumption of this block is only 0.3 mW. The results reveal that the proposed architecture is suitable for use in low power applications due to low hardware requirement and very low power consumption.

The image processing block is activated by the network processor only when required. The amount of time the image processing block needs to complete its operation depends on the size of the new (extracted) object in the image. For these reasons, it is difficult to measure the real-time power consumption on the actual hardware. Instead, we have used PicoScope to record the total power consumption over a period of time. Fig. 14 shows the total power consumed by the FPGA board as a function of time. When idle, the standby power consumed by the entire FPGA board is ~54 mW. The total power consumption rises to ~82 mW during transmission of an image frame. The difference of ~28 mW is the power required for transmission of an image frame. Fig. 15 is another real-time plot capturing power consumed during object extraction and

**Fig. 9.** Simulation set up for comparison with RAIT [9].

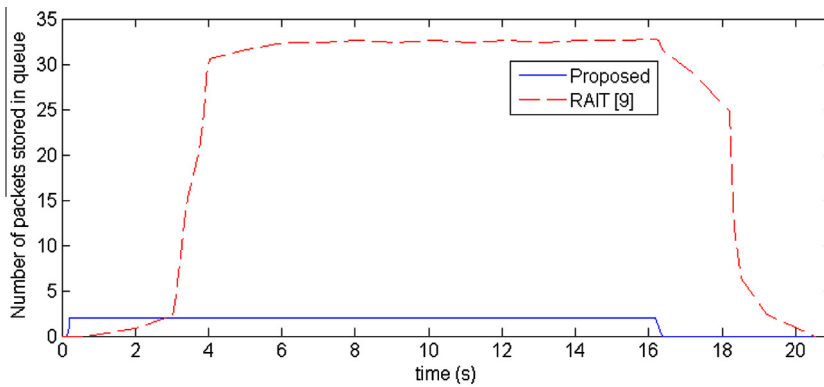


Fig. 10. Number of packets stored in the queue of router node #2.

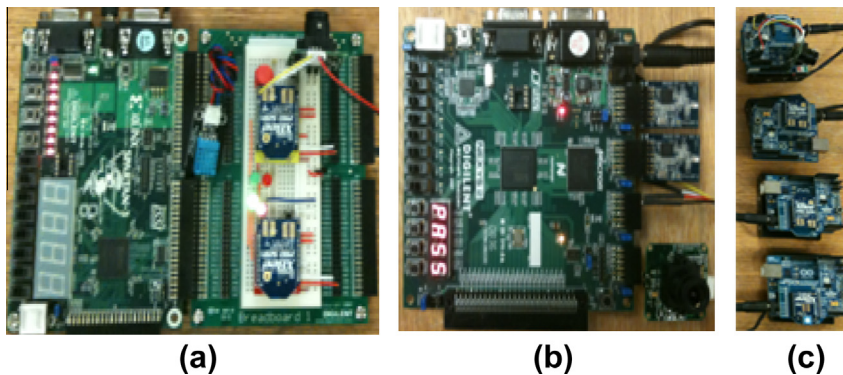


Fig. 11. WMSN prototype nodes (a) Spartan-3 FPGA and Xbee transceiver, (b) Spartan-3E FPGA and MRF24J40 transceiver and (c) ATmega328p microprocessor and Xbee transceiver.

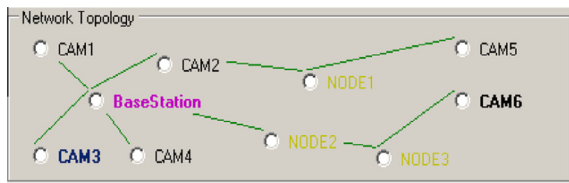


Fig. 12. Topology of a WMSN under test.

DWT processing for a background image of 640×480 pixels when the image processing block is operating at 50 MHz. These operations consume approximately 10 mW. Clearly, the power consumption for image processing (~ 10 mW) is much less than the power consumption for data communication (~ 28 mW).

5.2. Energy simulation

An estimation of energy consumption was done by collecting the energy profiles of the components used in the proposed WMSN system and putting them into a comprehensive simulation. A Matlab simulation model was developed for this purpose. This model is not meant for full network traffic simulation as such, it rather provides a

mechanism to easily estimate the energy consumption associated with the processing and transmission of images of various size. The simulation model incorporates energy consumption estimates for all major constituent blocks, i.e. network processor, image processing blocks and wireless transceiver. The simulation model utilized the packet error rate (PER) and throughput data obtained from the practical measurements conducted on the network of Fig. 12 using the proposed protocol with queue control (see Table 5). As per Fig. 12, the number of hops for image communication was 2 and the image packet size was set to 16. Based on the practical results on power measurements (see Figs. 14 and 15), energy consumption for processing and communication of 1-bit of image information was calculated. This information, combined with the PER, was used to calculate the energy consumption for various image size.

Fig. 16 shows the simulation results on energy consumed by a single node for processing and transmission of images. These results show that the energy used to process an image (object extraction and DWT) is really low compared with the energy used for transmitting an image. The results also show that much less energy is used for communicating smaller images. Therefore, the proposed object extraction and DWT techniques can help significantly reduce the energy required for image communica-

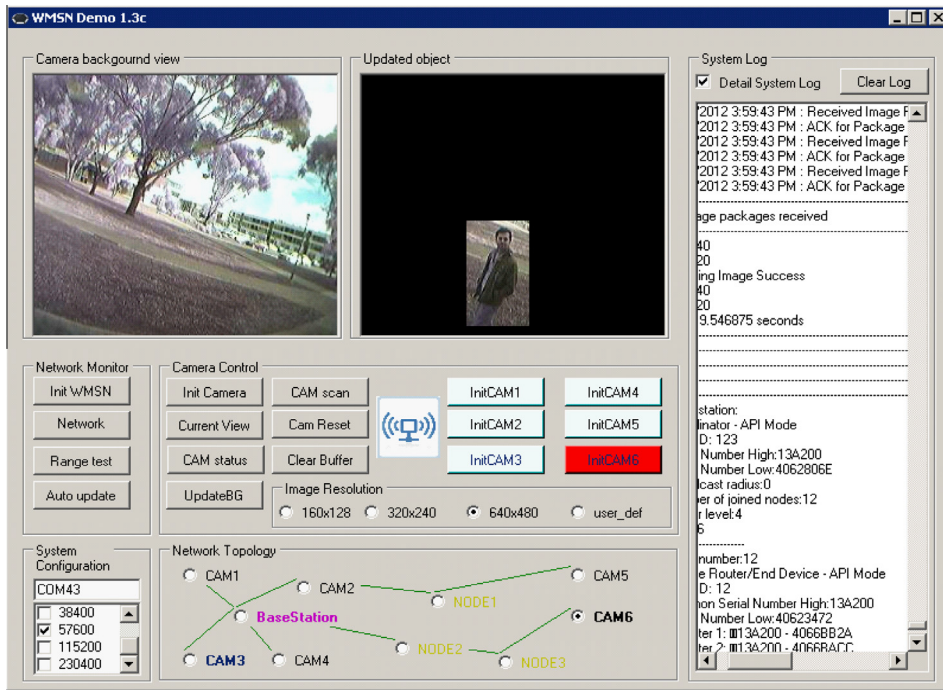


Fig. 13. Background and updated object images shown on the base station GUI.

Table 6

Power consumption of the image processing block.

Clocks	10.11 mW
Logic	4.02 mW
Signals	4.57 mW
BRAMs	3.26 mW
Total	21.96 mW

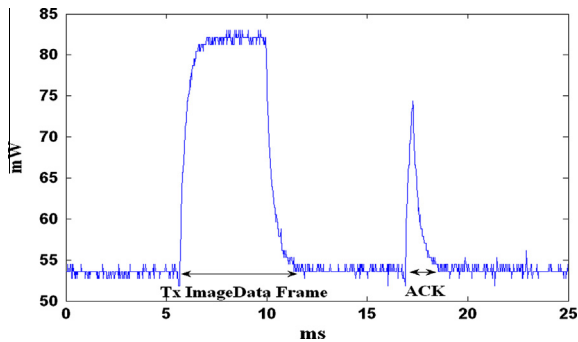


Fig. 14. Real-time plot showing the power consumed during image transmission.

tion by communicating only the updated (extracted) portion of the image. For example, in Fig. 16, the total power required for processing and sending an updated object of size 160×100 is 20 times less than that required for sending the whole image of size 640×480 . Because the energy used by the proposed system for processing the image (object extraction and DWT) is very small, the combined energy consumption for image processing and

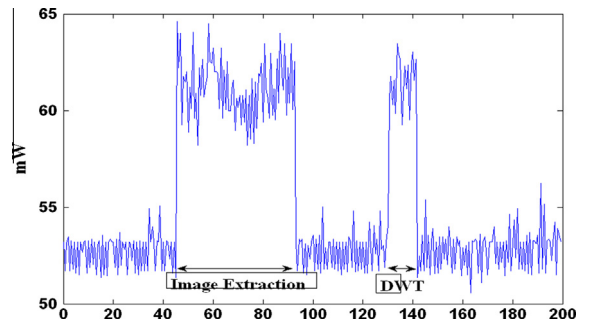


Fig. 15. Real-time plot showing the power consumed during image processing.

communication of the extracted object is significantly less than that required to transmit the raw image, although the latter does not involve any energy consumption for image processing.

5.3. Comparison with COTS-based WSN nodes

To compare the operational characteristics and energy consumption of the proposed WMSN processing system with COTS-based WSN, software was developed to execute the same object extraction and DWT functions on the ATmega328p microcontroller based WSN node (shown in Fig. 11c). In the active mode, the power consumption of the ATmega328p node at 16 MHz, 3.3 V, 25 °C is about 11mW, while that of the proposed architecture on the

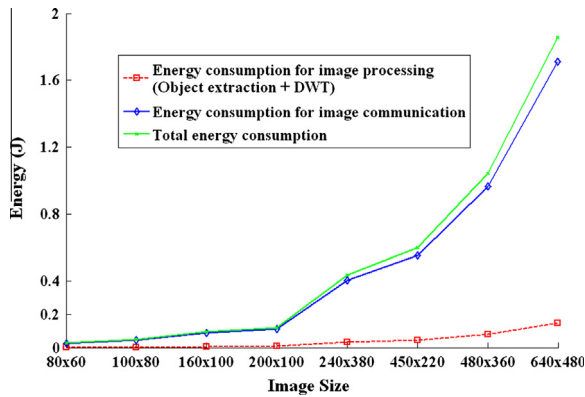


Fig. 16. Simulation of energy consumption of the system.

Table 7

Comparison with COTS-based WSN node performing the same operations.

System	ATmega328p @16 Mhz	Xilinx XC3s1000 @50 Mhz
Number of clock cycles	33,177,600	2,457,600
Time (ms)	2073.6	49.15
Total energy consumption	22.8 mJ	1.08 mJ

XC3s1000 FPGA at 50 MHz is 21.96 mW (from Table 6). The ATmega328p system requires more than 2 s to complete the image processing task, while the proposed processing system completes the same task in only 49.15 ms, i.e. more than 40 times faster. Consequently, the total energy consumption of the ATmega328p node is 22.8 mJ, which is ~20 times higher than that of the proposed system. The energy results are summarized in Table 7. These results demonstrate that the proposed WMSN processing system is highly energy efficient and much faster than COTS processor based systems. In addition to [37], the results presented in this paper have demonstrated that the proposed system is practically suitable for fast and energy efficient processing and communication of images over wireless sensor networks.

While the proposed WMSN processing architecture consumes very little energy for object extraction and DWT operations, it works effectively in detecting any updated objects in the camera view. Also, the proposed architecture requires much less energy for data communication because the size of the transmitted image is reduced significantly. In addition, the proposed transmission protocol contributes to reducing the energy required for image communication by reducing the cost of retransmission of image packets.

6. Conclusions

The challenge to efficiently process and transmit large volume of image data over wireless sensor networks has been addressed in this paper by developing a highly optimized architecture for object extraction and transmitting

the updated object using a robust image transmission protocol. The proposed protocol implements an effective queue control strategy with built-in error detection capability at the application layer. This has led to reduction in *packet error rate* and increase in *data throughput*. The image processing block (object extraction and DWT) runs at a high frequency to provide fast processing, and is activated by a separate network processor only when required to process images. The network processor, which is responsible for executing basic node operations and network instructions, runs at all times and is therefore designed to operate at low frequency. Practical test and simulation results have confirmed the very low energy requirement of the proposed scheme for image processing and communication. In addition, the proposed FPGA-based architecture consumes ~20 times less energy than a COTS-based mote performing the same object extraction and DWT operations. Unlike some recent literature asserting that multi-hop transmission of JPEG2000 images over wireless sensor networks is not feasible, the work presented in this paper has demonstrated that it is indeed feasible. This has been made possible by the combination of the energy efficient protocol and the efficient hardware architecture. In the authors' opinion, solely developing efficient protocols without considering the practical implementation issues and resource limitations of the sensor nodes will make image communication almost impossible to achieve.

References

- [1] D. Culler, D. Estrin, M. Srivastava, Overview of wireless sensor networks, *IEEE Computer, Special Issue in Sensor Networks* 37 (8) (2004) 41–49.
- [2] C. Buratti, A. Conti, D. Dardari, R. Verdone, An overview on wireless sensor networks technology and evolution, *Sensors* 9 (9) (2009) 6869–6896.
- [3] M. Tubaishat, S. Madria, Sensor networks: an overview, *IEEE Potentials* 22 (2) (2003) 20–23.
- [4] S. Stanislava, H. Wendi, A survey of visual sensor networks, *Advances in Multimedia 2009* (2009) 21p. Article ID 640386.
- [5] I. Akyildiz, T. Melodia, K. Chowdhury, A survey on wireless multimedia sensor networks, *Computer Networks* 51 (4) (2007) 921–960.
- [6] D. Ganesan, B. Krishnamachari, A. Woo, D. Culler, D. Estrin, S. Wicker, Complex Behavior at Scale: An Experimental Study of Low-power Wireless Sensor Networks, Technical Report CS TR 02–0013, UCLA, 2002.
- [7] A. Boukerche, D. Yan, F. Jing, R. Pazzi, A reliable synchronous transport protocol for wireless image sensor networks, in: *IEEE Symposium on Computers and Communications (ISCC)*, Marrakech, Morocco, 2008, pp. 1083–1089.
- [8] T. Le, W. Hu, P. Corke, S. Jha, ERTIP: energy-efficient and reliable transport protocol for data streaming in wireless sensor networks, *Computer Communications* 32 (7–10) (2009) 1154–1171.
- [9] J.-H. Lee, I.-B. Jung, Reliable asynchronous image transfer protocol in wireless multimedia sensor networks, *Sensors* 10 (3) (2010) 1486–1510.
- [10] S. Misra, M. Reisslein, X. Guoliang, A survey of multimedia streaming in wireless sensor networks, *Communications Surveys and Tutorials, IEEE* 10 (4) (2008) 18–39.
- [11] G. Pekhteryev, Z. Sahinoglu, P. Orlik, G. Bhatti, Image transmission over IEEE 802.15.4 and ZigBee networks, in: *IEEE International Symposium on Circuits and Systems (ISCAS)*, Kobe, Japan, 2005, pp. 3539–3542.
- [12] I. Lee, W. Shaw, X. Fan, *Wireless multimedia sensor networks, Guide to wireless sensor networks*, in: S. Misra, I. Woungang, S.C. Misra (Eds.), *Computer Communications and Networks*, Springer, London, 2009, pp. 561–582.
- [13] M. Rahimi, R. Baer, O. Iroezji, J. Garcia, J. Warrior, D. Estrin, M. Srivastava, Cyclops: in situ image sensing and interpretation in

- wireless sensor networks, in: Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems (SenSys), New York, USA, 2005, pp. 192–204.
- [14] D. Xie, T. Yan, D. Ganesan, A. Hanson, Design and implementation of a dual-camera wireless sensor network for object retrieval, in: Proceedings of International Conference on Information Processing in Sensor Networks (IPSN '08), St. Louis, Missouri, USA, 2008, pp. 469–480.
- [15] W. Feng, E. Kaiser, W. Feng, M. Baillif, Panoptes: scalable low-power video sensor networking technologies, ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP) 1 (2) (2005) 151–167.
- [16] P. Kulkarni, D. Ganesan, P. Shenoy, Q. Lu, SensEye: a multi-tier camera sensor network, in: Proceedings of the 13th Annual ACM International Conference on Multimedia, Singapore, 2005, pp. 229–238.
- [17] C.H. Zhiyong, L.Y. Pan, Z. Zeng, M.Q.H. Meng, A novel FPGA-based wireless vision sensor node, in: Proceedings of IEEE International Conference on Automation and Logistics (ICAL), Shenyang, China, 2009, pp. 841–846.
- [18] M. Yoshimura, H. Kawai, T. Iyota, Y. Choi, FPGA-based image processor for sensor nodes in a sensor network, The Open Signal Processing Journal 2 (2009) 7–13.
- [19] Z.Y. Cao, Z.Z. Ji, M.Z. Hu, An image sensor node for wireless sensor networks, in: International Conference on Information Technology: Coding and Computing (ITCC), Las Vegas, Nevada, USA, 2005, pp. 740–745.
- [20] Y. Benezeth, P.M. Jodoin, B. Emile, H. Laurent, C. Rosenberger, Review and evaluation of commonly-implemented background subtraction algorithms, in: Proceedings of the 19th International Conference on Pattern Recognition (ICPR), Florida, USA, 2008, pp. 1–4.
- [21] M. Piccardi, Background subtraction techniques: a review, in: Proceedings of 2004 IEEE International Conference on Systems, Man and Cybernetics, vol. 4, The Hague, The Netherlands, 2004, pp. 3099–3104.
- [22] Xilinx Inc., Xilinx UG130 User manual. <http://www.xilinx.com/support/documentation/boards_and_kits/ug130.pdf>.
- [23] C. Wren, A. Azarbayejani, T. Darrell, A. Pentland, Pfunder: real-time tracking of the human body, in: Proceedings of the Second International Conference on Automatic Face and Gesture Recognition, Killington, Vermont, USA, 1996, pp. 51–56.
- [24] J. Oliveira, A. Printes, R. Freire, E. Melcher, I.S.S. Silva, FPGA architecture for static background subtraction in real time, in: Proceedings of the 19th Annual Symposium on Integrated Circuits and Systems Design, Ouro Preto, Brazil, 2006, pp. 26–31.
- [25] M. Rabbani, R. Joshi, An overview of the JPEG 2000 still image compression standard, Signal Processing: Image Communication 17 (1) (2002) 3–48.
- [26] C. Christopoulos, A. Skodras, T. Ebrahimi, The JPEG2000 still image coding system: an overview, IEEE Transactions on Consumer Electronics 46 (4) (2000) 1103–1127.
- [27] P. McCanny, S. Masud, J. McCanny, Design and implementation of the symmetrically extended 2-D Wavelet Transform, in: Proceedings of 2002 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Orlando, Florida, USA, vol. 3, 2002, pp. III-3108–III-3111.
- [28] S.M. Aziz, D.M. Pham, Efficient parallel architecture for multi-level forward discrete wavelet transform processors, Computers and Electrical Engineering – An International Journal 38 (5) (2012) 1325–1335.
- [29] S.M. Aziz, B. Ramamurthy, Design and FPGA implementation of an efficient 2-D DWT processor, in: IFIP International Conference on Very Large Scale Integration System-on-Chip, Western Australia, Perth, 2005, pp. 498–503.
- [30] S. Guangming, L. Weifeng, Z. Li, L. Fu, An efficient folded architecture for lifting-based discrete wavelet transform, IEEE Transactions on Circuits and Systems II: Express Briefs 56 (4) (2009) 290–294.
- [31] S. Khanfir, M. Jemni, Reconfigurable Hardware Implementations for lifting-based DWT image processing algorithms, in: Proceedings of 2008 International Conference on Embedded Software and Systems, Sichuan, China, 2008, pp. 283–290.
- [32] A. de Baynast, P. Mähönen, M. Petrova, ARQ-based cross-layer optimization for wireless multicarrier transmission on cognitive radio networks, Computer Networks 52 (4) (2008) 778–794.
- [33] P.S. Boluk, S. Baydere, A.E. Harmanci, Robust image transmission over wireless sensor networks, Mobile Networks and Applications 16 (2) (2011) 149–170.
- [34] Digi International Inc., XBEE Multipoint RF Modules. <http://www.digi.com/pdf/ds_xbeemultipointmodules.pdf>.
- [35] Microchip Technology Inc., MRF24J40 Data Sheet IEEE 802.15.4 2.4 GHz RF Transceiver. <<http://ww1.microchip.com/downloads/en/devicedoc/39776c.pdf>>.
- [36] OmniVision. OV7640 Color CMOS VGA Datasheet. <[http://www.comedia.com.cn/OV%20SENSOR%20PB/30W/OV7640_PB%20\(2.1\).pdf](http://www.comedia.com.cn/OV%20SENSOR%20PB/30W/OV7640_PB%20(2.1).pdf)>.
- [37] S.M. Aziz, D.M. Pham, Energy Efficient Image Transmission in Wireless Multimedia Sensor Networks, IEEE Communications Letters 17 (6) (2013) 1084–1087.



Duc Minh Pham received B.S. degree in electronic engineering from HCM National University of Technology, Vietnam in 2003 and M.S. degree in micro systems technology from the University of South Australia in 2008. Mr Pham is currently working as a Research Associate and doing his Ph.D. at University of South Australia in the field of wireless multimedia sensor networks. He has strong industry experience in system design for both FPGA and ASIC platforms. His research interests are in the fields of VLSI implementation of communication systems such as SoC for next generation networking, automation in VLSI design, FEC (Forward Error Correction), APS (Application Specific Processor), sensor networks, coding theory, image and video processing.



Syed Mahfuzul Aziz received Bachelor and Master Degrees in EEE from Bangladesh University of Engineering & Technology (BUET) in 1984 and 1986 respectively. He received a Ph.D. degree in electronic engineering from the University of Kent (UK) in 1993 and a Graduate Certificate in higher education from Queensland University of Technology, Australia in 2002. He was a Professor in BUET until 1999, and led the development of the teaching and research programs in integrated circuit (IC) design in Bangladesh. He joined the University of South Australia in 1999, where he is currently an associate professor. In 1996, he was a visiting scholar at the University of Texas at Austin when he spent time at Crystal Semiconductor Corporation designing advanced CMOS integrated circuits. He has been involved in numerous industry projects in Australia and overseas, and has attracted funding from reputed research organisations such as the Australian Research Council (ARC) and Australian Defence Science and Technology Organisation (DSTO). He has authored over 115 refereed research papers. His research interests include digital system design and testability, high performance processing systems, wireless sensor networks, biomedical engineering and engineering education. Prof Aziz is a senior member of IEEE and a member of Engineers Australia. He has received numerous professional and teaching awards including the Prime Minister's Award for Australian University Teacher of the Year (2009). The journals he has reviewed in recent years include *IEEE Transactions on Computer*, *IEEE Transactions on Image Processing*, *IEEE Transactions on Education*, *IEEE Communications Letters*, *Electronics Letters*, *Computers & Electrical Engineering – An International Journal*, *International Journal of Electronics & Communications*, *Optics & Laser Technology*.