Energy Driven Selection and Hardware Implementation of Bi-Level Image Compression

Muhammad Imran
Mid Sweden University, Sweden
Holmgatan 10
Sundsvall
0046-60-14 84 16
muhammad.imran@miun.se

Mattias O'Nils
Mid Sweden University, Sweden
Holmgatan 10
Sundsvall
0046-60-14 87 80
mattias.onils@miun.se

Najeem Lawal
Mid Sweden University, Sweden
Holmgatan 10
Sundsvall
0046-60-14 85 61
najeem.lawal@miun.se

ABSTRACT
Wireless Vision Sensor Nodes are considered to have smaller resources and are expected to have a longer lifetime based on the available limited energy. A wireless Vision Sensor Node (VSN) is often characterized to consume more energy in communication as compared to processing. The communication energy can be reduced by reducing the amount of transmission data with the help of a suitable compression scheme. This work investigates bi-level compression schemes including G4, G3, JBIG2, Rectangular, GZIP, GZIP_Pack and JPEG-LS on a hardware platform. The investigation results show that GZIP_pack, G4 and JBIG2 schemes are suitable for a hardware implemented VSN. JBIG2 offers up to a 43 percent reduction in overall energy consumption as compared to G4 and GZIP_pack for complex images. However, JBIG2 has higher resource requirement and implementation complexity. The difference in overall energy consumption is smaller for smooth images. Depending on the application requirement, the exclusion of a header can reduce the energy consumption by approximately 1 to 33 percent.

Keywords
Bi-level, compression schemes, wireless vision sensor node.

1. INTRODUCTION
Wireless Vision Sensor Networks (WVSNs) is an emerging field with a number of potential applications including, industrial process monitoring, environmental monitoring, traffic monitoring, surveillance, and smart homes [1][10][24][25]. A WVSN consists of a number of wireless Vision Sensor Nodes (VSNs) which are assumed to be powered by batteries or by means of alternative energy sources. A VSN is expected to monitor its field of view for a long time with limited resources including computational capabilities, memory, power and wireless bandwidth [1]. A VSN can be implemented on software and/or hardware platforms. In this paper, the term software platform represents a micro-controller, while the term hardware platform represents a Field Programmable Gate Array (FPGA). A VSN implemented with the compared to a hardware implemented VSN [3][10]. However, the currently available software platforms, consumes greater energy as design and development time on a software platform is smaller because of the availability of ready to use image processing libraries. On the other hand, a hardware platform, such as FPGA, offers parallel computing while still retaining the programmability of software at a relatively low cost. This makes FPGA a suitable choice for embedded vision processing [3].

For VSN implementation, researchers generally employ two approaches. In the first approach, as shown in Figure 1 (a), no local processing is performed on the VSN and data is transmitted to a server for processing. This strategy has a smaller design complexity and consumes a smaller amount of processing energy. However, this strategy consumes a greater communication energy because the amount of data being transmitted is large [1][10]. In the second approach, depicted in Figure 1 (b), all the required vision tasks are performed on the VSN and only the final features are transmitted to a server for further analysis [24]. This strategy consumes a smaller communication energy because the transmission data is small. However, this strategy consumes a greater amount of processing energy on the currently available software platforms and has a high design complexity on a hardware platform [10].

In comparison to the aforementioned two approaches, a balanced strategy involves dividing the processing load between the VSN and a server [10]. This strategy is shown in Figure 1 (c). The initial front end vision tasks i.e. preprocessing, segmentation are performed on the VSN and the post segmentation tasks including labeling, feature extraction, classification, and recognition are performed on a server. After segmentation, the data is in binary format, thus enabling the use of binary vision tasks, which have reduced latency and hardware resource requirements [3]. The binary data can be transmitted to the server for further processing. This strategy will reduce both the processing energy consumption [31] and the design complexity because the complex tasks are

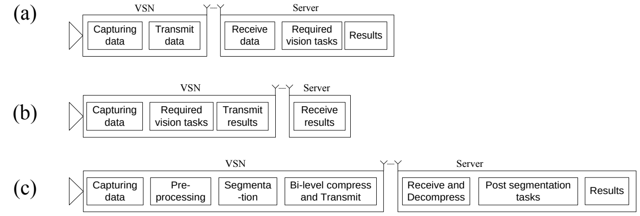

Figure 1. Different VSN architectures. (a) no vision tasks on the node, (b) all required vision tasks on node (c) Some on node and some on server

ICDSC '14, November 04 - 07 2014, Venezia Mestre, Italy

http://dx.doi.org/10.1145/2659021.2659061

moved to a general platform, a server, which has reduced constraints [10]. This strategy will assist in proposing a generic VSN architecture because, in many applications, the preprocessing tasks (tasks before segmentation) are the same and the post segmentation tasks are different [10][37]. In this strategy, binary data needs to be transmitted to server and transmission of raw binary data account for higher communication energy [10][25]. The amount of data can be reduced with suitable compression schemes. This reduction in the amount of data with a suitable compression scheme will reduce the overall energy consumption, which, in turn, will result in a lifetime extension of a VSN [2]. For example, an FPGA based VSN with binary data transmission can have a lifetime of 1.3 years for a sample period of 5 minutes while a VSN with a G4 compression capability can have lifetime of 5.1 years [10]. A detailed discussion of the energy conservation and lifetime maximization can be found in [10][28].

The investigated bi-level compression schemes include G4 [14], G3 [14], JBIG2 [9], Rectangular [19], GZIP [18], GZIP_pack [4][18], and JPEG_LS [17]. Our goal is to investigate a lightweight bi-level compression scheme, which has reduced energy consumption, reduced hardware resource requirements i.e. logics cells and memory, offers a better compression ratio and has a reduced design complexity.

Following the introduction, section II provides a summary of the related work, section III provides an overview of the evaluated compression schemes, section IV describes the selection of the compression methods. Section V covers the implementation on the hardware, section VI presents the results and a discussion and section VII summarizes the conclusions drawn from this work.

2. Related work

Before describing the work related to image compression, some of the sample VSN systems are presented. Kerhet *et al*. [24] proposed a VSN, MicrelEye, for cooperative video processing applications. The images were captured by using a CMOS image sensor with a resolution of 320×240 and the vision tasks were processed on an FPGA and a microcontroller. For frame storage, a Static Random Access Memory (SRAM) was used and for data transmission, a Bluetooth was used. Gasparini *et al*. [25], proposed design principles for VSN in the context of a long-lifetime. The authors used a binary image sensor of 128×64 resolutions for data capturing and a flash based FPGA for processing. Casares *et al*. [26] presented a lightweight and resource efficient foreground object detection and tracking algorithm for VSN applications.

The following paragraphs discuss some of the published research work within the field of image compression. De Carvalho *et al*. [20] analyzed different bi-level image compression techniques, which include G4 based on static Huffman coding, LZW, and JBIG. The authors concluded that a compression scheme based on arithmetic coding, JBIG, provided a better compression ratio. Kolo *et al* [2] presented an adaptive lossless data compression (ALDC) scheme for wireless sensor networks. The proposed compression scheme enabled compression to dynamically adjust to a changing source and compared the proposed compression scheme with other compression schemes such as the Lossless Entropy Compression (LEC) and sensor node LZW (S-LZW). Marcelloni *et al* [27] proposed a simple LEC scheme for wireless sensor networks. The proposed compression scheme uses a small dictionary. The authors compared the results with GZIP, BZIP2, RAR, classical Huffman and arithmetic coding. However, the authors in [2][27] only considered wireless sensor networks with scalar data.

Soro *et al*. [1] analyzed different colour /greyscale compression techniques for WVSN applications, operating in resource constrained requirements. Lin *et al*. [17] proposed an architecture for JPEG-LS, considering the WVSN applications and compared its compression efficiency for high and low correlated data. Lee *et al*. [21] explored the energy tradeoffs involved in JPEG compression for energy constrained platforms. It is observed that JPEG offers lossy compression and has a wide range of tradeoffs as the compression ratio is related to the energy consumption during compression and transmission [21]. This means that for each application, the requirement is to perform independent analysis. In addition to this, JPEG is generally used for colour and greyscale compression and is computationally intensive. The authors in [22] proposed a technique in which changes in the image are detected in order to locate a Region Of Interest (ROI) and then a fast-Discrete Cosine Transform (DCT) is used to compress the ROI data for the WVSN.

By investigating the published work, it can be observed that researchers have focused on particular aspects of some specific compression schemes. In this work, we have investigated the energy consumption, hardware resource requirements and compression efficiency for seven potential bi-level compression schemes for a hardware implemented VSN with two different data rate transceivers. The term compression efficiency is referred to as the compression ratio in this work. In addition to this, the effect of removing the header information on energy consumption is also investigated.

3. COMPRESSION SCHEMES AND PLATFORMS

Compression can be categorized into lossy and lossless schemes. In lossy compression methods, the image quality is related to the energy consumed during compression and transmission. For a required image quality, calculations are required by using an algorithm, which can determine the best set of compression techniques [30]. This is particularly in-efficient for a WVSN scenario in which a VSN is expected to be deployed for a number of applications with fixed VSN's parameters, i.e., optics, resolution, lighting and distance from field of view. For each application, the compression quality is required to be analyzed in order to ensure the pixel accuracy of the systems. This situation is critical for machine vision [32] because a system is required to detect a specific size of object.

In comparison to lossy compression schemes, lossless compression offers no losses in the data and the exact contents can be reconstructed on the receiver side, which reduces the tradeoffs among computation, compression efficiency and energy consumption [30]. This assists in proposing generic solutions for machine vision applications in which the accuracy of the system is critical. For example, different system parameters such as image resolution, focal length, distance to object and lighting are calculated for one application in order to detect a 1 millimetre (mm) object. The lossless compression would offer an accuracy of 1 mm pixels for different types of applications because lossless compression scheme offers no degradation in the image quality with respect to a change in the system setup such as image resolution, focal length and optics, provided the pre-processing tasks can correctly segment the data. In systems with lossy compression schemes, pixels' accuracy is required to be ensured for every application, which will require changing the system parameters.

In this paper, the bi-level compression schemes, ITU-T G3, ITU-T G4, JBIG2 [5], JPEG_LS [35], Rectangular, GZIP [34] and GZIP_pack [34] have been investigated. All bi-level compression schemes are lossless in nature except for JBIG2, which supports both lossy and lossless image compression [29]. JPEG_LS is not a specifically made for bi-level image but it is selected because of the claim regarding its suitability for resource constrained platforms [17]. The lossless compression schemes reduce the data redundancy in two steps. In the first step, the spatial redundancy is removed with the help of run length encoding, predictive techniques, transform techniques or other decorrelation techniques. In the next step, entropy coding is performed to remove the coding redundancy with the assistance of arithmetic coding, Huffman coding and LZW [33]. Following this, the processing platforms, and energy measurements are discussed.

3.1 Processing platform and energy measurements

A hardware implemented VSN will have an FPGA platform together with a micro-controller, radio transceiver, SRAM and FLASH memories [10][24][25]. In this work, we have evaluated compression schemes on a processing platform and communication modules which are often used for VSN systems [3][25][31]. The schematic of the processing platform and the communication module is shown in Figure 2.

For investigation of the compression schemes on a hardware platform, a Xilinx Spartan6 XC6LX25FPGA [6] has used together with a SENTIO32 [10]. The selected Spartan 6 FPGA, has a 229 Kbits distributed Random Access Memory (RAM) and has 936 Kbits (52×18Kbits) of block RAMs. The Spartan 6 FPGA has been selected in this work because it has sufficient hardware resources to implement different compression schemes and also has the availability of support [7]. The FPGA power is calculated by using a Xilinx power measuring tool, XPower Estimator and XPower Analyzer [7]. The execution time, for the compression schemes on the hardware platform is calculated by using Eq. 1.

$$T = (R \times (C + Ls) + Lt) / f \qquad \text{(sec)} \qquad (1)$$

where Lt is the latency, f is frequency, R represent rows, C represents columns and Ls represents low line sync.

For communication purposes, we have used a SENTIO32 platform which has been developed at Mid Sweden University for wireless sensor network applications. The platform has an integrated AVR32 micro-controller, AT32UC3B0256 [15] and CC2520 RF transceiver [16]. The AVR32 has a 32bit RISC MCU, with a 256KB FLASH and a 32KB SRAM.

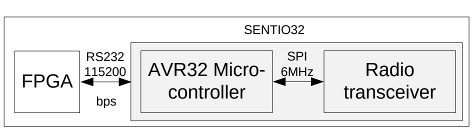

Figure 2. Hardware platform for a VSN. FPGA together with SENTIO32.

The CC2520 RF transceiver transmits data by using the IEEE 802.15.4 protocol (Zigbee), with a data rate of 250 Kbps. The power consumption and the transmission time of the data transmission on the SENTIO32 platform is measured with respect to each compression scheme. The radio transceiver, based on the IEEE 802.15.4 protocol, has smaller data rates, therefore, in order to analyze the effect on the energy consumption with a greater data rate transceiver, we have employed an RN-171[13] transceiver, which is based on the IEEE 802.11 protocol (WiFi). The RN-171 has a data rate of 2 Mbps. This will assist in determining a compression scheme that is suitable for different types of transmission scenarios. For calculating the transmission time T_IEEE_802.11 in Eq.2 with respect to the IEEE 802.11 compliant transceiver, we have used a transmission model [12].

$$\text{T\_802.11} = 192 + \frac{288 + (X \times 8)}{R} \qquad (\mu\text{sec}) \qquad (2)$$

where 192 μsec is related to the long preamble type and 288 bits are the header overhead; X is the data in bytes and R is the data rate. The RN-171 [13] has a typical power consumption of 379 mW. The wakeup time of the IEEE 802.11 compliant transceiver is high but, with the advancements in technology, the gap is reducing to the level of the IEEE 802.15.4. Thus, it has enabled battery operated applications to use WiFi [11]. It is worth mentioning that the selection of the transceiver depends on the specific requirements of the application but here, for proof of concept, we have selected one with a small data rate and another with greater data rates.

3.2 Compression analysis

Khursheed *et al.* [4] have investigated the bi-level compression schemes on a personal computer (PC). However, the FPGA based embedded platform has different constraints and requirements as compared to a PC. Therefore, in this work, we have investigated the processing energy, communication energy and compression efficiency with consideration to embedded platforms. In relation to this, input image datasets include 160 images produced by the image generation model and 88 images produced by four real applications, including particle detection [10], remote meter reading [31], bird detection [31] and people counting [25]. The images for the two applications, people counting and bird detection, have been captured in an outside environment. These two applications represent surveillance applications. The image data sets include greater variation in the contents. The images for the two applications, meter reading and particle detection, have been captured indoors and the images have smaller variations in their content. These two applications represent machine vision systems which have an integrated lighting module such as an LED ring in order to achieve a greater signal to noise ratio.

For creating an image data set which characterizes different objects, a statistical model has been developed. This model is now discussed.

3.2.1 Image generation model

Images in WVSN applications, in which data is reduced to binary after segmentation, contain distinctive objects from the background. These objects are present in the images in random numbers, locations, shapes and sizes [1] [10]. Therefore, to test the seven compression schemes with all of these scenarios on a real set of images, for different applications, is a challenging and time consuming task. To represent different scenarios, we have developed a statistical model to generate images with a resolution of 640×400. It is important to mention that this statistical model has been complemented with actual image data sets for four real aforementioned applications. In a binary image, each pixel can have a maximum of two colours, which makes the characterization of the objects an easy task as compared to grayscale and coloured objects. The binary objects, after segmentation, can be characterized by means of shapes such as circles, semi-circles, quarters of circles, ellipses, semi-ellipses, and quarters of ellipses, rectangles and curves [4]. These shapes have been used in generations of statistical models. In the statistical model, we have handled two cases regarding the placement of objects in the images.

In the first case, we have increased the number of objects while in the second case, we have increased the object size while retaining a constant number of objects. In each case, we have 8 sets, in which each set has shapes namely circles, semi-circles, quarters of circles, ellipses, semi-ellipses, and quarters of ellipses, rectangles and curves. For each shape feature, we have 10 images and there is a 10 percent increase in object size and number of objects in each successive image. Thus, there are a total of 160 images, which include objects with various features such as different sizes, random locations, different shapes and different numbers of objects. The sample images for the increasing number of objects are shown in Figure 3 and images for real applications are shown in Figure 4. The complete set of sample images can be downloaded for experimentation [23]. Using these image data sets, our goal is to investigate compression schemes which offer better compression in the presence of varying image contents.

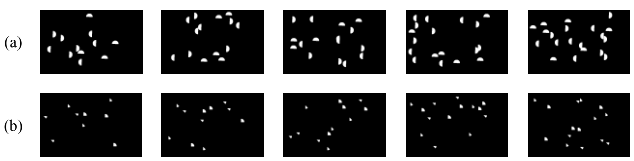

Figure 3. Sample images for randomly placed increased number of objects. (a) semi ellipses. (b) quarter circles.

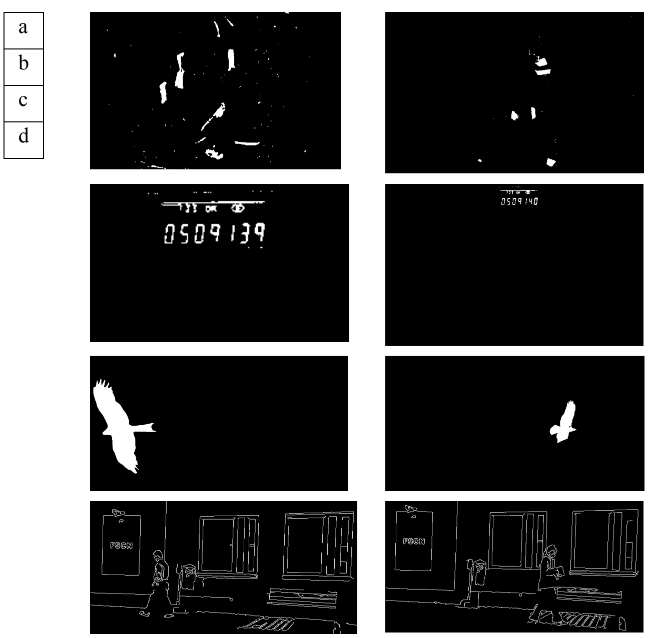

Figure 4. Different real applications' sample images used for compression. (a) particle detection (b) remote meter reading. (c) bird detection. (d) people counting.

Table 1. The average and standard deviation of output data of compression schemes

Compression schemes	Average output data (bytes)	SD of data
G4	455	166
G3	2196	254
JBIG2	302	79
Rectangular	703	478
GZIP	2123	311
GZIP_pack	598	276
JPEG-LS	745	288

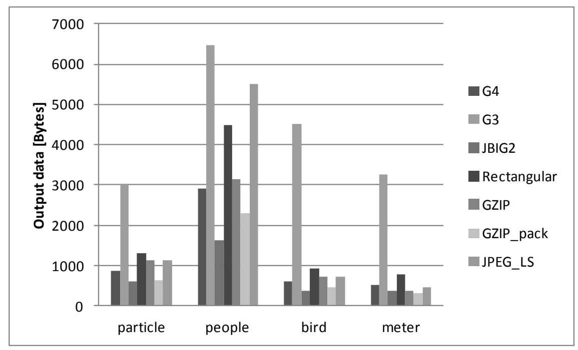

Figure 5. Average output compressed data of seven compression schemes with respect to different applications.

4. SELECTION OF CANDIDATES

For hardware implemented VSN, a suitable compression scheme must be selected which has smaller processing energy, implementation complexity and resource requirement but, still offers a greater compression ratio so as to reduce the communication energy. In this work, the processing energy and compression efficiency of the compression schemes with respect to the same input data sets have been considered as a deciding factor for selecting compression schemes for a hardware implemented VSN. For JBIG2, JPEG_LS, GZIP, existing published designs were used whereas for G3, G4 and rectangular, RTL level designs were developed. For the compression efficiency test, we first performed compression on the input synthetic images. The average output data produced for the 160 sample images, in Table 1 shows that the G4, JBIG2 and GZIP_pack produced small amounts of data as compared to other compression schemes. The standard deviation (SD) of the output data shows that JBIG2 has a small standard deviation for changes in the image contents, which is then followed by G4.

The investigated compression schemes were then tested on real images for four applications involving bird detection, remote meter reading, particle detection and people counting. The input images have a resolution of 640×400 for particle detection and remote meter reading, 1020×768 for bird detection and 640×320 for people counting. Table 1 and Figure 5 show that G4, GZIP_pack and JBIG2 have greater compression efficiencies for the same set of input images as compared to the other compression schemes. Table 2 shows that G3, G4, GZIP, GZIP_pack and Rectangular have smaller processing energy as

Table 2. Average power consumption, processing time and energy of different compression schemes.

Compression schemes	Average power (mW)	Processing time (msec) (640×400)	Energy (mJ)
G4	31	9.98	0.31
G3	30	10.00	0.30
JBIG2	56	10.03	0.56
Rectangular	32	9.98	0.32
GZIP	36	9.98	0.36
GZIP_pack	36	9.98	0.36
JPEG-LS	46	10.15	0.47

compared to JBIG2. However, compression efficiency must be considered as it is related to the amount of data being produced for transmission purposes and the output data is associated with the communication energy. It must be noted that processing energy in this paper includes only FPGA power. Depending on the implementation, microcontroller and transceiver power can be added for actual implementation. Table 1, Table 2 and Figure 5, show that the compressions schemes G4, GZIP_pack and JBIG2 have a smaller difference regarding their processing energies with respect to other compression schemes. However, the difference between the compression efficiency is greater. Therefore, compression schemes including G4, GZIP_pack and JBIG2 have been selected for further analysis. The selected candidates are now investigated on a hardware platform.

5. IMPLEMENTATION ANALYSIS ON HARDWARE

The compression schemes, GZIP_pack, G4 and JBIG2 are investigated in relation to their implementation on a hardware platform.

5.1 GZIP_pack

GZIP is a lossless compression scheme that uses Lempel-Ziv77 (LZ77), and Huffman coding. The GZIP_pack is similar to GZIP, apart from the fact that the input binary images are first packed in byte format. In GZIP, the input data is encoded by using an LZ77 to remove the redundancy in the data and then the Huffman coding is used to remove the coding redundancy [18]. The resources utilizations and power consumption of the GZIP_pack are presented in Table 3 for the FPGA device mentioned in sub-section 3-3.1. The power consumption is calculated by using an Xpower estimator [7] for a design implemented on an Xilinx XCV 400 FPGA [34].

Table 3. GZIP resources utilizations and power consumption.

Number of LUTs	Number of FFs	Number of BRAMS	Power (mW)
2308	1536	16	36

5.2 G4 compression method

G4 is a lossless compression scheme, which encapsulates data in a TIFF file format. In G4, the coding scheme uses a two-dimensional line-by-line coding in which the position of each changing picture element, rather than alternating white and black runs in each scan line, is considered. In this manner, the vertical features in the source image can be used to achieve better compression ratios [14]. G4, implemented in this work, includes two stages. In stage1, the three encoding modes, the pass mode, vertical mode and horizontal mode, are identified. In addition to this, black and white runs for the horizontal mode are calculated at this stage. In the implementation, three line buffers are used at stage1 in order to have information for the coding and reference lines. Two of these buffers are used for storage of the reference line and the coding line while the third line buffer is used for saving the current row data. When the data is being saved, it then becomes the coding line and the previous coding line becomes the reference line. The previous reference line becomes the saving line for the next row and is overwritten by the new pixel data.

In stage2, the pass mode, vertical mode and horizontal mode are assigned Huffman codes. The Huffman codes of the pass and vertical modes are assigned at runtime. The Huffman codes of the horizontal mode are stored in the on-chip memory of the platform. The resource utilization of G4 is given in Table 4.

Table 4. G4 logic and memory utilization.

Number of LUTs	Number of FFs	Number of BRAMS	Power (mW)
3466	389	3	31

5.3 JBIG2 compression method

JBIG2 is an international standard that supports both lossless and lossy compression of binary images as compared to its predecessor, the JB1G1, which only possessed lossless compression capability. The design goal for JBIG2 was to provide lossy compression with almost no visible degradation of quality and to provide a higher compression ratio as compared to other lossless compression schemes [36]. It offers better compression ratios as compared to the existing lossless compression schemes [4][29][36]. A typical JBIG2 encoder decomposes the input bi-level image into several regions and codes these regions separately using a different coding method. JBIG2 defines the requirements of a compliant bit stream and defines the decoder behavior. The encoder design is not explicitly defined, therefore, depending on the design, JBIG2 encoders will have varying complexity, speed and different compression performances [9].

Following this, the resource utilization and power consumption of JBIG2 is investigated on a hardware platform. In [5], a hardware architecture of JBIG2 is proposed and the hardware resource utilization is given in gate counts. Gate count metrics are used for ASIC underlying architectures whereas, in FPGA, the resource utilization is given in logic cells; however, gate counts information can be used for pre-implementation architecture evaluation on an FPGA. This requires the gate counts to be converted into Configurable Logic cells. In [8], an attempt is made to compare the FPGA and ASIC in terms of logic density,

Table 5. JBIG2 logic and memory utilization (* 9 Kbits,18 Kbits)**

Modul	# of Logic gates	# of logic cells	# of LUTs	# of FFs	# of RAMS	PWR (mW)
Stage 1	1496	99	2553	5106	3*	56
Stage 2	735	49	1254	2509	2**	
Stage 3	3094	206	5280	10561		
Total	5325	354	9087	18176		
Available	N.A.	24051	15032	3575	104* 52**	

circuit speed and power consumption. It should be noted that such comparisons are influenced by the benchmarks being used. However, a system architect can use the comparison results to assess the feasibility in relation to an FPGA implementation. In current Xilinx technology, the typical number of gates per slice is approximately 15 and each slice has four Look Up Tables (LUTs) and eight flip- flops [6] in the Spartan 6 FPGA. This information can be used to assess the JBIG2 implementation on an FPGA platform. In this work, we have used a design [5] in order to estimate the JBIG2 resource requirements by using the Xilinx tool called XPower estimator [7] for Spartan 6 FPGA. The memory requirement of JBIG2 is estimated in Table 5 for the JBIG2 specifications of profile5 [9]. The resource utilization and power consumption of JBIG2 are shown in Table 5. Following this, the results are presented.

6. RESULTS AND DISCUSSION

In this section, the processing and communication energy with two different data rate radio transceivers IEEE 802.15.4 and IEEE 802.11 are investigated for selected compression schemes including G4, JBIG2 and GZIP_Pack.

6.1 Energy consumption, resource utilization and complexity

Figure 6 and Table 2 show that JBIG2 has a greater processing energy by up to a factor of 2 as compared to the G4 and the GZIP_pack. For compression schemes, it is important to consider its capability regarding data reduction, which is related to the communication energy. Therefore, at this point, we will discuss the overall energy consumption of the three compressions.

Figure 6 shows that for images captured with external light source such as LEDs, the difference among the overall energy consumption with respect to the three compression schemes was small. It is worth mentioning that the meter reading and particle detection represents machine vision applications, which require LED lighting in order to achieve a high signal to noise ratio and that the variation in content is small. Figure 6 shows that with header data, JBIG2 has a 1 up to 27 percent smaller energy consumption as compared to that for the GZIP_pack and G4.

Figure 6 shows that with header data, the difference among the JBIG2, GZIP_pack and G4 for communication energy is greater for two applications, people counting and bird detection. JBIG2 offers a smaller overall energy consumption by 3 up to 43 percent as compared to the G4 and GZIP_pack. This energy reduction trend is shown for both ZIGBEE and WIFI transceivers in Figure 6. In these two applications, the images were taken with ambient lighting conditions and the contents were relatively complex and random as compared to that for the meter reading and particle detection applications images.

6.2 Exclusion of header

Each of the compression schemes has data overhead in terms of a header, which will account for the extra communication energy consumption. In some applications, a header is required to be embedded on the node side. In applications where the sender and receiver have the knowledge of the compression scheme, the header can be discarded and can be added on the receiving side. This will save communication energy. In relation to communication energy, the energy on the IEEE802.15.4 was measured on real hardware while the energy on IEEE 802.11 was estimated from a model [12].

The GZIP_pack, a variant of GZIP, consists of a 10 bytes header, the payload and 8 bytes footer. The header contains a magic number, a version number and a time stamp. The payload is the compressed data. Thus, the header and footer information of the GZIP is 18 bytes (excluding the unnecessary extra header information).

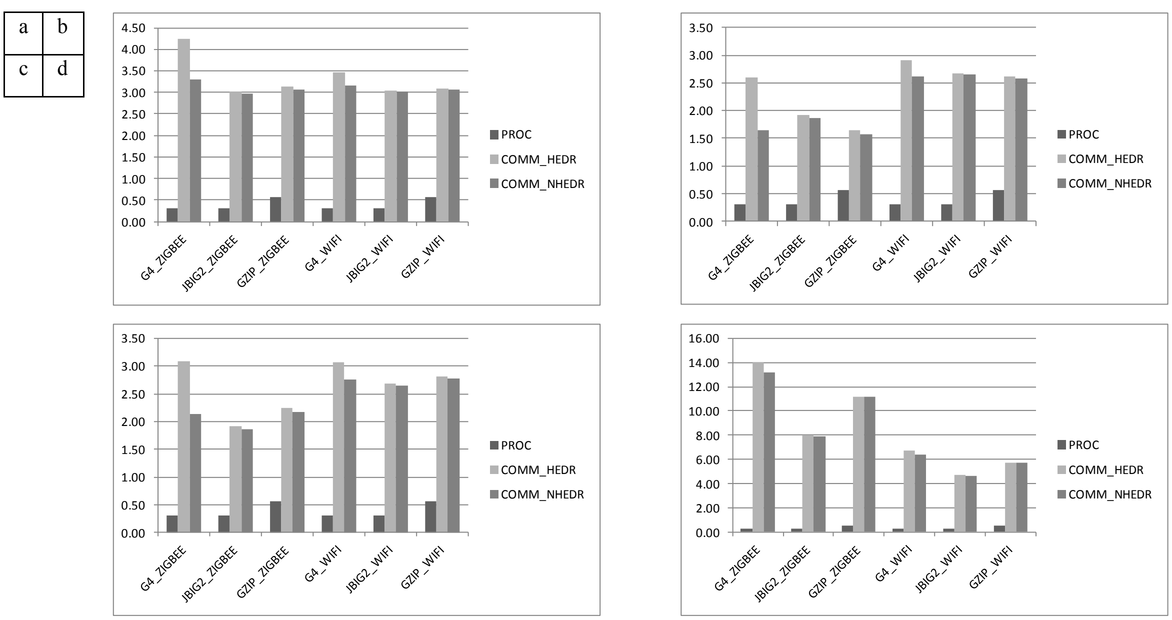

Figure 6. Processing (PROC) and communication energy consumption with header (COMM_HEDR) and without header (COMM_NHEDR) stripes for four test cases. (a) Particle detection. (b) Meter reading. (c) Bird detection. (d) People counting.

TIFF file format has192 bytes for Group 4 in our implementation. The header of JBIG2 is 13 bytes long. Depending on the radio transceiver, the exclusion of the header can reduce the energy consumption by as much as 4 up to 33 percent for the G4 compression scheme. The JBIG2 and the GZIP_pack have less than a 3 percent energy reduction with the exclusion of the header because of the smaller header requirements for the two compression schemes.

The conclusion based on the aforementioned aspects is that JBIG2 offers a small overall energy consumption for complex images with varying content, taken in an outdoor environment.

However, the difference in overall energy consumption of JBIG2 with respect to G4 and GZIP_pack for images with smaller changing content, taken in the presence of an external light, is small. Consideration must be given to the fact that the JBIG2 has greater resources and has a higher implementation complexity as compared to the G4 and GZIP_pack. The GZIP_pack has a higher memory requirement and has a larger standard deviation in the output compressed data as compared to that for both JBIG2 and G4.

Depending on the application, any of the three compression schemes GZIP_pack, G4 and JBIG2 can be used for a hardware implemented VSN. Nonetheless, the smaller processing energy, reasonable resource requirements in terms of logics cells, smaller memory requirements and smaller design complexity makes G4 a suitable candidate.

7. CONCLUSION

This paper investigates seven bi-level image compression schemes inclduing G4, G3, JBIG2, Rectangular, GZIP, GZIP_pack and JPEG_LS, for a resource constrained wireless vision sensor node, implemented on a FPGA based hardware platform. The goal is to select a lightweight compression scheme which has a smaller design complexity, consumes a smaller amount of energy and requires reduced hardware resources. To achieve this goal, the processing and communication energy consumption of the aforementioned compression schemes are investigated on a hardware platform together with two different data rate transceivers IEEE 802.15.4 and IEEE 802.11.

From the seven, the three compression schemes GZIP_pack, G4 and JBIG2 have reduced energy requirements on a hardware platform. For complex images, the overall energy consumption of JBIG2 is small as compared to that for G4 and GZIP_pack. For images with smaller changing content, the difference between the overall energy consumption is small among the three compression techniques. JBIG2 requires greater resources in terms of logics and has a greater implementation complexity whereas GZIP_pack requires approximately 5 times greater memory as compared to JBIG2 and G4. GZIP_pack has greater variation in output data for the same set of input data as compared to G4 and JBIG2. It is further concluded that the exclusion of the header data can conserve energy by approximately 33 percent.

Depending on the requirement, any of the three compression scheme can be selected for bi-level compression in vision sensor node. However, the smaller complexity and smaller standard deviation for the output with respect to different image content and smaller resource requirements makes G4 a suitable candidate for a hardware implemented vision sensor node. The compression scheme will assist in reducing the overall energy consumption of the currently available sensor platforms and thus increasing the lifetime of the sensor nodes.

8. REFERENCES

[1] Soro, S., and Heinzelman, W.: "A Survey of Visual Sensor Networks. Advances in multimedia, pp. 1-21, 2009.

[2] Kolo, J. G., Shanmugam, S. A., Lim, D. W. Gin., Ang, L., and Seng, K. P.: "An Adaptive Lossless Data Compression Scheme for Wireless Sensor Networks", Journal of Sensors, Sept. 2012

[3] Bailey, D. G.: "Design for Embedded Image Processing on FPGAs", John Wiley & Sons, Asia, 199-231, 2011.

[4] Khursheed, K., Imran, M., Ahmad, N., O'Nils, M.: "Selection of bilevel image compression methods for reduction of communication energy in wireless visual sensor networks", Proc. of SPIE, Real-Time Image and Video Processing, Belgium, April, 2012.

[5] Chun-Chia, C., Yu-Wei, C., Hung-Chi, F., Liang-Gee, C.: "Analysis and architecture for memory efficient JBIG2 arithmetic encoder", 48th Midwest Symposium on Circuits and Systems, pp.1191-1194, 2005.

[6] Xilinx, Product Selection Guides (2013), Available online: http://www.xilinx.com/publications/matrix/Product_Selection_Guide.pdf.

[7] Xilinx, XPower Estimator User Guide, (2012), Xilinx power tools tutorial (2010). Available: http://www.xilinx.com

[8] Kuon, I., Rose, J.: "Measuring the gap between fpgas and asics". Proc. of the 14th intl. symposium on Field programmable gate arrays, USA, pp.21-30, 2006.

[9] "Application profiles for recommendation T.88 - lossy/lossless coding of bi-level images (JBIG2) for facsimile.", ITU-T Recommendation T.89, Sept, 2001.

[10] Imran, M., Khursheed, K., Lawal, N., O'Nils, M., Ahmad, N., "Implementation of Wireless Vision Sensor Node for Characterization of Particles in Fluids", IEEE Trans. on Circuits and Systems for Video Technology, pp. 1634 – 1643, 2012.

[11] Tozlu, S., "Feasibility of Wi-Fi enabled sensors for internet of things", Proc. of the 7th IEEE Intl. Wireless Communications and Mobile Computing Conference, 2011.

[12] Campoccia, F., Sanseverino, E. R., Zizzo,G.: "Analysis of the Limitations of WiFi Communications Managed by the IEEE 802.11 Protocol in Data Transmission in Automated Power Distribution Systems", Speedam, 2010.

[13] RN-171, Available: http://www.rovingnetworks.com/, 2012.

[14] TIFF, Revision 6.0, Available: http://www.itu.int/itudoc/itu-t/com16/tiff-fx/docs/tiff6.pdf, 1992.

[15] AT32UC3B0256,AVR32,Available:http://www.atmel.com/

[16] CC2520 transceiver, Available: http://www.ti.com/

[17] Lin, C. C., Chuang, C. C., Chiang, C. W., Chang, R. I.: "A Novel Data Compression Method using Improved JPEG-LS in Wireless Sensor Networks," IEEE Intl. Conf. on Advanced Communication Technology, pp. 346-351, 2010.

[18] Rigler, S., Bishop, W., Kennings, A.: "FPGA-Based lossless data compression using Huffman and LZ77 algorithms," Canadian Conf. on Elec. and Com. Eng., pp.1235-1238, 2007.

[19] Mohamed, S. A., Fahmy, M. M.: "Binary Image Compression Using Efficient Partitioning into Rectangular

Regions," IEEE Trans. on Comm., vol. 43, no. 5, May, 1995.
[20] De Carvalho, T. B. A., Tenorio, D. J., Ren, T. I., Cavalcanti, G. D. C., Jyh, T. I.: "Neighborhood coding for bilevel image compression and shape recognition", IEEE Intl. Conf. on Acoustics Speech and Signal Processing, pp. 1302–1305, 2010.
[21] Lee, D. U., Kim, H., Rahimi, M., Estrin, Villasenor, D. J. D.: "Energy-efficient image compression for resource-constrained platforms", IEEE Trans. Image Process., vol. 18, no. 9, pp. 2100 -2113, 2009.
[22] Xiong, Z., Fan, X., Liu, S., Zhong, Z.: "Low complexity image compression for wireless multimedia sensor networks", Intl. Conf. on Inf. Science and Tech., pp. 665 – 670, 2011.
[23] Sample images, Available: http://apachepersonal.miun.se/~muhimr/index_files/sample_images.rar.
[24] Kerhet, A., Magno, M., Leonardi, F., Boni, A., Benini, L.: "A low-power wireless video sensor node for distributed object detection", Journal of real-time image processing, vol.2, pp. 331-342, Oct. 2007.
[25] Gasparini, L., Manduchi, R., Gottardi, M., Petri, D.: "An Ultralow-Power Wireless Camera Node: Development and Performance Analysis", IEEE Trans. on Instr. and Meas.,vol 60, pp 3824 - 3832, Dec. 2011.
[26] Casares, M., Velipasalar, S.: "Adaptive Methodologies for Energy-Efficient Object Detection and Tracking With Battery-Powered Embedded Smart Cameras", IEEE Transactions on Circuits and Systems for Video Technology, pp. 1438 – 1452, Oct 2011.
[27] Marcelloni, F. A., Vecchio, M.: "An Efficient Lossless Compression Algorithm for Tiny Nodes of Monitoring Wireless Sensor Networks", The comp. journal, pp. 969-987, Nov 2009.
[28] Anastasi, G., Conti, M., Di Francesco, M., Passarella, A.: "Energy conservation in wireless sensor networks A survey", Ad Hoc Networks 7(3), 537–568, May 2009.
[29] P. Howard , F. Kossentini , B. Martins , S. Forchhammer , W. Rucklidge and F. Ono "The Emerging JBIG2 Standard", IEEE Trans. on Circuit and Systems for Video Technology, vol. 8, no. 5, pp.838 -848, 1998.
[30] Lee, D. U., Kim, H., Rahimi, M., Estrin, D. and Villasenor, D., "Energy efficient image compression for resource-constrained platforms", IEEE Trans. Image Process., 18(9), 2100–2113 (2009).
[31] Imran, M., Ahmed, N., Khursheed, K., Waheed, M. A., Lawal, N. and O'Nils, M. "Implementation of Wireless Vision Sensor Node with a lightweight bi-level Video Coding", IEEE Journal on Emerging and Selected Topics in Circuits and Systems, vol. 3, pp. 198-209, June, 2013.
[32] Parsonage, K. D., Petrell, R. J., Accuracy of a machine-vision pellet detection system, Aquacultural Engineering, vol. 29, pp.109-123, Dec. 2003.
[33] Yang, M., Bourbakis, N.: "An Overview of Lossless Digital Image Compression Techniques," 48th IEEE Midwest Symposium on Circuits & Systems,vol. 2, pp. 1099-1102, Aug, 2005.
[34] Akil, M., Perroton, L., Grandpierre, T., FPGA based architecture for hardware compression/decompression of wide format images, Journal of Real-Time Image Processing, vol 1, pp. 163-170, Dec. 2006.
[35] JPEGLS encoder, CAST Ins., Available online, (2010) http://www.cast-inc.com/ip-cores/images/jpegls-e/cast_jpegls-e-x.pdf
[36] Ono, F., Rucklidge, W., Arps, R., Constantinescu, C., "JBIG2-the ultimate bi-level image coding standard," Image Processing, 2000. Proceedings. 2000 International Conference on , vol.1, pp.140-143, 2000.
[37] Imran, M., Benkrid, K., Khursheed, K., Ahmad, N., Waheed, M. A., O'Nils, M., and Lawal, N.,"Analysis and characterization of embedded vision systems for taxonomy formulation", SPIE Real-Time Image and Video Processing, USA, Feb. 2013.