Energy Efficiency of Image Transmission in Embedded Linux based Wireless Visual Sensor Network

Nyoman Putra Sastra, Wirawan, and Gamantyo Hendrantoro

Original scientific paper

Abstract—**Wireless Visual Sensor Network (WVSN) is a system that consists of visual sensor nodes with an embedded processor. WVSN devices have limited resources of energy, computation capability, memory, and bandwidth. Due to these limitations the implementation of WVSN for large multimedia data, such as images, become a challenging task. Therefore, it is required compressed images prior to transmission. In addition to the limited resources, the system implementation strongly affects the efficiency of the working system.**

The main contribution of this research is to offer a technical solution of simpler image compression on the WVSN platform. JPEG 2000 is investigated as an alternative compression method to reduce the size of data transfer on WVSN using Embedded Linux as its operating system. Compressed images are transferred to a receiver on communication of IEEE 802.15.4.. This paper shows that the energy consumption for compression and transmission will reduce to only 10.48%, 13.60%, and 17.11% compared to raw image. BER will significantly reduce by implementing image compression. Therefore, it is demonstrated that this model significantly increases energy efficiency, memory utilization efficiency, and data transfer time with acceptable PSNR, compared to uncompressed images.

Index Terms—**Wireless Sensor Network, Distributed Computation, Energy Efficiency, Image Compression, Image Transmission.**

I. Introduction

Wireless Visual Sensor Network (WVSN) is a platform in a communication system that consists of small visual sensor nodes in the form of a camera with an embedded processor and the ability to communicate wirelessly [1-4, 22-23]. The main difference of WVSN from other sensor networks is its ability to transfer visual data such as images or videos. WVSN is capable of sending multimedia data with relatively larger size than scalar data. Generally, WVSN uses batteries to power the device, hence it has limited lifetime for its sensing, processing, and data transmission operations. WVSN also has the capabilities of processing visual data within sensor nodes, and of transmitting real-time data.

Selecting a platform for the WVSN operating system is a critical step. Operating Systems such as TinyOS and SOS have higher stage or step time costs when compiling and flashing, since they repeat the same steps for the implementation of every new method, code, or application [5]. The re-flashing process requirement that leads to shorter board lifetime, and inflexibility in regard to application modification. The use of Embedded Linux eases the task and lengthens board lifetime [6] as re-flashing is no longer required and new applications can be applied while the system is running. New methods or applications that will be applied to the Embedded Linux WVSN needs to be compiled on Linux computer host by the compiler that corresponds to the WVSN processor [7].

The limited resources on WVSN, such as energy, computational capability, bandwidth and memory become constrained in visual applications. Furthermore, the main limitation for WVSN is energy, and generally, energy consumption for transmission is greater than that for computation [8]. To lower energy consumption for data transmission, the system has to cut transmission time without affecting image quality. In order to reduce transmission time, smaller data size is needed. One method to reduce data size is the use of image compression techniques with optimal compression parameters for efficient use of energy, memory, and bandwidth while maintaining image quality.

Investigations of JPEG 2000 image compression, focusing on energy consumption, memory utilization, and computational complexity, have been reported in [9-11]. In [9], the research focuses on the effect of energy consumption and PSNR when both wavelet subband parameters are altered in resolution scale and wavelets coefficients level. This research was not carried out on an Embedded Linux platform and the results do not show correlation between image compression and transmission time of images. The authors of [10] investigated the method of energy efficiency of wavelets integrated with Embedded Block Coding with Optimized Truncation (EBCOT). Comparison between compression ratio and quantization level was made based on observed PSNR values. The research was not specifically conducted in a WVSN environment and was not concerned with WVSN characteristics such as computation, bandwidth and memory constraint.

JPEG 2000 is a more robust compression method for dealing with error transmission and error decoding, with scalability and degradation tolerance [13], than JPEG. This robustness comes from wavelet transformation. The downside of JPEG 2000 is

Manuscript received June 21 2015; revised September 28 2015.

N. P. Sastra1 and Wirawan are with the Department of Electrical Engineering, Universitas Udayana, Bali, 80114, Indonesia.

G. Hendrantoro is with the Department of Electrical Engineering, Institut Teknologi Sepuluh Nopember, Surabaya, 60111, Indonesia. (E-mails: putra.sastra@unud.ac.id, {wirawan, gamantyo}@ee.its.ac.id).

high memory utilization and high-energy consumption. High memory is use caused by processes that include the whole frame. In addition, high-energy consumption results from complex computation requirements.

Meanwhile, in [13], research of image transmission on WVSN is focused on Low-MAC FEC optimization for IEEE 805.15.4 standards. This reference notes that it needs to integrate Low-MAC FEC and RCPC coder at sensor node LR-WPAN to send images. References [13-15,21] are investigation of compression method on WVSN using JPEG. Particularly in [15], experiments were carried out using TinyOS to find relations between quality factors, voltage drop and PSNR in JPEG.

From the above discussion, there are three important issues affecting the image transmission process in WVSN [9-16], i.e. (1) the use of embedded operating system for image compression implementation, (2) Image compression enhancement, and (3) efficient resource usage. However those methods are implemented mostly on TinyOS that has many limitations, such as the re-flashing process requirement that leads to shorter board lifetime and these limitations will be detailed later in section 2.

The use of JPEG 2000 as the image compression method in this research is aimed at finding optimum parameters of tile size, compression level, bitrate per pixel, block code and DWT level. The present research focuses on optimization of JPEG 2000 when it is used as an image compression method on WVSN with the Intel XScale PXA271 processor and Embedded Linux OS, since JPEG 2000 optimization with Embedded Linux for WVSN has not been explored yet. The advantage of the implementation of Linux rather than other operating systems such as SOS and TinyOS is the flexibility of Linux to be modified while the system is running or working. On the contrary, TinyOS requires the flashing process from the initial application if any changes or any additions are made to the algorithm.

The main contribution of this research is to offer a technical solution of simpler image compression on the WVSN platform, which is discussed in sections 2.2 and 2.4, and optimization of the parameters of the JPEG 2000 image compression on WVSN as reported in Section 3. The research focuses on energy consumption, memory utilization, and encoding time that affect the lifetime of WVSN network. The evaluation methods are measurements of voltage and current in WVSN devices using various compression parameters. Section 2 of this paper describes briefly the design of the system, followed by Section 3, in which the measurement results are presented. The conclusion of this research is described in Section 4.

II. System Design

One contribution of this research is to provide a model as seen in Fig 1. The model is developed for image compression on WVSN. The Embedded Linux OS is selected as the WVSN platform, as the node uses the 32 bits PXA271 processor. TinyOS as the default Bootloader of the node is replaced by Embedded Linux OS as the new Bootloader.

The following steps are applied. Firstly, in the WVSN node, the image is captured by an OV7670 camera by running Video for Linux 2 (V4L2) library, followed by compressing the image in JPEG 2000 format, which is produced by the OpenJPEG [18] library. Thirdly, tools for wireless communication among hops using IEEE 802.15.4 standard are installed. The tools have been configured before utilizing them in an ARM processor, in which Embedded Linux OS had been installed. Finally, the system performance is evaluated by measuring its energy consumption and processing time, PSNR, and memory utilization versus image compression parameters and transmission distance between transmitter node and receiver node.

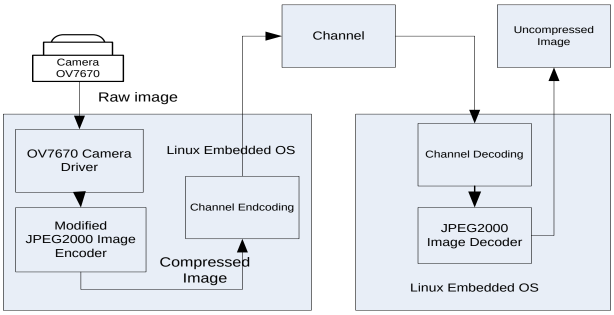


Figure 1. The Proposed System Design of WSN Imote2 for Image Transmission

A. *Wireless Sensor Network Platform*

The WVSN platform being used is Imote2 [16]. The Imote2 board uses the Intel Xscale PXA271 processor. This platform has a capacity of 32 MB SDRAM, 32 MB flash, and IO port as GPIO, 2x SPI, 3x UART, PC, SDIO, USM host, and USB client, and integrated wireless communication via the 802.15.4 (CC2420) standard. This board is designed to run under Intel Platform X, SOS, and TinyOS. In addition, a multimedia sensor board is used to capture raw images [17]. This board consists of a miniature speaker and line output, color image and video camera chip, omniVision OV7670, audio capture and playback CODEC WM890, onboard microphone and line input, and Passive InfraRed (PIR) motion sensor. The interface board performs data transfer, debugging, and flashing in radio board from computer host through cable. Table 1 displays Imote2 ‘s consumption energy parameters.

TABLE I.
PARAMETERS OF IMOTE2 RADIO BOARD [16-17]

Initial Energy	200 Joules
Active Camera withdraws	40 mW
PXA Processor Active	192.3 mW
Image Capture (including Sensing)	0.04 J
Radio Active	46.39 mikroJ
Read from SD Flash	5.32 nJ/bit
Write in SD Flash	7.65 nJ/bit
Image Dimension	640x480 pixel
Image Size	900 KB

B. *Embedded Linux Operating System*

Embedded Linux has been successfully developed in Imote2 board [7], replacing the TinyOS bootloader. With this embedded system, development of applications for scalar and multimedia data becomes easier and more dynamic. The advantages of the Embedded Linux OS are support for image sizes up to 28MB, and independent configuration or data storage. The communication model is technology independent (programming via wireless may be possible with same communication protocol). Linux also has the flexibility to be modified while working on a running system. Table 2 explains the advantages of Embedded Linux application, compared to the TinyOS bootloader.

TABLE 2.
TINYOS VS LINUX BOOTLOADER COMPARISION

	TinyOS	Embedded Linux
Image Size	Image size allowed is only 1 – 2 Mb. Therefore one node only processes one application.	Image size up to 28 Mb, one node has capability to process more than one application.
Code Loader	Platform specific codeloader. Specific platform and re-flashing are required to implement new application or modify existing application.	Platform independent. Only small modification of driver (or existing application) is required when new application is implemented.
Communication	Unreliable.	Independent, supports wireless communication.
Programming	Low data programming speed < 20 kbps on USB	Higher data programming speed due to better USB utilization of 350 kbps for permanent flash programming and of 950 kbps in RAM-execution
Community	Small number of TinyOS developers results in small number of references. Thus investigation of TinyOS will be slow.	Linux has many developer communities. Hence it is promising for developing research.

The critical step in the implementation of Embedded Linux OS is installing or uploading bootloader, kernel, and filesystem. Bootloader or bootloader object (Blob) is the system basic start-up used to initialize system components, i.e. system memory initialization, system clock management, and GPIO initialization. When Blob finishes system initialization, the system will automatically run the Linux kernel. The Kernel is generally read in a system file known as JFFS2. The JFFS2 itself is a system file that had been activated by the bootloader. The embedded system installed in flash memory puts the system file in the end of memory. The implementation steps are shown below.

Embedded Linux OS Implementation on WVSN Radio board

```
A. Embedded Linux Implementation
   1. Compiling Linux Kernel for WVSN
      Implementation
   2. Flashing Bootloader, Linux Kernel,
      and File system to WVSN Platform
B. Visual Sensor Implementation
   3. Modified Visual Sensor for Linux
      Implementation
   4. Cross compiling and upload Linux
      OV7670 driver
   5. Cross compiling and upload binary
      coding to capture image
C. JPEG 2000 Compression Implementation in
   WVSN
   6. OpenJPEG Library modification
   7. OpenJPEG Cross compiling for XScale
      Processor
   8. Upload binary code from Linux Host
      to WVSN with Embedded Linux
```

C. JPEG 2000

Generally, in the JPEG 2000 compression method, there are three preprocessing steps required before DWT processes, i.e. tiling, level offset, and Irreversible Color Transform (ICT) [14]. The tiling process is a process of dividing a source image into an image of similar size with non-overlapping rectangular tiles, except for tiles positioned on the image's border. Each tile is then compressed independently using its own compression parameter. This tiling process is aimed at reducing memory usage.

The C-Library source code of OpenJPEG [18] was used to implement JPEG 2000 on the system. The implementation steps are: 1) Compile OpenJPEG on the host computer by defining the desired target, i.e. Processor XScale PXA271; 2) Set up USBnet interface communication by configuring the IPV4 protocol for communication between the host computer and WVSN board; 3) upload the bin files and library to the WVSN board via the USBnet interface. By these three steps, JPEG 2000 has already been implemented on a WVSN platform that uses Processor XScale PXA271 and Linux as an embedded system, as shown in Fig. 1.

D. Communication Stage

The wireless communication used in Imote2 is the Zigbee/IEEE 802.15.4 method. In Embedded Linux OS, a driver called tosmac is required to activate IEEE 802.15.4. In order to transmit compressed images, files are split into packets, each just 28 bytes in length.

There are two steps performed in the transmission processes, i.e. in the request node and in the visual sensor node, as shown in Fig. 2. Communication is established when a user from request node asks sender node to capture an image. Request node then activates IEEE 802.15.4 and acts as coordinator, while the sender or visual sensor node periodically activates and deactivates IEEE 802.15.4 communication to acquire image capture requests. Camera OV7670 in the visual sensor node captures images as requested by running the V4L2 library. After capturing, the raw image is compressed into JPEG 2000 format by the OpenJPEG library. The compressed image is divided into packets of 28 bytes as the payload of the IEEE 802.15.4 communication standard.

IEEE 802.15.4. which is a low–rate wireless network standard, specifies the physical layer and Medium Access Control (MAC) layer. The MAC frame can be transmitted by the use of the physical channel and by the availability of the MAC layer. Therefore the 28 bytes payload packets, which are encapsulated by frames of the MAC layer, are sent to the user or receiver. An error control process for each payload of 28 bytes is not performed in this MAC layer communication.

The system will activate an IEEE 802.15.4 connection automatically when the system recognizes the availability of a transmitted packet. The user node will receive packets that are sent by camera node. The communication will be 'off' if the node has received a special packet from the camera node. The special packet is a notification that all packets have been sent by the user node. Then the packets are saved in a file.

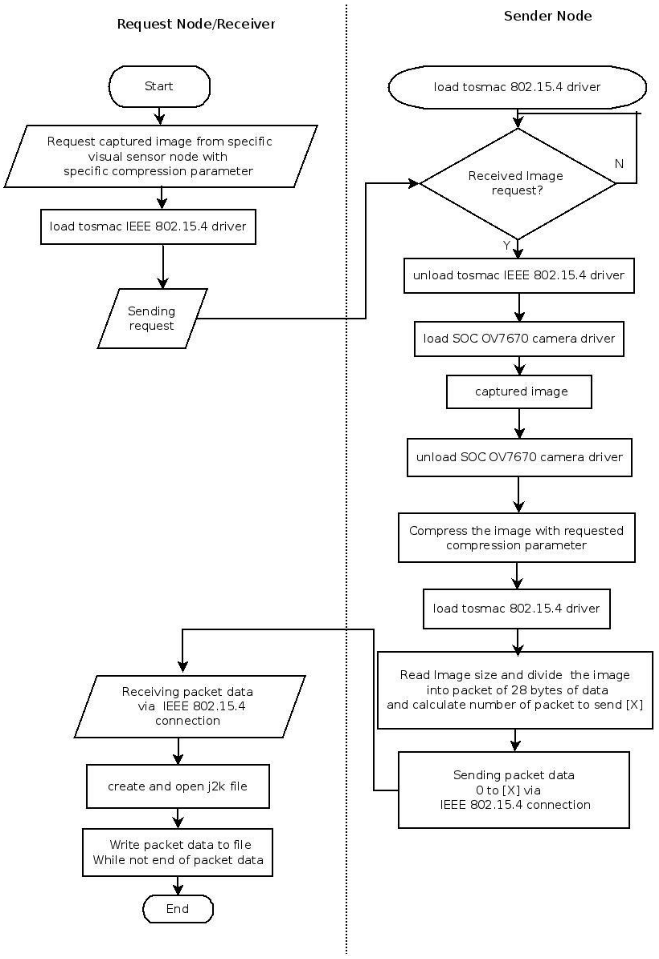

Request Node/Receiver
Sender Node
Start
Request captured image from specific visual sensor node with specific compression parameter
load tosmac IEEE 802.15.4 driver
Sending request
Receiving packet data via IEEE 802.15.4 connection
create and open j2k file
Write packet data to file While not end of packet data
End
load tosmac 802.15.4 driver
Received Image request?
N
Y
unload tosmac IEEE 802.15.4 driver
load SOC OV7670 camera driver
captured image
unload SOC OV7670 camera driver
Compress the image with requested compression parameter
load tosmac 802.15.4 driver
Read Image size and divide the image into packet of 28 bytes of data and calculate number of packet to send [X]
Sending packet data 0 to [X] via IEEE 802.15.4 connection

Figure 2. Communication Stage of Image Transmission over IEEE 802.15.4

E. Energy Consumption

Since the energy source of WVSN is solely from battery with limited power, energy consumption becomes a critical criterion on implementation. This means that the main priority of one particular model or any implementation method of the image virtual view generator is to produce low energy consumption. In general, there are three tasks that affect energy consumption of WVSN, i.e.: 1) image capture, 2) image processing, and 3) image transmission. These three energy consumption factors are formulated as below:

$$E_{WVSN} = E_{capt} + E_{proc} + E_{comm} \tag{1}$$

where E_{capt} denotes image capture energy consumption, E_{proc} denotes image processing energy consumption, and E_{comm} denotes image transmission energy consumption. Furthermore, energy consumption can be manually computed by measuring voltage (V), current (I), and time of the whole process i.e. image capture time (t_{capt}), image processing time (t_{proc}), and image transmission time (t_{comm}), with or without any compression methods, as:

$$E_{WVSN} = V \times I \times \left(t_{capt} + t_{proc} + t_{comm}\right) \tag{2}$$

The energy consumption of JPEG 2000 image compression can be estimated from DWT processes [19]. In this research, OpenJPEG was used as the JPEG 2000 representation using the Daubechies filter model 5-tap/3-tap in low and high pass analysis, respectively [19]. Equations regarding the filter in the Z-domain are:

$$L_0(z) = \frac{1}{8}\left(-z^2 + 2z + 6 + 2z^{-1} - z^{-2}\right) \tag{3}$$

$$H_0(z) = \frac{1}{2}\left(z + 2 - z^{-1}\right) \tag{4}$$

where $L_0(z)$ defines the low frequency subband of the DWT filter and $H_0(z)$ defines the high frequency subband of the DWT filter. The impulse responses of the filter are [20]:

$$L[2n] = \left[\begin{array}{c} -x[2n-2] + 2x[2n-1] \\ \dfrac{+6x[2n] + 2x[2n+1] - x[2n+2]}{4} \end{array}\right] \tag{5}$$

$$H[2n+1] = \left[\frac{-x[2n] + 2x[2n+1] - x[2n+2]}{2}\right] \tag{6}$$

Analyzing the energy consumption of DWT image compression requires the size and type of some basic operations of the given process. Basic operations such as shifting (S) and adding (A) can be derived from (5) and (6) [20]. In (5), filter decomposition requires 8 shifts (S) and 8 adds (A) operations to convert 1 pixel of an image into low pass coefficient Meanwhile, (6) requires 2 shifts (S) and 4 adds (A) to convert into high pass coefficient. Overall computation of an image on DWT decomposition is acquired by counting the whole operations. The computation of one image of size $M \times N$ with L DWT level is estimated step by step as follows. Assume that the first step is horizontal decomposition. Since the whole pixels at even positions are decomposed into low pass coefficients and pixels at odd positions are decomposed into high pass coefficients, such that the total weight at horizontal is $\frac{1}{2}MN(10S+12A)$. The vertical decomposition has similar weight. Image's size will reduce with factor of 4 from its initial size for every transformation level. Then, the total computation load becomes:

$$C_{DWT}(M,N,L) = MN(10S + 12S)\sum_{l=1}^{L}\frac{1}{4^{l-1}} \tag{7}$$

If the image is divided into several tiles with size $Q\times R$ that is less than or equal to $M\times N$ before the DWT, then (7) becomes:

$$C_{DWT}\left(Q,R,L\right) = \sum_{1}^{\left(M/R\right)}\sum_{1}^{\left(N/Q\right)}\left(QR\sum_{l=1}^{L}\frac{1}{4^{l-1}}\left(10S + 12A\right)\right) \tag{8}$$

Apart from arithmetic operations on transformation steps, energy is significantly consumed from memory access. Within memory access processes, each pixel goes through reads and writes 2 times, so memory access load for an $M \times N$ image is:

$$C_{read}\left(Q,R,L\right)=\sum_{1}^{\left(M/R\right)}\sum_{1}^{\left(N/Q\right)}2QR\sum_{l=1}^{L}\frac{1}{4^{l-1}}$$

$$C_{write}\left(Q,R,L\right)=\sum_{1}^{\left(M/R\right)}\sum_{1}^{\left(N/Q\right)}2QR\sum_{l=1}^{L}\frac{1}{4^{l-1}} \qquad (9)$$

Total processing load is computed as sum of computational load (C_{DWT}) and memory access load, (C_{read}) and (C_{write}), from (8) and (9) and becomes:

$$C_{proc}=C_{DWT}+C_{read}+C_{write}$$

$$C_{proc}\left(Q,R,L\right)=$$

$$\sum_{1}^{\left(M/R\right)}\sum_{1}^{\left(N/Q\right)}\left(QR\sum_{l=1}^{L}\frac{1}{4^{l-1}}\left(10S+12A\right)\right)+ \qquad (10)$$

$$\sum_{1}^{\left(M/R\right)}\sum_{1}^{\left(N/Q\right)}2QR\sum_{l=1}^{L}\frac{1}{4^{l-1}}+\sum_{1}^{\left(M/R\right)}\sum_{1}^{\left(N/Q\right)}2QR\sum_{l=1}^{L}\frac{1}{4^{l-1}}$$

Furthermore, if E_{shift} denote energy consumption for shift (S) operation, E_{add} denote energy consumption for add (A) operation, E_{read} denote energy consumption for read memory access, and E_{write} denote energy consumption for write memory access, then the overall energy consumption for processing becomes:

$$E_{proc}=E_{DWT}+E_{read}C_{read}+E_{write}C_{write}$$

$$E_{proc}\left(Q,R,L\right)=$$

$$\sum_{1}^{\left(M/R\right)}\sum_{1}^{\left(N/Q\right)}\left(QR\sum_{l=1}^{L}\frac{1}{4^{l-1}}\left(10E_{shift}+12E_{add}\right)\right)+$$

$$E_{read}\sum_{1}^{\left(M/R\right)}\sum_{1}^{\left(N/Q\right)}2QR\sum_{l=1}^{L}\frac{1}{4^{l-1}}+E_{write}\sum_{1}^{\left(M/R\right)}\sum_{1}^{\left(N/Q\right)}2QR\sum_{l=1}^{L}\frac{1}{4^{l-1}}$$

$$E_{proc}\left(Q,R,L\right)=$$

$$QR\sum_{1}^{\left(M/R\right)}\sum_{1}^{\left(N/Q\right)}\left(10E_{shift}+12E_{add}+2E_{read}+2E_{write}\right)\sum_{l=1}^{L}\frac{1}{4^{l-1}} \qquad (11)$$

Image transmission is generally in the form of data packets. If the number of packets is m, average packet size is t, and energy required to transmit 1 bit data is E_{b}, transmission energy is defined as:

$$E_{comm}=m\times t\times E_{b} \qquad (12)$$

The IEEE 802.15.4 communication system on Embedded Linux OS uses the tosmac structure data packet [1], with the payload of each packet being 28 bytes. From this model and substituting (11) and (12) into (1), the energy consumption for transmitting a JPEG 2000 compressed image becomes:

$$E_{WVSN}=E_{capt}+E_{proc}+E_{comm}$$

$$E_{WVSN}\left(Q,R,L\right)=E_{capt}+$$

$$QR\sum_{1}^{\left(M/R\right)}\sum_{1}^{\left(N/Q\right)}\left(10E_{shift}+12E_{add}+2E_{read}+2E_{write}\right)\sum_{l=1}^{L}\frac{1}{4^{l-1}}+ \qquad (13)$$

$$m.t.E_{b}$$

Equation above, (13), indicates that, 1) On the energy consumption in the compression process, the use of smaller tiles reduces memory utilization, since the same memory size can be used to process more tiles. But processing small tiles requires extended time for splitting image into tiles; 2) Compression ratio or coding rate affects image size. Lower bit rate means smaller image size, M × N, which also means smaller size of file. Since M × N is factor of energy consumption in compression process, energy consumption is getting lower for smaller bit rate; 3) Energy consumption increases when the value of variable L increases. Because L is the number of DWT level being used, energy consumption increases with the DWT level; 4) Energy consumption for transmission is affected by the number of packet *m* and packet size *b*. Energy consumption increases when either or both m and b increase, and vice versa.

F. Peak Signal to Noise Ratio (PSNR)

Error metrics often used to compare various image compression techniques are the Mean Square Error (MSE) and the Peak Signal to Noise Ratio (PSNR). The MSE is the cumulative squared error between the compressed and the original image, whereas PSNR represents a measure of the peak error. The mathematical formulae for PSNR and MSE are

$$MSE=\frac{1}{vw}\sum_{i=0}^{v-1}\sum_{j=0}^{w-i}[I(i,j)-K(i,j)]^{2} \qquad (13)$$

$$PSNR=20\log_{10}\left(\frac{MAX_{I}}{\sqrt{MSE}}\right) \qquad (14)$$

where *I* represents matrix of original images, *K* is compressed image, *m* is number of pixel row of image and i is index of that row, n represents number of image's pixel column and j is index of that column.

G. Bit Error Rate (BER)

The bit error rate (BER) is defined as the rate at which errors occur in a transmission system. This can be directly translated into the number of errors that occur in a string of a stated number of bits. This definition can be translated into a (15) as follows:

$$BER=\left(\frac{\sum_{1}^{n}e_{b}}{\sum_{i}^{n}b}\right) \qquad (15)$$

III. Result and Discussion

A. Measurements Method

Specific compilation is required such that OpenJPEG is adaptable on the target processor XScale PXA271 with Embedded Linux OS. Source image is the original image produced by capturing an image using a multimedia sensor with 640 × 480 pixel resolution Fig. 7a). Moreover, two approaches are used to evaluate JPEG 2000 in the WVSN environment. The first is testing OpenJPEG with its default parameter. The second is modifying JPEG 2000 parameters on DWT level, compression level, tile size, and block code size, as shown in table 3. The experiments are carried out as many as 250 times using simultaneous encoding processes. Image compression of OpenJPEG Library with different parameters alters energy values of (12).

The measurement of initial voltage and current in each scenario are assumed as initial power, as resulted from $V \times I$ = 4.5×20 = 90 mW. This initial value is assumed to be the power of the system before any process is performed, or to be 100% of total power. By this, the energy reduction stated in this paper is energy consumption as a percentage of initial energy in each scenario.

TABLE 3.
TESTING PARAMETERS

DWT Level	1, 2, 3, 4, 6
Compression	Lossless
Bit Rates	1, 0.125, 0.050, 0.0167, 0.010
Size of Block Code	32x32 (300 bcs), 64×32 (150 bcs), 64×64 (75 bcs)
Size of tiles	640×480 (1 tiles), 320×240 (4 tiles), 128×128 (8 tiles)

These experiments are conducted under various conditions:

1. Scenario I. Measurement of energy consumption and encoding time for DWT level of 1, 2, 3, 4 and 6 with bitrate 1 bpp and 0.017 bpp.
2. Scenario II. Measurement of energy consumption, encoding time and PSNR for tile size optimization at a certain bit rate.
3. Scenario III. Measurement of energy consumption and encoding time, and computation of PSNR for different sizes of block code.
4. Scenario IV. Performing test of transmission time for various compression ratios and tile number, where distances of object to camera are set to be 10, 20, and 30 meters.
5. Scenario V. Measurement of energy consumption and transmission time for distances of 10, 20, and 30 meters.

B. *Experiment Results*

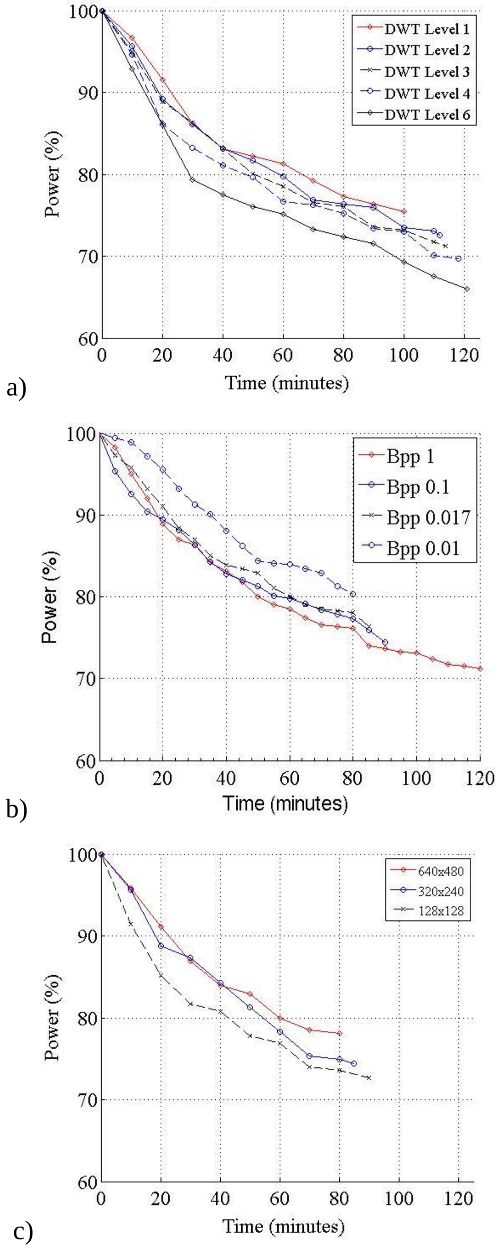
In scenario I, at a rate 1 bpp, it is observed that varying the level of DWT affects energy consumption. Fig. 3a shows that energy consumption increases at higher DWT level. Energy consumption for DWT level 1 is 25.51% of available initial power, and this is the lowest value of energy consumption compared to other DWT levels. The highest energy consumption occurs when DWT level 6, i.e. 33.99% from initial power. In general, less energy is required for lower DWT levels. These results are consistence with review of (13) in the processing energy where the energy consumption will increase along with DWT Level, *L*. On the other hand, the level of DWT has effects on encoding process time as shown in Fig. 3a, where a higher DWT level consumes more time for the encoding processes.

Although JPEG 2000 in OpenJPEG is capable of delivering lossless and lossy compression, according to description in Section 2.5, in this paper only lossless compression methods were used. It was found that at a rate of 1 bpp, changing the DWT level does not alter the PSNR value. For lower bit rates, the PSNR value is altered as shown in Fig. 4. The PSNR value of DWT level 1 is 11.26% lower than the PSNR of DWT level 6.

Fig. 4 shows PSNR versus varying bit rate on DWT level. This figure shows that the change in DWT level influences the PSNR values. The lower level of DWT generates lower PSNR, and vice versa. The effect of DWT level to PSNR values for DWT level of 1, 2, 3 and 4 is relatively higher, and even higher on lower bit rate. However the difference PSNR values of DWT level of 4 and 6, i.e. less than 2%, is not significant. Furthermore, from these experiments, it is seen that varying tile sizes and block codes do not change the PSNR value.

Bit rate changes affect energy consumption. Fig. 3b shows energy consumption of DWT level 3 with various bit rates. A lower bit rate consumes less energy; for example, energy consumption at bit rate of 0.1 is 25.53% while at 0.01 is 19.65%. Bit rate also affects encoding time. Lower bit rate results in shorter encoding times as shown in Fig. 3b.

The result for Scenario II is shown in Fig. 3c. Tile size or number of tiles in an image affects both energy consumption and encoding time. Energy consumption is higher for smaller tile size and the time required for the encoding process is longer. Fig. 3d shows the results of the scenario III. Smaller block code size consumes more energy, whereas the time required for the encoding process is longer.



a)

b)

c)

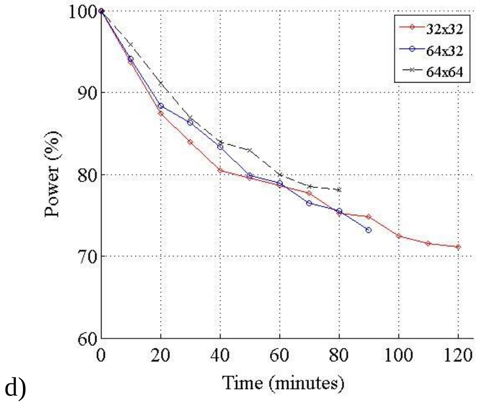


Figure 3. Energy consumption on various JPEG 2000 parameters (a) Energy consumption for various DWT Levels at bit rate of 1 (b) Energy consumption at various bit rate. (c) Energy consumption at various tile sizes (d) Energy consumption at various block code sizes.

C. Memory Utilization

The constraint of memory of WVSN defines the system's computation capability. Reducing the tile size can reduce the memory usage. Furthermore, reducing the block code size will reduce the memory usage too. Smaller size will save memory but increase energy consumption as shown in Fig. 3c. In addition, the tile size of 128×128 and 320×240 cost less memory as much as 75% and 50% compared to tile size of 640×480.

D. Transmission Process

Fig, 5 depict the results of the fourth scenario, the experiments of data transmission. Sending one raw image (with no compression) requires time of 10 to 15 minutes. The transmission time decreases as much as 70 seconds and between 10 to 15 seconds by using JPEG 2000 compression method with bit rate of 0.1 and 0.01 bpp respectively.

Fig. 6 depicts the result of the fifth scenario. This indicates that reducing the distance from 30 meters to 10 meters does not significantly reduce energy consumption. It is found that within 300 minutes of transmission time at 10, 20, and 30 meters distances, the energy consumptions are 17.19%, 18.10%, and 19.41% from initial power.

Table 4 shows BER values for distances of 10, 20, and 30 meters at bit rate of 1, 0.2, and 0.1 bpp. This table is indicates that distance between transmitter and receiver affect BER. BER value will increase as the distance between transmitter and receiver is increased.

TABLE 4.
BER VS VARIOUS DISTANCE AND BIT RATE

range (meter)	Raw	bpp=1	bpp=0.2	bpp=0.1
10	0.0400	0.0308	0	0
20	0.0512	0.0447	0	0
30	0.0695	0.0508	2.51E-05	0

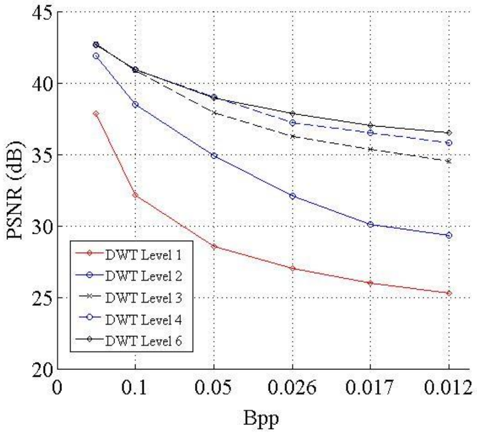


Figure 4. PSNR vs bpp on various DWT Level

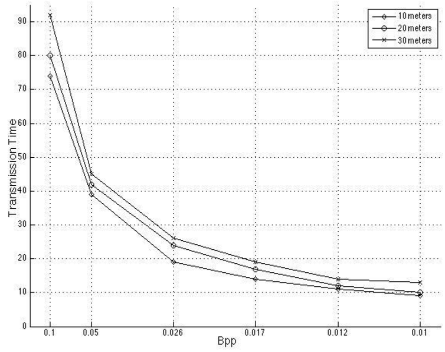


Figure 5. Transmission time on various Bpp

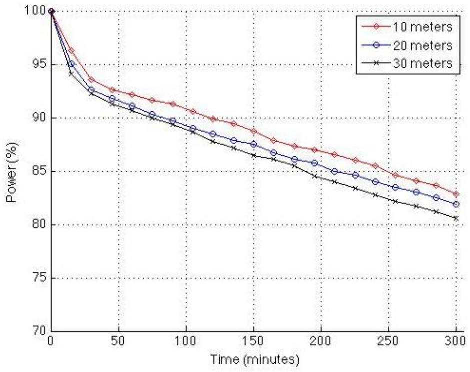
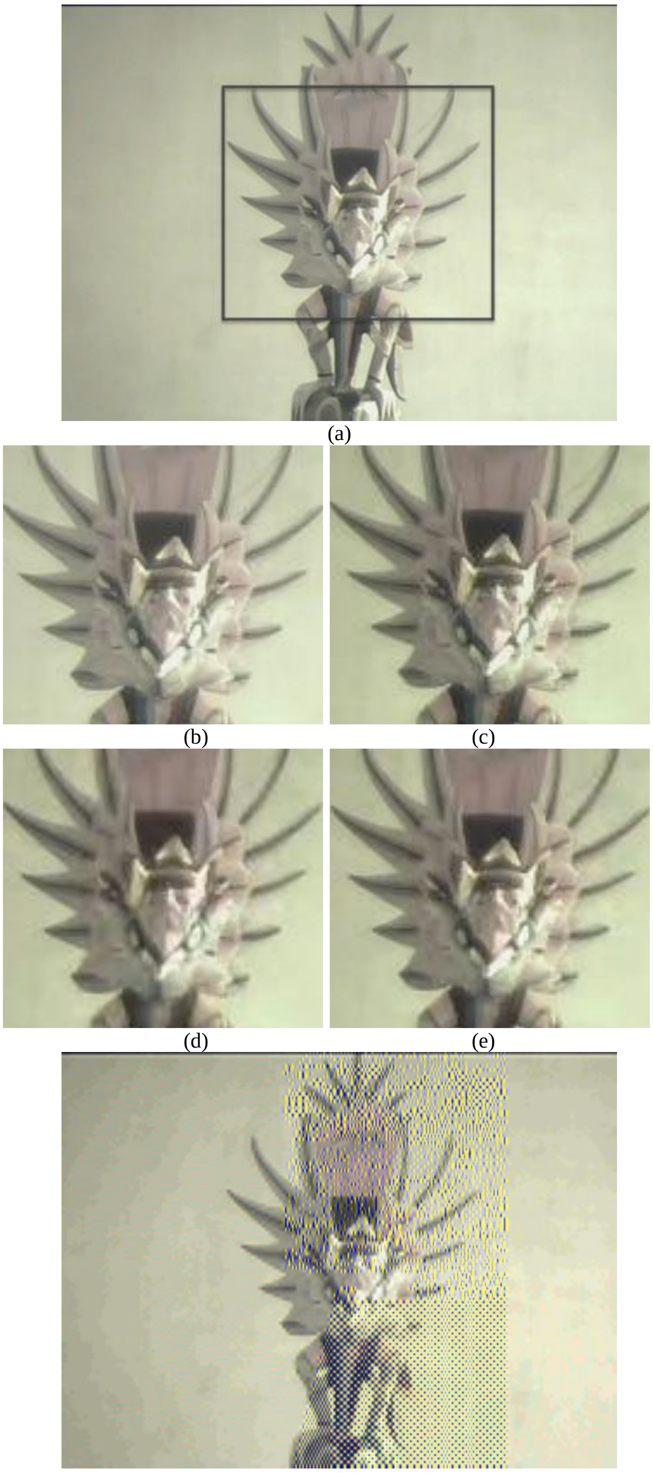


Figure 6 Power Consumption at various Tx – Rx distances

IV. Conclusion

The implementation steps of JPEG 2000 image compression with OpenJPEG library on Embedded Linux have been described. The wireless communication system is developed for IEEE 802.15.4 from the tosmac driver to send an image in the form of 28-byte data packets. The communication system has demonstrated that BER increased when the distance between Tx node and Rx node, and the bit rate are increased. Lower BER can be achieved with the use of an error control mechanism added in the layer above the MAC layer. However, the mechanism increases energy consumption as two-way communication is required for error checking, and packet resend is also required if an error packet is obtained by the receiver.



(a)

(b) (c)

(d) (e)

(f)

Figure 7 Raw images and result images. (a) Raw image, capture from OV7670 visual sensor (b) Raw image (zoomed) (c) Compressed image, bpp 0.1, tile 128×128 (d) Compressed image, bpp 0.01, Tile 128×128 (e) Compressed image, Bpp 0.01, tile 256×256, (f) Received image at Bpp 0,01, tile 640×480, and 20 meters of Tx – Rx distance.

In this research, the energy consumption modelling for image communication on WVSN environtment as in 13) has been proven by measurements. It is also observed that the parameters of JPEG 2000, in terms of DWT level variations, bit rates, tile sizes, tile number, and block code sizes have significant effects on both energy consumption and memory utilization. Five experimental scenarios were designed to observe energy consumption, and showed that the optimal energy efficiency is achieved at the parameters of DWT level 3, bit rate of 0.01 bpp, tile number of 1, and block code size of 64□64. This setting yields energy efficiency as much as 14.34% at PSNR 34.15 dB compared to the default

OpenJPEG setting (bpp 1, DWT level 6, tile number 1, and block code = 64□64). Finally, it was found that bit rate affects energy consumption significantly. Smaller bit rates consume less energy. Similarly, tile size affects energy consumption, such that a small tile number utilizes less energy. The same condition occurs on varying the block code size. The energy consumption increases when block code is smaller. Lastly, variations of DWT level influence the PSNR values. The lower level of DWT generates lower value of PSNR, whereas encoding time increases for higher DWT level.

References

[1] D. G. Costa, I. Silva, L.A. Guedes, F. Vasques, P. Portugal Availability issues in wireless visual sensor networks. *Sensors (Basel, Switzerland)* 2014; 14(2): 2795-2821. Available: http://dx.doi.org/10.3390/s140202795.

[2] Ian F. Akyildiz and Mehmet Can Vuran, Wireless Sensor Network. John Wiley & Sons Ltd., 2010, pp. 349-352.

[3] Stanislava Soro and Wendi Heinzelman, “A survey of visual sensor networks,” *Advances in Multimedia*, vol. 2009, Article ID 640386, 21 pages, 2009. Available: http://dx.doi.org/10.1155/2009/640386.

[4] Y. Charfi, N. Wakamiya, Murata, M., "Challenging issues in visual sensor networks," *Wireless Communications, IEEE*, vol.16, no.2, pp.44,49, April 2009. Available: http://dx.doi.org/10.1109/MWC.2009.4907559

[5] P. Levis, TinyOS Programming. Cambridge University Press, 2009, pp. 13-18, pp. 137-138.

[6] D. Agrawal, “Exploiting the interplay between memory and flash storage for sensor data management,” in Proc. *IEEE 16th International Conference on Embedded and Real-Time Computing Systems and Applications*, 2010. Available: http://dx.doi.org/10.1109/RTCSA.2010.10.

[7] R. Parthasarathy, N. Peterson, S. WenZhan, A. Hurson, and B. A. Shirazi, “Over the air programming on imote2-based sensor networks,” in Proc. *43rd Hawaii International Conference on System Sciences (HICSS)*, 2010, pp. 1–9. Available: http://dx.doi.org/10.1109/HICSS.2010.300.

[8] M. Wu and C. W. Chen, “Collaborative image coding and transmission over wireless sensor networks,” *EURASIP J Appl Signal Process*, vol. 2007, no. 1, pp. 223–223, 2007. Available: http://dx.doi.org/10.1155/2007/70481.

[9] L. Vincent, D.-F. Cristian, and K. Nicolas, “Energy-efficient transmission of wavelet-based images in wireless sensor networks,” *J Image Video Process*, vol. 2007, pp. 15–15, 2007. Available: http://dx.doi.org/10.1155/2007/47345.

[10] D. V. Babu, N. R. Alamelu, P. Subramanian, and N. Ravikannan, “EBCOT using energy efficient wavelet transform,” in Proc. *IEEE International Conference on Computing, Communication and Networking*, 2008, pp. 1–6. Available: http://dx.doi.org/10.1109/ICCCNET.2008.4787746

[11] Kyu-Yeul Wang, et. al, “Low-MAC FEC controller for JPEG2000 image transmission over IEEE 802.15.4,” in Proc. *the World Academy of Science*, Engineering and Technology, 2009. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.192.8739&rep=rep1&type=pdf.

[12] David S. Taubman; Michael W. Marcellin, JPEG 2000 Image Compression Fundamental, Standard and Practice. Kluwer Academic Publisher, 2002, pp. 3-17.

[13] Zhe-yuan Xiong, Xiao-ping Fan, Shao-qiang Liu, and Zhi Zhong, “Low complexity image compression for wireless multimedia sensor networks,” in Proc. *Int. Conf. Inf. Sci. Technol*. March 26-28 2011 Nanjing Jiangsu China, 2011. Available: http://dx.doi.org/10.1109/ICIST.2011.5765335.

[14] Dong-U Lee, Hyungjin Kim, Mohammad Rahimi, Deborah Estrin, and John D. Villasenor, “Energy-efficient image compression for resource-constrained platforms,” *IEEE Trans. IMAGE Process*., vol. VOL. 18, 2009, pp. 2100-2113. Available: http://dx.doi.org/10.1109/TIP.2009.2022438.

[15] M. Mowafi, F. Awad, E. Taqieddin, and O. Banimelhem, “Experimental evaluation of image compression and transmission for TinyOS-based imote2 platform,” in Proc. *Innovations in Information Technology* (IIT), 2011 International Conference on, 2011, pp. 173–178. Available: http://dx.doi.org/10.1109/INNOVATIONS.2011.5893811.

[16] Imote2 Hardware Reference Manual, Revision A, September 2007 PN: 7430-0409-01, Availabel: http://wsn.cse.wustl.edu/images/9/90/Imote2_hardware_ref.pdf. Last access: July 3, 2014.

[17] Imote2 Multimedia, IMB 400, CrossBow Technology INC, Document Part Number: 6020-0390-01 Rev-A PRELIMINARY Subject to change without prior notice, http://www.xbow.jp/imb400.pdf Last Access: http September 14, 2013.

[18] OpenJPEG,” 2012. http://openjpeg.org. Accessed September 13, 2013.

[19] D. Le Gall and A. Tabatabai, “Sub-band coding of digital images using symmetric short kernel filters and arithmetic coding techniques,” in *Acoustics, Speech, and Signal Processing, 1988. ICASSP-88., 1988 International Conference on*, 1988, pp. 761–764 vol.2. Available: http://dx.doi.org/10.1109/ICASSP.1988.196696.

[20] Dong-Gi Lee and S. Dey, “Adaptive and energy efficient wavelet image compression for mobile multimedia data services,” in *Communications, 2002. ICC 2002. IEEE International Conference on*, 2002, vol. 4, pp. 2484–2490 vol.4. Available: http://dx.doi.org/10.1109/ICC.2002.997290.

[21] A. Canclini, et.al., “Enabling visual analysis in wireless sensor networks,” in *Image Processing (ICIP), 2014, IEEE International Conference on*, pp.3408-3410, 27-30 Oct. 2014. Available: http://dx.doi.org/10.1109/ICIP.2014.7025690.

[22] M. Gonzalez, J. Schandy, N. Wainstein, N, Barboni, L, A. Gomez, C. Croce, "Wireless image-sensor network application for population monitoring of lepidopterous insects pest (moths) in fruit crops," in *Instrumentation and Measurement Technology Conference (I2MTC) Proceedings, 2014, IEEE International*, pp.1394-1398, 12-15 May 2014. Available: http://dx.doi.org/10.1109/I2MTC.2014.6860975.

[23] N. P. Sastra, Wirawan, G. Hendantoro, “Virtual view image over wireless visual sensor network,‘ *Telkomnika*, vol. 9, no. 3, pp. 489-496, 2011. Available: http://dx.doi.org/10.12928/telkomnika.v9i3.740

Nyoman Putra Sastra received the B.Eng. and M.Eng. degree in electrical engineering from Institut Teknologi Bandung (ITB), In-donesia, in 1998 and 2001, respectively. In 2015 he receivied his Ph.D. degree from Institut Teknologi Sepuluh Nopember (ITS), Surabaya. Since 2001, he joined with Universitas Udayana, Bali as lecturer. His research interests include wireless multimedia sensor networks and multimedia signal processing. He is a Member of the IEEE.

Wirawan received the B.E. degree in telecommunication engineering from Institut Teknologi Sepuluh Nopember (ITS), Surabaya, Indonesia, in 1987, the DEA in signal and image processing from Ecole Suprieure en Sciences Informatiques, Sophia Antipolis, and the Dr. degree in image processing from Telecom ParisTech (previously Ecole Nationale Suprieure des Tlcommunications), Paris, France, in 1996 and 2003, respectively. Since 1989 he has been with ITS as lecturer in the Electrical Engineering Department. His research interest lie in the general area of image and video signal processing for mobile communications, and recently focuses more on underwater acoustic communication and networking and on various aspects of wireless sensor networks. He is a member of IEEE.

Gamantyo Hendrantoro received the B. Eng degree in electrical engineering from Institut Teknologi Sepuluh Nopember (ITS), Surabaya, Indonesia, in 1992, and the M. Eng and PhD degrees in electrical engineering from Carleton University, Canada, in 1997 and 2001, respectively. He is presently a Professor with ITS. His research interests include radio propagation modeling and wireless communications. He has recently been engaged in various collaborative studies, including investigation into millimeter wave wireless systems for tropical areas with Kumamoto University, Japan, implementation of digital TV systems with BPPT, Indonesia, and development of Indonesia’s first educational nanosatellite (IINUSAT) together with five other universities and LAPAN. Dr. Gamantyo Hendrantoro is a senior member of the IEEE.