Error Performance of JPEG and Watermark Authentication Techniques in Wireless Sensor Network

Mohammad Abdus Salam and Hesham Elashhab

Department of Computer Science
Southern University, Baton Rouge, Louisiana 70813, USA

ABSTRACT

Wireless sensor network (WSN) iscomposed of tiny sensors and they are powered by energy-constraint battery. The continual need to utilize limited bandwidth to absorb most of the packets has led to a great deal of research in the field of data compression. WSN has been exploited to balance the maintenance of readable information content with acceptable error and the cost of energy consumed by the collecting nodes during the compression. In this paper, we compared two compression techniques: JPEGand watermark authentication. We applied both techniques and used error measurement to indicate system performance regardless of the number of execution instructions. Wefound that it is not enough to use errormeasurement onlybut the number of executing instructions must also beconsidered. The digital watermark approach showed a lower average error compared to JPEGapproach. However, there were fewer instructions in the JPEG approach compared towatermark approach.

KEYWORDS

Wireless Sensor Network, Data Compression, Aggregation, JPEG, Security

1. INTRODUCTION

Advancements are emerging in the field of ad-hoc networks, and wireless sensor networks are no exception. Although these networks have achieved high performance and provide a practical environment for applications in this field, a great deal is still to be gained through the use of such technology. Understanding, knowledge, and experience will bring effective results and additional control to this technology; however, WSN suffer from problems, such as security and power consumption, on a variety of levels.Energy consumption is one of the major concerns in WSN. Such a restriction has motivated many researchers to look elsewhere in the network, such as at the node level, to save power; these results are reflected in the network’s lifetime and efficiency [1]-[5]. We suggest a criterion for compression method utilization in order to compress raw data at the collecting sensor nodes before transmitting any information to the sink (base station).

Our contribution to the problem is to shed light on developing and finding better ways to use compression algorithms in WSN and to define a performance measurement depends on measuring

DOI : 10.5121/ijsptm.2014.3301

the error at the receiving end. It is important to keep inmind the number of steps used to compress raw data and decompress it at the sink inorder to calculate power consumption in the network.

2. DATA COMPRESSION

In general, compression algorithms provide increased power efficiency and makeroom for more data in the memory; they are used in almost all aspects of wirelesstechnology [6]. Blellochdescribes compression as occurring in two units [6]:

1- An encoding algorithm to process the signal or the raw data and provide arepresentation for the data in which it should be less than the original size.
2- A decoding algorithm to extract (reconstruct) the original signal and decompressthe data at the destination.

Wireless sensor networks require compression algorithms because their sensors havelimited memory and restricted power capacity [3]. A variety of effective compressionalgorithms are available; however, in the case of WSN, the choice is complicated becauseof these restrictions.For these reasons, it is difficult to implement some of the algorithms in WSN, wherethey will not serve the purposes of improving overall efficiency and optimizing batterylife.Thealgorithms are dividedinto two categories:

1- Lossless compression - In which the information is compressed, sent, anduncompressed exactly as in the original state - not one bit is missing.
2- Lossy compression -It works under the same principles as lossless compression,except it does not compress the whole message; it does not compress very highfrequency bits. Especially in pictures, the absence of high frequency bits will notdamage the important content and the eye is still able to recognize the picture.

Both the nature of the application and the data affect the choice of which type ofcompression algorithm is applied in WSN. This does not mean that any changes shouldnot be adjusted to the network’s design to suit the purpose of the application.In general, most applications of WSN that use compression algorithms tend to use lossycompression because the requirements of the algorithm are considered moderate and notas high as the lossless requirements. In applications that need precision in transmittingand receiving data, text in particular, it is better to use lossless compression to avoid anychance of missing any part of it. Using lossy compression techniques on data that has textcontent is possible but with minimum tolerances [6].One well-known example of lossy compression that we will use in this paper inassociation with WSN is JPEG (Joint Photographic Experts Group).

In WSN, when using applications that deploy compression algorithms, lossycompression is used to save energy, which means that some of the received data ismissing. At this point, an error should be calculated to see how different the results arefrom the original data. This paper highlights the cases of two different compressionalgorithms, which are examined to determine their similarities and differences and toevaluate their performance. We measured performance by estimating the error inboth systems.

In WSN, it is necessary to choose the algorithm that occupies less memory, has less access to the memory, and consumes less power. We use theaverage error as a performance measurement to estimate the error resulting from thedifferences in data before and after transmission. The accuracy of the measurementperformance is not perfect, but it provides a good guideline for making a decision.In order to compare two data compression techniques, we studied JPEGcompression and a watermark-based approach by Wei Zhang[8]. Studying each system and understanding the equations for each technique, as well asapplying some examples will help users to choose the best technique for their specificapplication and type of WSN.

Two important scientific terms need to be defined beforemoving forward. The first one is aggregation, which means organizing and minimizingrepetition of information. Sometimes aggregation is just a simple mathematical operation.The second term is compression, which is the operation that describes the handling andprocessing of raw data to minimize its size in order to minimize the overhead whentransmitting packets without affecting the overall purpose of the message.

2.1 Aggregation

The process of summarizing and combining data from sensor networks to minimize theload of data transmitted to a destination is called aggregation [10]. Some of the benefitsof aggregating data before transmitting it include efficient utilization of the bandwidthand improvement of network performance. On the other hand, delays occur as a result ofprocessing and sorting before the transmission, which affects the time consumed inaggregating data. Accuracy and security are other issues that researchers look forward tosolving [11].

2.2 Secure Data Aggregation

Security is always a concern when it comes to WSN. Some of the issues in suchnetworks occur because of attacks by hackers trying to take control of servers. The leastdamage that a hacker could cause is obtaining knowledge about the information that youare attempting to keep secure from the public—either because it involves privacyagreements or sensitive information. In some cases, the system is fully operational and nostorage units or memories are affected by the attack; however, this is still viewed as aharmful action. The many studies and papers regarding secure data aggregation have ledto a number of options for securing data with some tradeoffs; these should be consideredcarefully, depending on the application. To examine the subject of security more closely,we need to study the work of Zhang et al. [8], which uses digital watermarking as anauthentication method for transmitted data and includes both aggregation andcompression.Typically, security and data aggregation involves: (i) decrypting the data when it is received by the aggregators, (ii) aggregating the data according to the aggregating function, and (iii) encrypting the data before sending it forward to the next node.

2.3 Digital Watermarking

Watermarking technology is mainly used to protect copyright ownership. The mainidea is to hide information within itself where the malicious node sees the data asnoise [7]. In general, most watermarking generation systems consist of an embedder and a detector.

2.4 JPEG Compression

The JPEG standard is a well-known compression method based on DiscreteCosine Transform (DCT). Using JPEG, the accuracy of the data at the receiving end is 8bits, and the elements in the quantized array are limited to 11bits [9]. This method is usedin a number of different applications, depending on the type of JPEG compressionpreferred by the customer [9]. In general, JPEG has three baselines (1) a lossy baseline coding system, (2) an extended coding system, and (3) a lossless independent coding system. We have chosen the lossy baseline system, which depends on DCT. The compressionsequencesare (a) DCT computation, (b) quantization, and (c) variable–length code assignment.

3. METHODOLOGY

The methodology for approaching the comparison is statistical, where the error is defined and applied to both methods to determine which method has less error.The watermark and JPEG approaches have similarities in their transformingmethods. Their difference lies in quantization during the compressing process. In thewatermark approach, the quantization uses a different method depending on K-largestcoding. The methodology for approaching the comparison is statistical, where the error isdefined and applied to both methods to determine which method has less error. In our research, the numbers of samples taken from each method are not equal becausethere is less sensory data from the watermark approach than from the JPEG approach. We substituted the missing elements with zeros to pad the restof the array to fulfill the concept of 8*8 pixel blocks in the JPEG approach; the resultswere not as expected.

3.1 Error Calculation

The error estimation is used to estimate the differences in the samples to determine how the results deviate from the original value.

$$E= X_0-X_1 \qquad (1)$$

WhereE represents the difference, X_0 represents the original sample before compressing, and X_1 represents the approximated value that is extracted at the receiving end. Making sure that the differences add up and preventing the effect of the polarity of the differences, we take the absolute value of the difference.

$$E= | X_0-X_1| \qquad (2)$$

The average or mean of the differences is expressed as:

$$M= \frac{1}{N}\sum_{0}^{N-1} | X_0 - X_1| \quad (3)$$

HereN represents the number of elements in the array. If we still want to relate the N to the dimensions of the array, then we have to make some modifications to Eq. (3) without affecting the results.

$$M= \frac{1}{N} \cdot \frac{1}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} | X_0 - X_1| \quad (4)$$

The Eq. (4) is more practical than the first one, because of the simplicity of using the same parameters. It is still useful to use the Eq. (3), but we have to make sure that we use the number of elements in the array not the number of bits. We used discrete cosine transform dct2 () function of the Matlab® to save time and make calculations faster and easier. This tool quickly calculates the discrete cosine transform. We only need to enter the array and the tool will compute the coefficient array.

4. ANALYSIS OF ERRORS

Understanding the error as a performance metric regardless of the number of execution stages is the core of this research. Based on the definition of the error measurement in the previous section, the array of differences in each method showed almost the same results, with small differences. In addition, the calculations in the watermark approach were performed with an unequal number of samples, an equal number of samples, and in another case, the same data was used in both approaches.

4.1 Unequal Number of Samples

4.1.1 JPEG Approach

JPEG is a lossy image compression technique [12]. Fig. 1 shows the different stages for the image compression technique using JPEG.

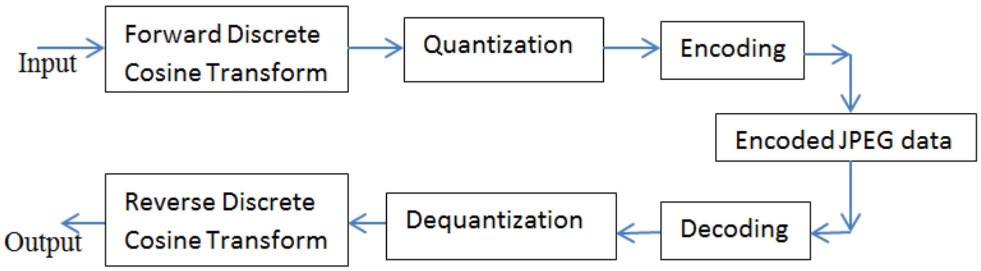

Figure 1: JPEG data compression technique

Let us assume that we have the following 8 x 8 input matrix as given in [9], [12].

$$Input = \begin{bmatrix} 52 & 55 & 61 & 66 & 70 & 61 & 64 & 73 \\ 63 & 59 & 55 & 90 & 109 & 85 & 69 & 72 \\ 62 & 59 & 68 & 113 & 144 & 104 & 66 & 73 \\ 63 & 58 & 71 & 122 & 154 & 106 & 70 & 69 \\ 67 & 61 & 68 & 104 & 126 & 88 & 68 & 70 \\ 79 & 65 & 60 & 70 & 77 & 68 & 58 & 75 \\ 85 & 71 & 64 & 59 & 55 & 61 & 65 & 83 \\ 87 & 79 & 69 & 68 & 65 & 76 & 78 & 94 \end{bmatrix}$$

After using the JPEG compression techniques mentioned in [9] and [12], we will get the following output matrix as given in [9] and [12].

$$Output=\begin{bmatrix} 58 & 64 & 67 & 64 & 59 & 62 & 70 & 78 \\ 56 & 55 & 67 & 89 & 98 & 88 & 74 & 69 \\ 60 & 50 & 70 & 119 & 141 & 116 & 80 & 64 \\ 69 & 51 & 71 & 128 & 149 & 115 & 77 & 68 \\ 74 & 53 & 64 & 105 & 115 & 84 & 65 & 72 \\ 76 & 57 & 56 & 74 & 75 & 57 & 57 & 74 \\ 83 & 69 & 59 & 60 & 61 & 61 & 67 & 78 \\ 93 & 81 & 67 & 62 & 69 & 80 & 84 & 84 \end{bmatrix}$$

Now, the error matrix based on Eq. (1) is as follows:

$$Error=\begin{bmatrix} -6 & -9 & -6 & 2 & 11 & -1 & -6 & -5 \\ 7 & 4 & -1 & 1 & 11 & -3 & -5 & 3 \\ 2 & 9 & -2 & -6 & -3 & -12 & -14 & 9 \\ -6 & 7 & 0 & -4 & -5 & -9 & -7 & 1 \\ -7 & 8 & 4 & -1 & 1 & 4 & 3 & -2 \\ 3 & 8 & 4 & -4 & 2 & 11 & 1 & 1 \\ 2 & 2 & 5 & -1 & -6 & 0 & -2 & 5 \\ -6 & -2 & 2 & 6 & -4 & -4 & -6 & 10 \end{bmatrix}$$

Now, we compute M, which is the average error from Eq. (4). We chose 8*8 to reflect the dimension of the array, and since we are on the verge of calculating a group of average errors, we sub-numbered the M in each case to differentiate between them. The average error is:

$$M_1= \frac{1}{8} \cdot \frac{1}{8} \sum_{x=0}^{7} \sum_{y=0}^{7} | X_0 - X_1 |= \frac{310}{64} = 4.84375.$$

4.1.2 Watermark Approach

The watermark authentication approach is shown in Fig.2.

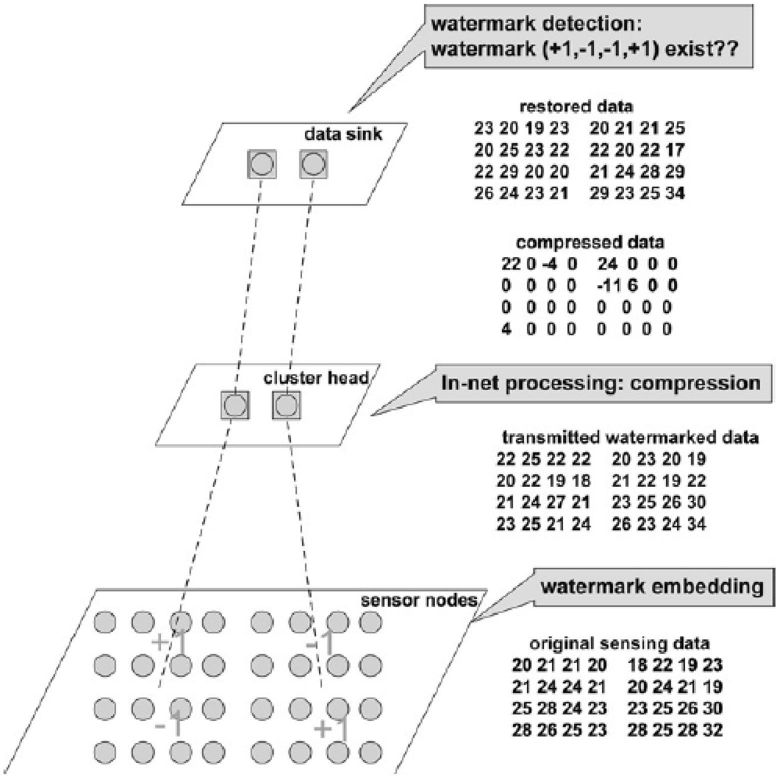

Figure 2: Watermarking technique [8]

The error matrix E=| X_0-X_1|is calculated by subtracting the restored data from the original sensing data of Fig. 2.

$$\begin{matrix} -3 & 1 & 2 & -3 & -2 & 1 & -2 & -2 \\ 1 & -1 & 1 & -1 & -2 & 4 & -1 & 2 \\ 3 & -1 & 4 & 3 & 2 & 1 & -2 & 1 \\ 2 & 2 & 2 & 2 & -1 & 2 & 3 & -2 \end{matrix}$$

The average error is:

$$M_2= \frac{1}{8} . \frac{1}{4} \sum_{x=0}^{7} \sum_{y=0}^{3} | X_0 - X_1 | = \frac{62}{32} = 1.9375.$$

By comparison, M_1>M_2,which means that there were fewer errors in the watermark approach than in the JPEG approach. The inequality in the number of data samples may be the reason for this margin in error measurement. Next, we will seeanother comparison where the same number of samples is compared to eliminate inequality from the calculation.

4.2 Equal Number of Samples

We assume the number of samples in each method is equal. In this scenario, we drop some samples into the JPEG approach for the sake of equality and fairness in the comparison.

4.2.1 JPEG Approach

In this case, we consider only the first 32 elements of the error array; in this case, there are 8 columns and 4 rows.

$$\begin{matrix} -6 & -9 & -6 & 2 & 11 & -1 & -6 & -5 \\ 7 & 4 & -1 & 1 & 11 & -3 & -5 & 3 \\ 2 & 9 & -2 & -6 & -3 & -12 & -14 & 9 \\ -6 & 7 & 0 & -4 & -12 & -9 & -7 & 1 \end{matrix}$$

We compute the average error M$_3$:as:

$$M_3 = \frac{1}{8} . \frac{1}{4} \sum_{x=0}^{7} \sum_{y=0}^{3} | X_0 - X_1 | = \frac{184}{32} = 5.75$$

After reducing the number of samples in the JPEG approach, we found $M_3 > M_2$,which means the result is in favor of the watermark approach. Under the same category of equal samples, different data were used in the two approaches. Similar data sources were used.

4.2.2 Applying JPEG on Watermark Data

We have only 32 elements in the watermark based sensor data. We added extra 32 elements with zeros to form an 8x8 matrix for JPEG operation. First 32 elements were collected from the original sensing data of Fig. 2.

$$\begin{matrix} 20 & 21 & 21 & 20 & 18 & 22 & 19 & 23 \\ 21 & 24 & 24 & 21 & 20 & 24 & 21 & 19 \\ 25 & 28 & 24 & 23 & 23 & 25 & 26 & 30 \\ 28 & 26 & 25 & 23 & 28 & 25 & 28 & 32 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix}$$

We also consider the gray levels are already subtracted. The two dimensional DCT computation of the above coefficient array can be done by using thefollowing equation:

$$G_{u,v}= \sum_{x=0}^{7} \sum_{y=0}^{7} \alpha(u)\alpha(v)g_{x,y} cos\left[\frac{\pi}{8}\left(x+\frac{1}{2}\right)u\right] cos\left[\frac{\pi}{8}\left(y+\frac{1}{2}\right)v\right]$$

After apply the DCT, the output array T (u,v) resulting from the application of the Matlab® function is as follows:

91.7500	0.9946	5.3519	-5.0868	1.7500	-1.4188	-2.7580	2.5975
79.2008	1.5400	3.5438	-4.2670	0.1933	-1.0152	-1.4815	0.9361
-8.8029	0.7675	-3.4911	1.8894	-3.7023	0.4686	3.5141	-4.0124
-34.3927	-1.8181	-5.2110	5.7755	-4.2086	0.3230	6.3907	-5.5238
3.0000	-3.8253	-0.6437	3.9651	-0.5000	-0.9071	4.3256	-2.3340
23.7825	-2.2670	3.9649	-1.8538	3.6373	-0.9951	-1.0810	1.8085
-2.3069	1.4897	4.2641	-6.4280	4.7807	0.3436	-5.2589	3.3341
-22.1054	2.7594	1.9987	-5.6114	2.9826	1.0238	-4.7030	2.1797

It is necessary to round the array elements of T (u,v) to the nearest integer.

92	1	5	−5	2	−1	−3	3
79	2	4	−4	0	−1	−2	1
−9	1	−4	9	−4	1	4	−4
−34	−2	−5	6	−4	0	6	−6
3	−4	−1	4	−1	−1	4	−2
−23	−2	4	−2	4	−1	−1	2
−2	2	4	−6	5	0	−5	3
−22	3	2	−6	3	1	−5	2

The next step is quantization. The coefficient array is ready to be quantized. A quantizing array Z (u,v) for the JPEG compression is defined as [9].

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

We must divide each element from T (u,v) by the corresponding element from Z (u,v) to obtain the quantized array $\hat{T}$ (u,v).

$$\hat{T}\ (u,v) = \text{round}\left[\frac{T(u,v)}{Z(u,v)}\right|$$

6	0	1	0	0	0	0	0
7	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
−2	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
−1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

At this encoding stage, a zigzag reordering of $\hat{T}$ (u,v) is [6 0 7 0 0 1 0 0 0 -2 0 0 0 0 0 0 0 0 0 0 -1 EOB].EOB means End of Blocks, where the rest of the blocks after it are all zeros.

The reverse transformation has to take place in order to retrieve the input data at the destination. The attention will be concentrated on the decoding, dequantization and the reverse of the coefficient array. Dequantization starts with the construction of the quantized array from the one-dimensional array, above which is another form of the two-dimensional quantized array.

The next step is to reconstruct the approximation of the coefficient array $\dot{T}$(u,v) according to the following equation:

$$\dot{T}(u,v) = \hat{T}(u, v)\ Z\ (u, v)$$

96	0	10	0	0	0	0	0
84	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
−28	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
−24	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

The last step in the reverse process is to reverse the approximation coefficient array $\dot{T}$(u,v).

21.7245	20.7678	19.4148	18.4581	18.4581	19.4148	20.7678	21.7245
31.1067	30.1500	28.7970	27.8403	27.8403	28.7970	30.1500	31.1067
25.9099	24.9532	23.6002	22.6435	22.6435	23.6002	24.9532	25.9099
15.7525	14.7957	13.4428	12.4860	12.4860	13.4428	14.7957	15.7525
11.5140	10.5572	9.2043	8.2475	8.2475	9.2043	10.5572	11.5140
1.3565	0.3998	-0.9532	-1.9099	-1.9099	-0.9532	0.3998	1.3565
-3.8403	-4.7970	-6.1500	-7.1067	-7.1067	-6.1500	-4.7970	-3.8403
5.5419	4.5852	3.2322	2.2755	2.2755	3.2322	4.5852	5.5419

This is the outcome of the Matlab® reverse transformation function. The results must now be rounded to the nearest integer.

22	21	19	19	19	19	21	22
31	30	29	28	28	29	30	31
26	25	24	23	23	24	25	26
16	15	13	13	13	13	15	16
12	11	9	8	8	9	11	12
1	0	−1	−2	−2	−1	0	1
−4	−5	−6	−7	−7	−6	−5	−4
6	5	3	2	2	3	5	6

The results shown above are the data retrieved from the source; it is an approximation (the original plus changes).

The error array is shown below. It is the difference between the original array and the retrieved array at the destination. We consider the absolute value for each element in the array.

2	0	2	1	1	3	2	1
10	6	5	7	8	5	9	12
1	3	0	0	0	1	1	14
12	11	12	10	15	12	13	16
12	11	9	8	8	9	11	12
1	0	1	2	2	1	0	1
4	5	6	7	7	6	5	4
6	5	3	2	2	3	5	6

After the array of differences has been obtained, we can compute the average error as follows.

$$M_4= \frac{1}{8} . \frac{1}{8} \sum_{x=0}^{7} \sum_{y=0}^{7} | X_0 - X_1 |= \frac{368}{64} = 5.7.$$

After calculating the average error for the four different cases, it is obvious that the results favor the watermark approach, but not enough to determine which approach is better in terms of less error measurement. Moreover, M_4 resulted in an answer that was not much different from the others. On the other hand, M_4 behaved differently than M_1. We also found that $M_4 > M_1$ although M_4 and M_1 are supposed to be almost the same.

5. CONCLUSIONS

We employed two compression algorithms in a wireless sensor network to determine if there were any significant changes in performance in terms of error measurement. We concluded that depending only on the error measurement to assess the performance of a network is not adequate. Our results led to small error margins and a slight preference in favor of the watermark approach. It is better to use a compression method that utilizes quantization methods that depend on K-largest coding, as suggested by Zhang, rather than the quantization methods used in the JPEG approach, which depend on the Human Visual System. The average error between the two approaches was very small. However, in WSN, algorithms with less error should be used in order to save energy and to conserve memory. In this paper, we explained the reasons for employing compression algorithms in WSN and how they contribute to the performance of the entire network. WSN is getting smaller and requiring less power than ever before, which demands more attention and careful choices of the type and size of software on the chip. The need for deploying of compression method in WSN is increasing due to the requirement of new application that process more information. In that sense, compatible compression methods are a necessity in WSN environments.

In the future, we will concentrate on quantization methods. The criteria will depend on the number of instructions and the error measurement, where the number of instructions is related to power consumption. This is a massive area of research. To enhance the security of WSN, it is required to look into transformation techniques to cover every aspect of lossy compression as well as other WSN compatible compression methods.

REFERENCES

[1] N. Kimura and S. Latifi, “A survey on data compression in wireless sensor networks”, Information Technology: Coding and Computing, 2005. International Conference on, volume: 2, pp.: 8-13, 4-6 April 2005.

[2] A. Vander Byl, R. Neilson, and R. Wilkinson, “An evaluation of compression techniques for wireless sensor networks”, IEEE African 2009, pp.: 1-6, 23-25 Sep. 2009.

[3] A. Deshpande, “Data compression in sensor networks”, Encyclopedia of Database Systems 2009, pp.: 564-565.

[4] M. Chen and M. Fowler, “Data compression trade-offs in sensor networks” Proc. SPIE 5561, Mathematics of Data/Image Coding, Compression, and Encryption VII, with Applications, October 18, 2004.

[5] S. Puthenpurayil, R. Gu, and S. Bhattacharyya, “Energy-aware data compression for wireless sensor networks”, Proceedings of the International Conference on Acoustics, Speech, and Signal Processing, volume: 2, pp.: 45-48, April 2007.

[6] G. Blelloch, “Introduction to Data Compression”, Carnegie Mellon University, Jan. 2013.

[7] I. Cox, M. Miller, and J. Bloom, “Digital Watermarking”, Chapter 1: Introduction, Morgan Kaufman, pp.: 9-11, 2002.

[8] W. Zhang, Y. Liu, S. Das, and P. De, “Secure data aggregation in wireless sensor networks: A watermark based authentication supportive approach”, Pervasive and Mobile computing, Elsevier, pp.: 659-680, May 2008.

[9] R. C. Gonalez and R. Woods, “Digital Image Processing”, Prentice Hall, 2002.

[10] S. Ozdemir and Y. Xiao, “Secure data aggregation in wireless sensor networks: A comprehensive overview”, Journal, Computer Networks: The International Journal of Computer and Telecommunications Networking, volume: 53, issue: 12, pp.: 2022-2037 March 2009.

[11] R. Vidhyapriya and P. Vanathi, “Energy efficient data compression in wireless sensor networks”, The International Arab Journal of Information Technology, volume: 6, number: 3, pp.: 297-303, July 2009.

[12] http://en.wikipedia.org/wiki/JPEG.

Authors

Mohammad Abdus Salamis an Associate Professor in the Department of Computer Science at Southern University, Baton Rouge, Louisiana. He received his BS degree in Electrical and Electronics Engineering from Bangladesh Institute of Technology, Rajshahi in 1991 and MS and Ph.D. degrees from Fukui University, Japan, respectively in 1998 and 2001. Prior to 2005, he worked as an adjunct facultymember of Mathematics and Computer Science at the City University of New York at York College, and as a postdoctoral fellow in the Department of Electrical and Computer Engineering at the University of South Alabama, Mobile, Alabama. He is a senior member of IEEE. His researchinterests include wireless communication, error-control coding, and sensor networks.

Hesham M. Elashhab earned a Master degree in Computer Science in 2013 from the Department of Computer Science at Southern University in Baton Rouge, Louisiana. He finished his Bachelor degree in Electrical Engineering from University of Benghazi in Libya in 2002 majoring in Telecommunications. In the period between 2003 and 2008, he worked in Sirte Oil Company, L ibya as Field Engineer and in AlmadarAljadidTelcom, also in Libya, as Operation and Maintenance Engineer. He also participated in cellular network extension projects. During his Master’s study, he participated in ongoing projects and conferences involving Wireless Sensor Networks and .NET programming.