J Inf Process Syst, Vol.12, No.2, pp.322~331, June 2016
http://dx.doi.org/10.3745/JIPS.03.0051

ISSN 1976-913X (Print)
ISSN 2092-805X (Electronic)

JOURNAL OF INFORMATION PROCESSING SYSTEMS JIPS

An Efficient Bit-Level Lossless Grayscale Image Compression Based on Adaptive Source Mapping

Ayman Al-Dmour*, Mohammed Abuhelaleh*, Ahmed Musa**, and Hasan Al-Shalabi***

Abstract

Image compression is an essential technique for saving time and storage space for the gigantic amount of data generated by images. This paper introduces an adaptive source-mapping scheme that greatly improves bit-level lossless grayscale image compression. In the proposed mapping scheme, the frequency of occurrence of each symbol in the original image is computed. According to their corresponding frequencies, these symbols are sorted in descending order. Based on this order, each symbol is replaced by an 8-bit weighted fixed-length code. This replacement will generate an equivalent binary source with an increased length of successive identical symbols (0s or 1s). Different experiments using Lempel-Ziv lossless image compression algorithms have been conducted on the generated binary source. Results show that the newly proposed mapping scheme achieves some dramatic improvements in regards to compression ratios.

Keywords

Bit-Level, Lempel-Ziv Coding, Lossless Image Compression, Source Encoding

1. Introduction

Nowadays, one can observe the massive amount of digital data generated by advanced technology and applications. This large amount of data is due mainly to image data. To handle this amount of data, many image compression techniques have been devised in the research. These techniques are classified as lossy and lossless, and it is known that lossy compression techniques might achieve high compression ratios [1]. However, this achievement usually affects the original image recreation process. On the other hand, lossless compression techniques retrieve the original image with acceptable compression ratios (CRs).

In general, there are three different approaches to compress image files. It is known that some pixel values are more common than others and the repetition of some information (redundancy) can be reduced. The first approach can reduce the coding redundancy based on the pixel values, such as Huffman coding and Lempel-Ziv-Welch (LZW) coding [2-4]. Coding redundancy reduces the number of source symbols by applying a source-mapping scheme. Furthermore, compression algorithms, such as Huffman, LZW, etc., are applied to achieve higher compression. Compared to other algorithms, Huffman algorithm generates minimum redundancy codes [5-7]. Huffman coding has been effectively

Manuscript received October 7, 2015; January 6, 2016.
Corresponding Author: Ayman Al-Dmour (ayman70jo@yahoo.com)
* School of Information Technology, Al-Hussein Bin Talal University, Ma'an, Jordan (ayman70jo@yahoo.com, mabuhela@my.bridgeport.edu)
** School of Engineering, Yarmouk University, Irbid, Jordan (as_shorman@yahoo.com)
*** School of Engineering, Al-Hussein Bin Talal University, Ma'an, Jordan (hmfnam@yahoo.com)

used in images, video compression, and conferencing systems. The second approach is called psychovisual redundancy, which is used when some color differences are imperceptible [8].

The last approach, reduce interpixel redundancy [8], deals with images based on neighboring pixel values. This brings up the question of 'how can the noise be tolerated?'. Three spatial methods have been developed to tolerate noise. The first method predicts the next pixel based on the previous one, which is called predictive coding. The second method describes the image in terms of recursive affine transformations. The last method transforms pixels to another domain, such as a frequency domain, in which the correlation between neighboring pixels is lower. Examples of these transforms are the discrete cosine transform, discrete Fourier transform, and the Karhunen-Loeve Transform [9].

Bit-level compression techniques have been proposed in several studies to accomplish enhanced text compression. In [10], the authors proposed a compression technique based on the bi-level technique to produce a dynamic and secure Arabic text compression using the bitwise Lempel-Ziv algorithm. In this research, the text file was prepossessed to produce a binary file by mapping a codeword to each character according to its appearance in the input file in order to reduce the number of the symbols in the file. The bitwise Lempel-Ziv algorithm then compressed the binary file in order to achieve a high CR. In [11,12], a modified compression technique is proposed for being applied to English text files. In [11], each character in the original file is assigned a weighted fix-length code to produce a binary file as an input to the LZ-78 algorithm for compression in order to produce a high CR with less complexity and less memory usage. In [12], an intermediate mapping scheme is used in which the original English text is first transformed to the decimal domain and then to the binary domain. Moreover, an efficient bitwise Huffman coding technique based on source mapping is proposed in [13].

On the other hand, some research has adopted a bit-level compression technique for image compression. A combination method for lossless image compression by applying the LZW code on the image and then processing the outputs using BCH error detection and correction algorithm in a loop ending with inflation detection in order to improve the CR without losing data [14]. In [15], the authors introduced a modification to the BCH encoder when converting an image, by dividing each part of the image into 7-block sizes divided into two parts—one for the image and one for the keys. These blocks are faded to the Huffman code in order to increase the CR without losing data. The authors in [16] proposed a mapping technique that is based on a new fixed-length code, instead of the ASCII code, before applying binary run-length compression techniques.

In this work, an efficient bit-level lossless gray-scale image compression technique is proposed. This proposed technique achieves a higher CR based on mapping the original image into a binary domain. Thus, the number of source symbols is significantly reduced. The mapping process (see Fig. 1) is achieved as follows: first, a pre-analysis of the original image is performed to count the frequency of occurrence of each symbol. Second, symbols are sorted in descending order based on this analysis. In turn, each symbol is replaced by an 8-bit unique weighted fixed-length code. Finally, the well-known compression algorithm, LZW, is applied to the generated binary source. This results in an improvement in CR while also recreating the exact original file.

The rest of the paper is organized as follows: in Section 2, an adaptive mapping scheme of images is presented in Sections 3 and 4 present the compression and decompression processes, which are mainly based on the bit-level LZW algorithm. Experimental results and the evaluation of compressing images based on an adaptive mapping scheme are discussed in Section 5. Finally, the conclusions are provided in Section 6.

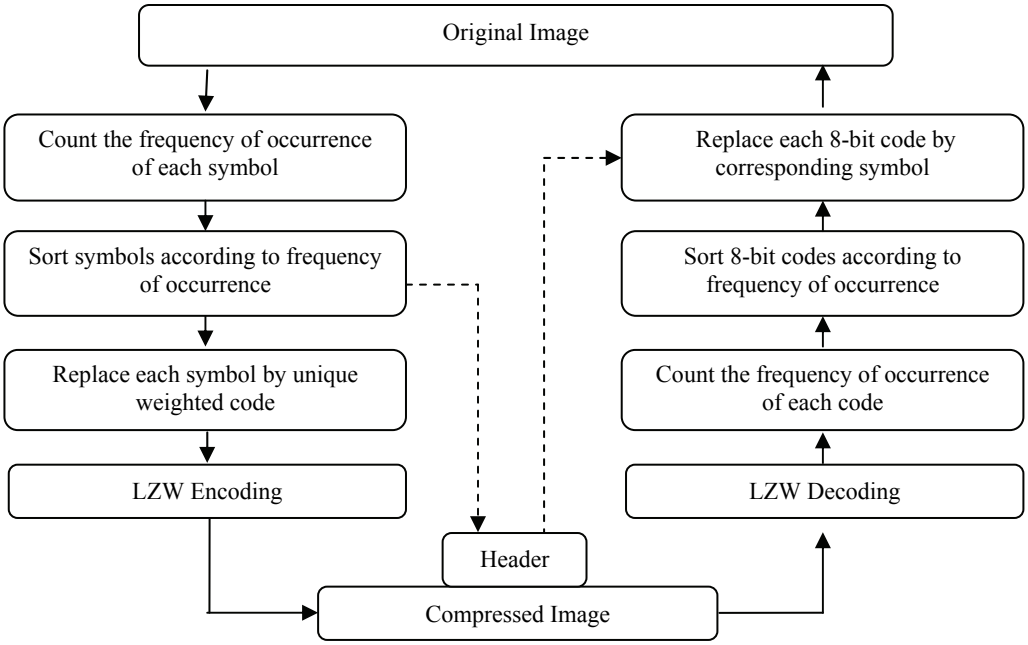

Fig. 1. Proposed compression technique.

2. Adaptive Mapping Scheme of Images

In general, symbol occurrences differ from one image to another [17]. In the source-mapping scheme, a statistical analysis is performed on a given image to determine the frequency of occurrence of each available symbol. In light of this analysis, all symbols are arranged from largest to smallest according to the number of times they appear in the image. Subsequently, each symbol is given an 8-bit weighted code in lieu of an arbitrary one (e.g., ASCII code). The weighted codes are assigned and ordered as follows: The code word with a Hamming weight of 8 (all ones 8-bit code) is assigned to the symbol that occurs the most frequently. In the same way, codes with a Hamming weight of 7 are assigned to the next eight symbols that are most probable. The assignment process is iterated until the code with a minimum Hamming weight is given to the symbol that occurs the least often.

According to the aforementioned assignment rule, a binary file B with a large number of ones compared to zeros is generated. As a result, the entropy of the generated binary file multiplied by the code length (N) is as close as possible to that of the original image I. Eq. (1) gives the difference in the entropy, ΔH. Therefore, the dynamic scheme will enhance the compression performance in comparison with other mapping schemes.

$$\Delta H = H(I) - NH(B) \tag{1}$$

Fig. 2 shows the matrices 1-a and 1-b of a given image. In the first matrix, one can see that a code '00001000' is given to symbol 8. This code is yielded from converting the symbol 8 from the decimal format to an equivalent binary format. However, if the dynamic mapping scheme is used, the code '11111111,' seen in matrix 1-b, is given to the symbol 8 as it appears the most often (i.e., six times).

Fig. 2 shows the frequency of both symbol one and symbol zero in both cases. From this figure, one can observe that symbol one has a higher probability of occurrence when a dynamic mapping scheme is used. Fig. 3 displays a complete adaptive source mapping scheme example.

$$\begin{bmatrix} 2 & 8 & 6 & 6 \\ 6 & 8 & 8 & 8 \\ 8 & 8 & 10 & 10 \\ 9 & 10 & 10 & 14 \end{bmatrix} \rightarrow \begin{bmatrix} 00000010 & 00001000 & 00000110 & 00000110 \\ 00000110 & 00001000 & 00001000 & 00001000 \\ 00001000 & 00001000 & 00001010 & 00001010 \\ 00001011 & 00001010 & 00001010 & 00001110 \end{bmatrix} \quad 1\text{-a}$$

$$\begin{bmatrix} 2 & 8 & 6 & 6 \\ 6 & 8 & 8 & 8 \\ 8 & 8 & 10 & 10 \\ 9 & 10 & 10 & 14 \end{bmatrix} \rightarrow \begin{bmatrix} 11111011 & 11111111 & 11111101 & 11111101 \\ 11111101 & 11111111 & 11111111 & 11111111 \\ 11111111 & 11111111 & 11111110 & 11111110 \\ 11110111 & 11111110 & 11111110 & 11110111 \end{bmatrix} \quad 1\text{-b}$$

Fig. 2. Symbols available in an image along with their: corresponding equivalent binary codes and dynamic codes.

$$\begin{bmatrix} 2 & 8 & 6 & 6 \\ 6 & 8 & 8 & 8 \\ 8 & 8 & 10 & 10 \\ 2 & 10 & 10 & 14 \end{bmatrix}$$

Symbol	Freq.
2	2
8	6
6	3
10	4
14	1

Symbol	Freq.
8	6
10	4
6	3
2	2
14	1

Symbol	Freq.
8	11111111
10	11111110
6	11111101
2	11111011
14	11110111

$$\begin{bmatrix} 11111011 & 11111111 & 11111101 & 11111101 \\ 11111101 & 11111111 & 11111111 & 11111111 \\ 11111111 & 11111111 & 11111110 & 11111110 \\ 11111011 & 11111110 & 11111110 & 11110111 \end{bmatrix}$$

Fig. 3. An adaptive source mapping scheme example.

A further step that goes beyond the source-mapping scheme is to apply the lossless compression techniques, LZW, to the resulting binary file. This binary file and the header file, which stores the symbols existing in the original source sorted in descending order, are sent to the receiver side. The binary file is used for the entire decoding process. Whereas, the header file is a key input to the demapping process, which is applied to the file that results from the decoding process. The demapping process works in the opposite way to the mapping one. The next section presents the encoding and decoding tasks of the proposed algorithm.

3. Compression Process

Further to the source-mapping scheme, the LZW algorithm has to apply Step 4 (shown in Fig. 4) onto the generated binary file. There are variants of compression algorithms presented in the literature.

These algorithms rely on data type to achieve optimal compression performance metrics, such as speed, compression ratio, complexity, etc. It is worth mentioning that Lempel-Ziv compression variants, such as LZ-77, LZ-78, LZW, etc., achieve high compression performance for a wide range of data types. In a fixed-length coding variant, the data stream is processed and stored in a dictionary.

Encoding Algorithm

1. Input the Original Image
2. Calculate frequency of occurrences for each symbol in the original image
3. Implement source symbols mapping
    I. Sort symbols in descending order depending on their frequencies
    II. Map each symbol to corresponding code
4. Implement compression algorithm
5. Output <compressed Image, ordered set of symbols>
6. End

Fig. 4. Encoding algorithm.

In this study, we apply LZW to the nth-order extension of the generated binary file. Needless to say, the number of symbols of the generated binary file is far less than that of the original image file. This eases the software and hardware implementations of the bit-level LZW algorithm.

The bitwise LZW algorithm processes the input binary stream to find a match between the strings in the file and the strings exist in a string-based dictionary, which has been previously constructed by the algorithm. This dictionary is filled with common subsequences based on the extension-order at which the binary file will be manipulated. For instance, the dictionary is pre-filled with 11, 10, 01, and 00 if an extension-order of two is used for manipulating a file.

4. Decompression Process

The decompression process is basically a reverse operation of the compression one (see Fig. 5). To successfully recreate the original image file, some parameters used in the compression process must be readily available. These parameters include all of the symbols found in the image sorted in the descending order of extension order (n), uniform block length (L), and the symbol-mapping table.

Decoding Algorithm

1. Input <compressed Image, ordered set of symbols>
2. Implement decompression algorithm
3. Implement source symbols de-mapping
    I. Sort codes in descending order depending on their frequencies
    II. Map each code to corresponding symbol
4. Output the Decompressed Image
5. End

Fig. 5. Decoding algorithm.

A bit-level LZW algorithm along with an image-mapping scheme escalates the compression/decompression complexity. This complexity arises by sharing the header file, which includes the symbols available in the original source sorted in descending order, between the compression and decompression processes. However, this extra complexity is ignored over the compression savings along with the simple implementation processes. Furthermore, one of the steganography techniques can be used to hide the key parameters. Thus, the compressed data is secured from malicious activities [18,19].

On the decompression side, a dictionary is also populated with the same subsequences used in the compression process. Subsequently, the binary data stream stored in the compressed file is parsed into a uniform block of L-length bits. The least n bits of each uniform block are put away. The remaining L-n bits provide the equivalent binary representation of the index of the root subsequence that matches the subsequence under consideration, excluding the innovation bits. Thus, the decoder simply appends the innovation bits to the obtained root subsequence. The simplicity in decoding the task is due to the fixed-length coding attribute of the bit-level LZW encoder.

Once the bit-level LZW decoding process is completed, the original generated binary file retrieves back. Consequently, the retrieved binary data are divided into 8-bit binary words. These binary words are then remapped into their corresponding symbols according to the received key parameters (i.e., symbol mapping table).

5. Experimental Results and Discussion

For evaluating the performance of the proposed algorithm, six different well-known grayscale test images were used (see Fig. 6). Furthermore, a MATLAB code was written to transform the original image file into a binary file based on an adaptive mapping scheme. In addition, another code was written to implement the byte-level and bit-level LZW algorithms.

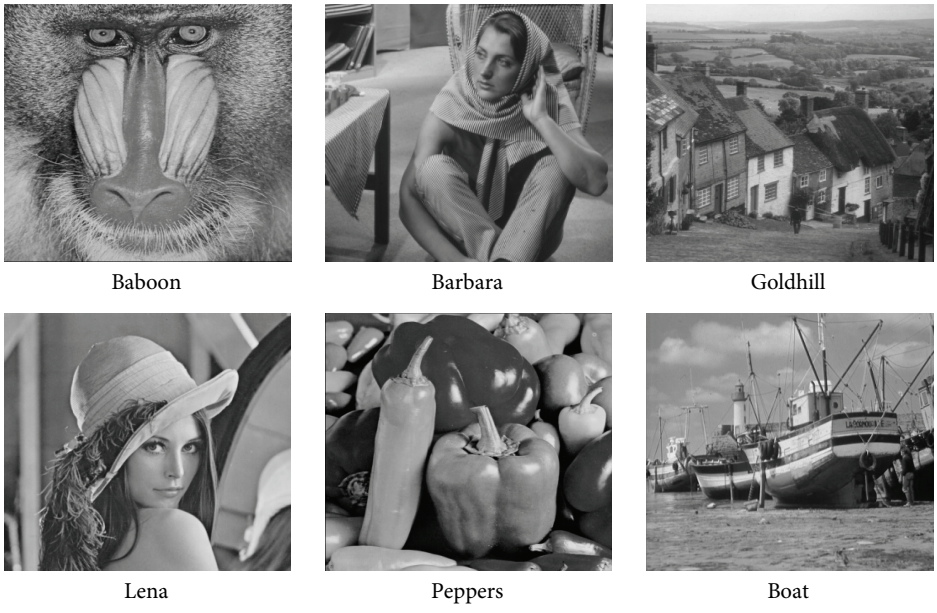

Fig. 6. Test images.

To illustrate the idea of the adaptive mapping scheme, the original image was transformed into binary files. These binary files were generated based on adaptive mapping schemes and the conventional ASCII encoding scheme. The number of 'ones' and 'zeros' were then computed and compared, as shown in Fig. 7. It is obvious that using the adaptive scheme generated a binary file with a large number of ones compared to zeros.

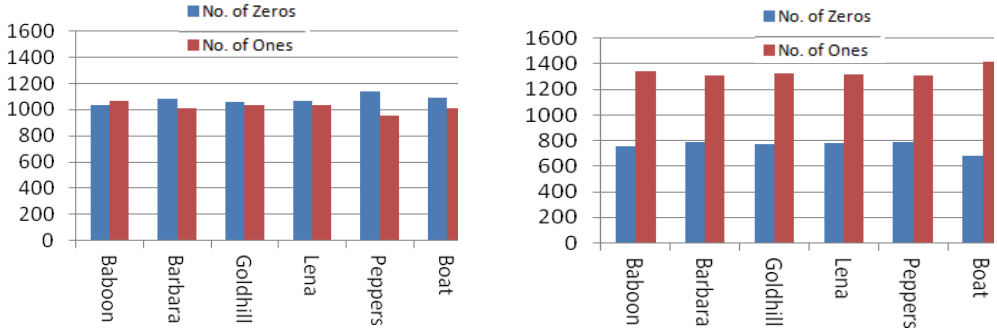

Fig. 7. Frequency of occurrences. (a) ASCII scheme and (b) adaptive scheme.

For evaluating the performance of the source adaptive scheme on a compression algorithm, binary files were generated based on adaptive mapping schemes and the conventional ASCII encoding scheme, which was compressed using the byte-level LZW algorithm. These files were then compressed using the bit-level LZW algorithm. Fig. 8 shows the compression ratio (e.g., CR = compressed image size / original image size) from which one can observe that compressing the image based on using an adaptive mapping scheme acquires some dramatic improvements in CRs.

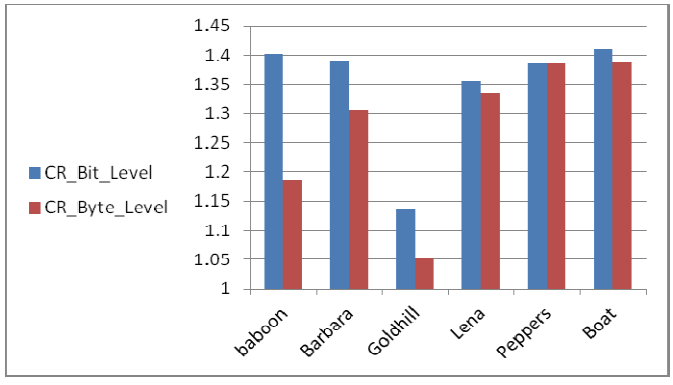

Fig. 8. Compression ratio comparison.

6. Conclusions

In this paper, we have presented an efficient bit-level lossless image compression method that utilizes the adaptive source mapping process before applying the LZW compression algorithm. The purpose of this mapping process was to convert an image file to a binary one, and at the same time, increase the

number of occurrences of symbols (i.e., ones) in the resulting binary file. Afterward, this binary file was compressed using the LZW algorithm. Experimental results showed that the CR for the bit-level compression technique that utilizes the adaptive encoding scheme outperformed the byte-level compression technique. It is worth mentioning that the required memory size, execution time, and hardware and software complications for compression/decompression techniques can be greatly reduced by using this type of equivalent binary source.

References

[1] R. C. Gonzales, R. E. Woods, and S. L. Eddins, *Digital Image Processing using MATLAB*. Upper Saddle River, NJ: Pearson Prentice Hall, 2003.

[2] D. A. Huffman, "A method for the construction of minimum-redundancy codes," *Proceedings of the IRE*, vol. 40, no. 9, pp. 1098-1101, 1952.

[3] J. Ziv and A. Lempel, "A universal algorithm for sequential data compression," *IEEE Transactions on Information Theory*, vol. 23, no. 3, pp. 337-343, 1977.

[4] J. Ziv and A. Lempel, "Compression of individual sequences via variable-rate coding," *IEEE Transactions on Information Theory*, vol. 24, no. 5, pp. 530-536, 1978.

[5] R. G. Gallager, "Variations on a theme by Huffman," *IEEE Transactions on Information Theory*, vol. 24, no. 6, pp. 668-674, 1978.

[6] D. E. Knuth, "Dynamic huffman coding," *Journal of Algorithms*, vol. 6, no. 2, pp. 163-180, 1985.

[7] J. S. Vitter, "Design and analysis of dynamic Huffman codes," *Journal of the ACM*, vol. 34, no. 4, pp. 825-845, 1987.

[8] H. Zha, "Progressive lossless image compression using image decomposition and context quantization," M.S. thesis, University of Waterloo, Waterloo, ON, 2007.

[9] M. Van der Schaar, Y. Chen, and H. Radha, "Embedded DCT and wavelet methods for scalable video: analysis and comparison," in *Proceedings of SPIE 3974: Visual Communications and Image Processing*. Bellingham, WA: International Society for Optics and Photonics, 2000.

[10] A. Musa, A. Al-Dmour, F. Fraij, O. Al-Khaleel, and M. Irshid, "A dynamic and secure Arabic text compression technique using bitwise Lempel-Ziv algorithm," *Information Technology Journal*, vol. 9, no. 4, pp. 673-679, 2010.

[11] A. Musa, A. Al-Dmour, O. Al-Khaleel, and M. Irshid, "An efficient compression technique using Lempel-Ziv algorithm based on dynamic source encoding scheme," *International Journal of Information and Communication Technology*, vol. 2, no. 3, pp. 210-219, 2010.

[12] A. Musa and A. Al-Dmour, "Enhancing text compression method using information source indexing," *Computer Engineering and Intelligent Systems*, vol. 5, no. 6, pp. 9-16, 2014.

[13] A. R. Elabdalla and M. I. Irshid, "An efficient bitwise Huffman coding technique based on source mapping," *Computers & Electrical Engineering*, vol. 27, no. 3, pp. 265-272, 2001.

[14] A. Alarabeyyat, S. Al-Hashemi, T. Khdour, M. H. Btoush, S. Bani-Ahmad, R. Al-Hashemi, and S. Bani-Ahmad, "Lossless image compression technique using combination methods," *Journal of Software Engineering and Applications*, vol. 5, no. 10, pp. 752-763, 2012.

[15] R. Al-Hashemi and I. W. Kamal, "A new lossless image compression technique based on Bose, Chandhuri and Hocquengham (BCH) codes," *International Journal of Software Engineering and Its Applications*, vol. 5, no. 3, pp. 15-22, 2011.

[16] A. R. Jaradat and M. I. Irshid, "A simple binary run-length compression technique for non-binary sources based on source mapping," *Active and Passive Electronic Components*, vol. 24, no. 4, pp. 211-221, 2001.

[17] M. Powell, "Evaluating lossless compression methods," Feb. 2001; http://corpus.canterbury.ac.nz/research/evaluate.pdf.
[18] P. Y. Chen and H. J. Lin, "A DWT based approach for image steganography," *International Journal of Applied Science and Engineering*, vol. 4, no. 3, pp. 275-290, 2006.
[19] N. I. Wu and M. S. Hwang, "Data hiding: current status and key issues," *IJ Network Security*, vol. 4, no. 1, pp. 1-9, 2007.

Ayman Al-Dmour http://orcid.org/0000-0002-9019-5379

He received his B.Sc. in Electronic–Communication Engineering in 1994 from Jordan University of Science and Technology, Irbid, Jordan. He pursued his M.Sc. and Ph.D. in 2003 and 2006, respectively, both in Computer Information Systems in the Arab Academy for Banking and Financial Sciences, Amman, Jordan. At Al-Hussein Bin Talal University (AHU), he has led the Department of Computer Information Systems, the Computer and Information Technology Center and the College of Information Technology. His research interests are in Arabic language processing, data compression and computer education.

Mohammed Abuhelaleh

He received the B.Sc. degree in Computer Science from Philadelphia University/Jordan in 1999, the M.S. and Ph.D. degrees in Computer Science and Engineering from University of Bridgeport, USA in 2007 and 2011, respectively. In March 2011, he has joined the Department of Computer Science in Alhussein Bin Talal University as an assistant professor. In September 2012, he was appointed the Chairman of Computer Science department for two years, then he was appointed the chairman of computer Information System department in addition to his position. In September 2014 he was appointed Associate Dean for the School of Information Technology at Alhussein Bin Talal University. In September 2015, he was appointed the Chairman of Software Engineering Department in addition to his current position.

Ahmed Musa

He was born on October 20, 1974 in Irbid, Jordan. He received a B.S. and M.S. in 1997 and 2000, respectively, both in Electrical Engineering from Jordan University of Science and Technology. He worked as a telecom engineer in Orange Jo from 1999-2001. In fall 2001, he joined the Ph.D. program in Computer Engineering at the University of Texas at El Paso, USA. In spring 2006, Dr. Musa worked as a full time associate professor at the faculty of Computer Engineering at Al-Hussein Bin Talal University (AHU), Ma'an, Jordan. In fall 2006, he is charged with leading the Department of Computer Engineering at AHU to the next level of distinction in research and teaching. In fall 2009, he was appointed as a vice dean, faculty of Engineering. Dr. Musa is currently an associate professor of Telecommunications Engineering at Yarmouk University, Irbid, Jordan. Dr. Musa main area of research interests includes optical fiber communications and networks, data compressions, computer networks and security, wireless communications and networks.

Hasan Al-Shalabi

He is a Full Professor of Computer Engineering at Al-Hussein Bin Talal University (AHU) in Jordan. He received his Ph.D. in Computer engineering in 1995 from the National Technical University of Ukraine, Kyiv, Ukraine. In the same year, he joined Al-Isra' Private University. In fall 1997, he joined the department of Computer Engineering at Princess Sumaya University for Technology, Amman, Jordan. He has led the Department of Computer Engineering and the Computer and Information Technology Center in 2003. He was the Dean of the College of Information Technology. In spring 2006. In fall 2013, he led a student's affairs deanship in Al-Hussein Bin Talal University Maan, Jordan. His research interests cover Computer Networks, Expert Systems, Software Agents, E-learning, Image processing, wireless sensor networks and Pattern Recognition. Currently, he is the vice president of Al-Hussein Bin Talal University (AHU) in Jordan.