Data compression and protection through representation with combinations and spiral path

Elda Cina[1], Hiba Tabbara[1], Esmerald Aliaj[2]
Department of Information Systems and Technology / Telecommunications and Networking Technology[1], Department of Informatics[2]
American University of Middle East[1], Athens University of Economics and Business[2]
Egaila[1], Athens[2]
Kuwait[1], Greece[2]
Elda.Cina@aum.edu.kw, Hiba.Tabbara@aum.edu.kw, Aliai16@aueb.gr

Received: October 12, 2020. Revised: November 20, 2020. Accepted: November 26, 2020.
Published: December 11, 2020.

Abstract: - We are living in Industry 4.0 era where enormous data need to be stored and processed. Though hardware is also becoming more abundant, its limitation in medium transmission speed and memory storage still beats this need. Therefore, data compression seems always an emerging necessity. Another big threat to data is also the unauthorized access especially while dealing with sensitive data. In this paper, we introduce an enhancement of a 2D signals compression with security tools through cryptography. Spiral path compression technique uses data representation through combinations to achieve satisfactory lossless compression rate but it also offers intrinsic encryption of data. Instead of representing an image as a matrix of pixels, we represent it through a group of index numbers, each belonging to a part of the image called mini-images. Every index is performed through a spiral path inside the mini-image starting from the most repeated pixel value. The histogram not only helps on defining these starting points of spirals but also decreases the number of bits needed to represent the index. Since there are many starting points possible for each mini-image, we use a random distribution to decide which of them to be selected. We also use a matrix of private keys to make possible the protection of the image from unauthorized use. We conclude that using this technique, we can achieve satisfactory compression rates compared to actual compression rates used nowadays and many other cryptographic possibilities are available for future studies.

Keywords: - Data compression, cryptography, histogram, combinations, random distribution, data security

I. INTRODUCTION

New technologies have improved the user's services significantly while processing enormous amount of data. We can easily see their implementation in Industry 4.0 [1], industrial internet of things [2], artificial intelligence [3], big data [4] and many other fields. However, the advancement of data storage technologies and network transmission media are still far away from the needs of such amount of data availability.

Therefore there is always need for techniques for data compression. There are different types of data in real world but all of them are represented as arrays of samples in the digital world: 1D (text, audio), 2D (images) and 3D (3D images) in 1D, 2D and 3D arrays of binary samples respectively.

Images are important data types not only for their relevance (transmitted television or satellite pictures, medical or computer storage pictures and much more) but especially because they tend to be larger and larger in size as per quality demands.

To minimize the size in images we could use two approaches: lossless compression and lossy compression. While lossless compression does not reduce the quality of the image and after reconstruction, we obtain a replica of the original image; lossy compression removes some of the redundancy to decrease the image size. This redundancy removal is not revertible. We sacrifice from the image quality but it is worthy compared with compression rates obtained. While lossless compression is a necessity in medical imaginary application or space research studies, lossy compression is widely used in general applications as in web applications photo and video cameras, video streaming. The lossy compression takes advantage of imperfection of visual system by removing what it is considered redundant or irrelevant [5].

In addition to compression of data, we need to secure the data while it is being transmitted. In industry 4.0, industrial devices are connected together through the internet and data is transmitted among them sometimes using untrusted connections and devices. This transmission needs to be secured especially in big data among manufactured parts [4].

The spiral path technique is pretty simple to implement and its compression rate is significantly better than the actual ones but what is important in this is its opportunity to implement also data protection features quite easily. With this paper, we are implementing a level of protection on images by modifying the Spiral path technique with data protection features. Images (2D signals) are taken into consideration, but it can be implemented in any electronic signal.

II. LITERATURE REVIEW

Images are compressed frequently for use in different contexts. Across the internet, in industry 4.0, images are transferred across different industrial systems all the time and need to be of acceptable sizes so that the transfer is efficient enough. Also, people save images on different platforms with limited storage sizes. We find that image compression is necessary and has been done for long time. Also, there has been some challenges in securing and protecting the images as types of data widely used across the network

Here, we do some review of existing image compression techniques as well as data encryption methods for industrial internet of things.

Two types of image compression exists: lossless and lossy. In lossless image compression, the image is compressed such that when it is uncompressed back, it will be exactly the same as the original image. In lossy image compression, the image is compressed such that some of its pixels values may be lost. In such case, the uncompressed image is not as precise as the original image. The advantage of this kind of compression is that the compression rate will be higher. Some applications can afford to have lossy compression hence making use of the resulting considerable savings in the sizes of the images.

The most common lossless image compression techniques are Hoffman coding [6], arithmetic coding [7] [8] and lossless predictive coding [9]. However, the compression rates in these techniques are low.

Their compression rate cannot exceed a compression factor of 2:1. The compression technique we are using in this research paper named Spiral Path offers a fixed compression ratio of 4:1 [10] lossless and 5.33:1 lossy [11].

There are various approaches to lossy image compression such as predictive coding, transform coding, vector quantization and neural networks.

In Predictive coding, one famous kind of prediction method is called differential pulse code modulation (DPCM). This method reproduces a pixel using the sum of the predicted pixel and a kind of error value known as the quantized error value [12] [13].

In transform coding, the basic transformations used are the Discrete Fourier Transform [14] [15], Discrete Walsh Hadamard Transform [16], Karhunen-Loeve Transform [17] and Discrete Cosine Transform [18].

In vector quantization, the compression technique uses a dictionary of patterns [19] [20].

Neural networks can be applied to the processing of visual information such as image compression in our case [21] [22] [23].

In terms of cyber physical systems in industry 4.0, in addition to compressing images, we need to have the data transmitted secure and reliable [24]. Different applications across the internet are communicating images in industries such as healthcare, environment, smart cities, and many others. A secure intelligent architecture for big data in healthcare industry 4.0 is proposed in [25].

III. DATA COMPRESSION THROUGH COMBINATIONS

A classic approach of images in computer is to represent them by a matrix of samples, where each sample is expressed by a sequence of bits. Therefore for any image, let's say gray image (each pixel is represented by 8 bits) of width W and height H we need $WxHx8$ bits to be stored in memory.

The "Data Representation through Combinations" [26] presents a different representation of images, a single number that represents the index of this image in the list of combination among all images of the same size. As such, every image can be represented by a file containing one numerical value.

If we consider the smallest set of images with 2 x 2 pixels (W=2 and H=2) and only 3 gray levels: Pixel 1 to Pixel 4 as pointed out by Table 1.

Table 1. Sample image with 4 pixels

Pixel[0]	Pixel[1]
Pixel[2]	Pixel[3]

All possible combinations of gray levels in that image are listed in Table 2.

Table 2. All combination, 4 pixels and 3 gray levels

Combination	**s[1]**	**s[2]**	**s[3]**	**s[4]**
0	0	0	0	0
1	0	0	0	1
2	0	0	0	2
3	0	0	1	0
...	...	...	...	...
7	**0**	**0**	**2**	**1**
...	....	....	....	....
10	0	1	0	1
11	0	1	0	2
....	....	....	....	....
18	0	2	0	0
...	....	....	....	....
27	1	0	0	0
...	....	....	....	....
54	2	0	0	0
....	....	....	....	.....
60	2	0	2	0
....	....	....	....	.....
70	2	1	2	1
....	....	....	....	.....
76	2	2	1	1
77	2	2	1	2
78	2	2	2	0
79	2	2	2	1
80	**2**	**2**	**2**	**2**

If we use the classical approach to store the image in the memory, we would always need:

4 pixels x 8 bits/pixel = 32 bits

However, if we use the index representation of the corresponding image from the above table, we can face many different scenarios as per the position where the image is ordered.

For example, if we have the image with index 7 we need only 3 bits to represent it instead of 8 bits (4 pixel x 2 bits per value) of the conventional case. Adding also the info that this image has only 3 gray levels (2 bits for it), the result is of size only 5 bits instead of 8. This leads to a compression ratio of 8/5 or 1.6:1. The last ordered image with index 80 needs at most 7 bits.

In general, we know that images are represented with a depth of 8 bits per color leading to 256 levels. Each pixel may have a value between 0 and 255=L-1 (where L expresses the color levels). The number of combinations is $L^{WxH} = 256^{2x2} = 2^{32}$, which means that the index of any image is between 0 and 2^{32}-1. To store this index in a file, we need between 0 and 32 bits (where 32 bits is the worst case). The lower the index is, the lower the number of bits is needed to store the image.

Therefore, the compression ratio varies from (example: combination # comb-1)

$$min = \frac{M \times N \times roundup(log_2(L))}{roundup(M \times N \times log_2(L))} : 1 \qquad (1)$$

to (combinations 0 and 1):

$$max = M \times N \times roundup(log_2(L)) : 1 \qquad (2)$$

Thus, depending on the combination, the compression ratio may be low (min) or very high (max).

For more details, you can refer to the original paper [26].

If we extract the histogram of the image, we can improve further the numbers in other approaches. First, having repeated pixels brings fewer combinations which leads to fewer bits. Also, an image might not have all grey levels therefore and we do not need to include all combinations [27].

Theoretically, the data representation through combinations seems an excellent opportunity on image compression. However, the nowadays images tend to be big in terms of MB, and since the index number tends to increase exponentially, it cannot be processed due to hardware limitations. To give life to this theory we must divide the image in smaller chunks and find the index of each piece.

IV. SPIRAL PATH

Any signal despite in how many dimensions we perceive it, in computer memory it is saved as a 1D signal in binary format. For images, the conventional approach is to place the image rows one after the other. In spiral path, a different way of transformation is used, moving through a spiral path over the image pixels as shown in Fig. 1 and Fig. 2.

100	100	100	100
123	123	123	115
123	127	129	114
150	123	144	140
152	152	149	136

Fig. 1. Gray image with size (5 * 4)

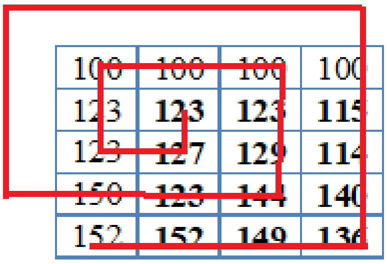


Fig. 2. Storing image in a spiral path

According to the idea of variability or "The Principle of Image Compression" [28] which is based on real images behavior, neighbor pixels tend to have similar color (gray level). By adopting this idea, we can take advantage of both, histogram and variability methods. Spiral path starts from the highest repeated pixel and continues as per a spiral to form a 1D stream of pixels with middle variability. After the spiral path, a second transformation is done to the original idea of data representation through combinations. The combination table is organized starting not from 0 but from middle variable image. The middle variable image is the image starting from the most frequent gray level. The pixels differ from the first pixels as shown in Fig 3.

As per these modifications, it has been proved that the image may not be arranged in the beginning of the table of combinations but obviously near it.

Since the start point is the most frequent pixel, it has been concluded that the best option to obtain the highest compression rate is to start with the pixel which leads the image with the lowest variability.

$$f(x) = \sum_{i=1}^{N} \left(\frac{|a(i) - a(i-1)|}{N} \right) = min \qquad (3)$$

For more details on this method or how to get better compression rate rearranging indexes, you can refer to the original paper.

Spiral path technique obtains a compression rate of 4:1 without loss. In the lossy version where we scarify two last significant biplanes, a compression ratio of 5.33:1 is achieved. The loss is insignificant with PSNR greater than 42.69 and similarity index SSMI index over 98%.

123	124	121	125	119	127	117	129	115	131	113	133	111	135	109	137	107	139	105	141

Fig. 3. Middle variable image

V. PROPOSED APPROACH: ENCRYPTED SPIRAL PATH

In the above sections, we mentioned that the start point of the spiral path is the most frequent pixel and to optimize the compression rate we use the one which builds the lowest variable image. However, if we want to enhance this technique with security features, we must give up on some optimization.

Many different ideas may be implemented to enhance the technique with protection features. In this paper, we have decided to use a random distribution function on the selection of the starting pixel of the spiral path.

The starting points' positions of each mini image are saved into a second matrix of private keys. The spiral path technique is used in both matrixes, the image and the keys one used for optimization.

The compressed image will have the size (# of blocks, (size of a block)^2) associated with a second matrix of keys (# of blocks, 2). The usage of random distribution and private keys will increase the total size of the image. However, this increase does not affect much the compression ratio. Still, the compression ratio achieved (3.6: 1 for lossless compression) is quite satisfactory compared with the traditional techniques.

A. Spiral path with cryptography algorithm

To enhance the spiral path with security layer we need to modify the algorithm. The new algorithm is as follows:

Compression Algorithm:

1. Divide into blocks (for each block):
   a. Find the start value through histogram
   b. Apply random function to select the starting pixel for each bock
   c. Walk spiral path starting form "Start value"
   d. Build the middle variable image
   e. Calculate the index according to the histogram properties
2. Save the keys by implementing the same steps
3. Assemble the indexes + keys

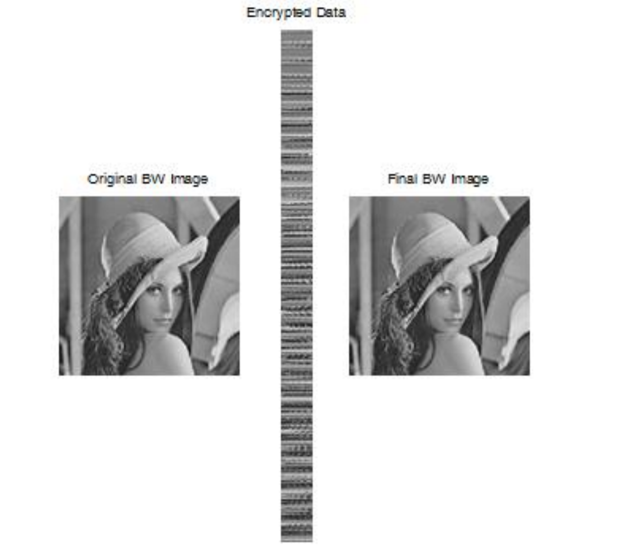
Decompression Algorithm:

1. Split the blocks by the keys
2. Divide the indexes
3. Build the blocks from the spiral
4. Assemble the original image.

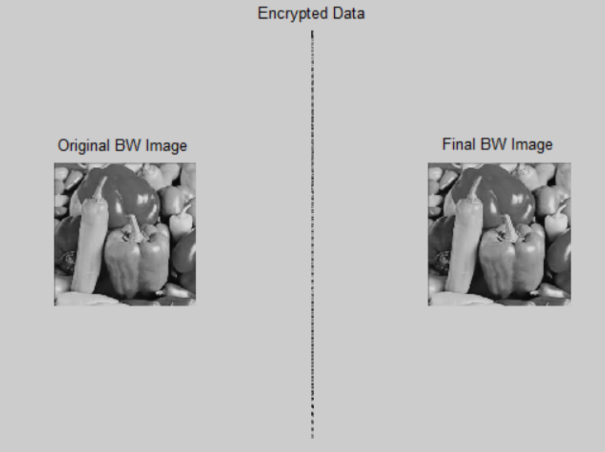
B. Sample Output

Table 3 presents some examples of images output of the implemented software, showing the comparison parameters between spiral path with and without encryption. The images are illustrated according to the following: Image 1, Original image before compression. Image two, Decompressed image without knowing the encryption keys. Image 3, decompressed images if correct keys are used.

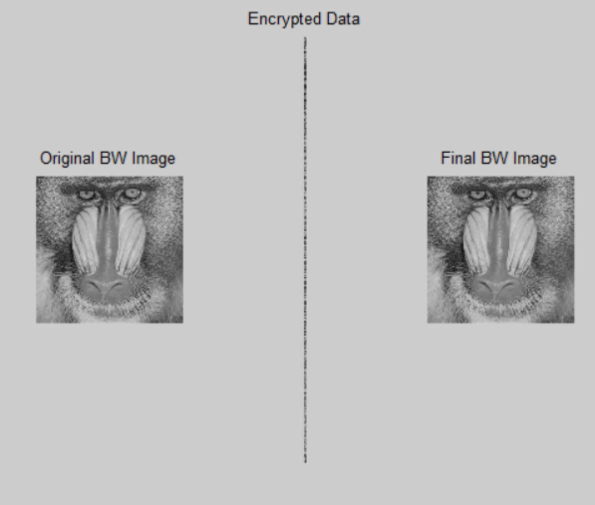
Table 3. Comparison Parameters Between Spiral Path with and without Encryption



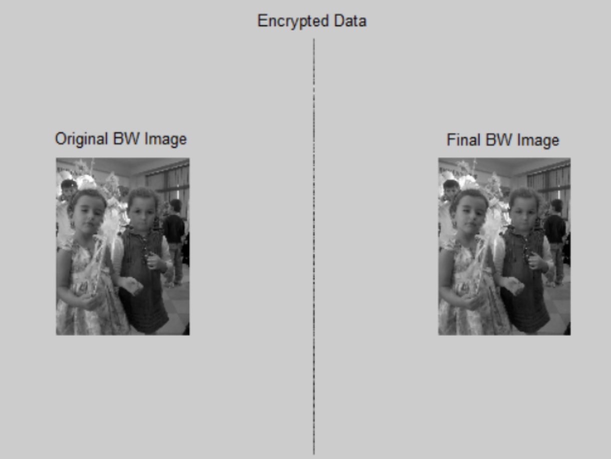
Comparison parameters	Spiral path	Spiral with encryption
Compression ratio	4:1	3.6:1
Time of execution	14s	8s



Comparison parameters	Spiral path	Spiral with encryption
Compression ratio	4:1	3.6:1
Time of execution	7s	6 s



Comparison parameters	Spiral path	Spiral with encryption
Compression ratio	4:1	3.6:1
Time of execution	15s	8s



Comparison parameters	Spiral path	Spiral with encryption
Compression ratio	4:1	3.6:1
Time of execution	339s	312 s

VI. ANALYSIS

From Table 3, it can be noticed that the algorithm used in this research paper is simpler in implementation than the original one since there is no need for extra calculation for the starting point. It leads to a faster execution time and it enhances the original technique with a security layer. The encryption layer can be implemented in both versions of Spiral path, lossless and lossy. Here, we analyze only the lossless version.

The increase of the header caused by the encryption keys decreases the corresponding compression rates with only 10% from the optimized version, which means from 4 to 3.6 for lossless compression and from 5.33 to 4.8.

We should not forget that even the lossy version of Spiral path is almost lossless with PSNR greater than 42.69 and similarity index SSMI index over 98%, which leads to quite satisfactory trade off in terms of robustness.

To check the position of this technique along with the well-known standards, we have taken into consideration the data from [29] and derived Table 4.

The squeeze chart library has lastly been updated in October 31 of 2018. It is a rich lossless file compression benchmark. Comparison has been made with the image data type sheet. Table 4.1 is an illustration of RGB data table.

The top 6 compressors are listed among 156 supported. Since the list includes multiple versions on the same compressor, only the version with the highest compression rate is taken into consideration for comparison.

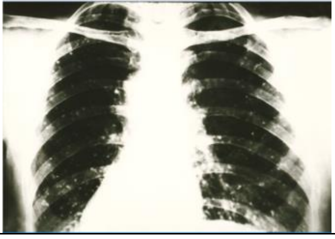
Table 4. Spiral path with encryption in regards to other techniques

Ranking	**Olympus** 	**CR**
1	***Spiral with encryption***	***3.6***
2	PAQ8PXD_v34	3.23
3	EMMA 0.1.21 [x64]	3.18
4	ZPAQ 6.36	3.16
5	FP8_v6	3.12
6	BMF 2.01	3.02

Ranking	**Fuji** 	**CR**
1	Spiral with encryption	**3.6**
2	EMMA 0.1.25	**2.67**
3	PAQ8PX_v120	**2.67**
4	ZPAQ 6.36	**2.61**
5	STUFFIT API 2.0.672.3563	**2.55**
6	FP8_v6	**2.54**

Ranking	**Sigma** 	**CR**
1	BMF 2.01	**4.03**
2	GraLIC 1.11	**3.93**
3	PAQ8PX_v120 -8	**3.89**
4	EMMA 0.1.25	**3.89**
5	ZPAQ 6.36	**3.81**
6	FP8_v6	**3.67**

Ranking	**Canon** 	**CR**
1	***Spiral with encryption***	***3.6***
2	PAQ8PXD_v34	2.92
3	EMMA 0.1.21 [x64]	2.85
4	FP8_v6	2.83
5	.DNG/lossless-JPEG	2.8
6	BMF 2.01	2.78

Ranking	**Radiograph** 	**CR**
1	PAQ8PXD_v34	6.70
2	EMMA 0.1.24	6.60
3	EMMA 0.1.4 fast 256MB	6.57
4	ZPAQ 6.36	6.47
5	FP8_v3 -7	6.47
6	FLIF 0.2.0rc16 [x64]	6.24

Ranking	**Humble** 	**CR**
1	***Spiral with encryption***	***3.6***
2	BMF 2.01	1.71
3	EMMA 0.1.25	1.68
4	GraLIC 1.11 demo	1.68
5	ZPAQ 6.36	1.67
6	PAQ8PXD_v34	1.67

Ranking	**Sony** 	CR
1	***Spiral with encryption***	***3.6***
2	PAQ8PX_v120 -8	2.84
3	RAW Image Format (.TIFF)	2.82
4	EMMA 0.1.21 [x64]	2.81
5	BMF 2.01	2.81
6	ZPAQ 6.36	2.80

It can be easily noticed that for *Canon, Fuji, Humble, Olympus and Sony*, the Spiral with encryption leads the list of compressors significantly. It is quite worse than many techniques for *Radiograph* where it holds 54th place and is ranked the 9th for *Sigma*.

As per other image compression tables:

1. **Artificial images** – the spiral with encryption is not quite satisfactory compared with existing techniques since they are implementations of mathematical patterns. It is better only in 2 out of 4 samples given in the list.
2. **Photographs, 8-bit Grayscale** – the spiral with encryption holds first place for MRI *Ultrasound*, and *Foveon X*. It holds the 4th place for *Electron Microscope blood cells* and the 55th place for *catheter*.
3. **Textures -** the spiral with encryption holds first place for *texture 1, 4, 5, 6, 7* and it holds 8th place in *texture 2,* and 9th place for *texture 3*.

To conclude the analysis, we can say that Spiral path with encryption offers very satisfactory results in terms of compression rate. Furthermore, a favoring factor for its advancement from other techniques is the added security layer.

We need to also mention that the time of execution is not taken into consideration for the comparison since there is no such data present in the library.

VII. CONCLUSION

Any In Industry 4.0 era, we are facing enormous data processing and transmission. There is always necessity on new techniques for compression, and more for data protection in the different fields employed in the Industry such as Artificial Intelligence, Big Data, Internet of Things and Deep Learning. In this paper, we proposed an enhancement of Spiral path compression technique with a cryptographic layer of security by using random distribution and private keys. The 2D signals have been taken into consideration for the experiments performed. As per results shown, we highlight that using Spiral path with encryption offers satisfactory compression rate compared to other methods. The proposed technique also increases the protection level of the data which is considered a major need in nowadays security requirements. One concern that can be noticed is the time of compression when images tend to be large. This is to be considered for future work.

References:

[1] J. Wan *et al.*, "Software-Defined Industrial Internet of Things in the Context of Industry 4.0," in *IEEE Sensors Journal*, vol. 16, no. 20, pp. 7373-7380, 2016.

[2] D Serpanos, M Wolf, “Industrial Internet of Things”. In: Internet-of-Things (IoT) Systems, Springer, Cham, pp. 37-54, 2018.

[3] Fei Tao, Qinglin Qi, Ang Liu, Andrew Kusiak, “Data-driven smart manufacturing”, Journal of Manufacturing Systems, Volume 48, Part C, 2018

[4] A. Adhikari, A. Hojjati, J. Shen, J. Hsu, W. King and M. Winslett, “Trust issues for big data about high-value manufactured parts”. IEEE International Conference on High Performance and Smart Computing (HPSC) and IEEE International Conference on Intelligent Data and Security (IDS), pp. 24-29, 2016.

[5] ProDavid Salomon, Giovanni Motta, Handbook of DataCompression, Handbook of Data Compression, Springer-Verlag London Limited 2010 ISBN 978-1-84882-902-2 page 448.

[6] D. A. Huffman, “A method for the construction of minimum redundancy codes”, Proc. IRE, No. 40, pp. 1098-1101, Sep. 1952.

[7] R. J. Clarke. Digital Compression of Still Images and Video. Academic Press, 2nd. Ed., London, 1996.

[8] G. Glen and J. Langdon, “An Introduction to arithmetic coding,” IBM J. Res. Develop., Vol. 28, No. 2, pp. 135-149, 1984.

[9] Gonzalez, R. C and R. E. Woods. (1992). Digital image processing, Addison-Wesley, Reading, pp. 307-411.

[10] Elda Cina, Esmerald Aliaj, Habib Hamam, “Compressing 2D Images Through Data Representation Using Spiral Path”, International Conference Information Systems and Technology Innovations, 6-7 June 2014

[11] Elda Cina, Esmerald Aliaj, Habib Hamam, “Increasing Compression Rate of Images Through Data Representation using Spiral Path and Biplanes”, International Journal of
Engineering Research & Technology, Volume. 3, Issue. 07 July – 2014 2278-0181

[12] A. Habibi, “Compression of nth-order DPCM Encoder with Linear Transformations and Block Quantisation Techniques”, IEEE Trans. Commun., Vol. COM-19, No. 6, pp. 948-956, 1971.

[13] A. K. Jain. Fundamentals of Digital Image Processing. Prentice Hall,Inc.,1989.

[14] H. C. Andrews and W. K. Pratt, “Fourier Transform Coding of Images”, Hawaii Int. Conf. Sys. Sci., Western periodicals Company, North Hollywood, California, pp. 677-679, 1968.

[15] G. B. Anderson and T. S. Huang, “Piecewise Fourier Transformation for Picture Bandwidth Compression”, IEEE Transactions on Communication Technol., Vol. 19, No. 2, pp.133-140, 1971.

[16] W. K. Pratt, J. Kane and H. C. Anderws, “Hadamard Transfer Image Coding”, IEEE Proceedings, Vol. 57, No. 1, pp. 58-68, 1969.

[17] P. A. Wintz, “Transform Picture Coding”, Proceedings IEEE, Vol. 60, pp. 809-820, July 1972.

[18] N. Ahmed, T. Natarajan, and K. R. Rao, “Discrete Cosine Transform”, In: IEE Transactions on Computers, Vol. 23, pp. 90-93, 1974.

[19] R. M. Gray, “Vector quantization,” IEEE ASSP Magazine. Vol. 1, pp. 4-29, April 1984.

[20] K. Sayood. Introduction to Data Compression. Academic Press, 2nd Edition, 2000.

[21] Sonehara, N., M. Kawato, S. Miyake, and K. Nakane, “Image data compression using neural network model,” IEEE proceedings of the international joint conference on neural networks, Washington DC, pp. 35-41, 1989.

[22] Dony, R. D. and S. Haykin, “Neural network approaches to image compression,” Proceedings of the IEEE, 83(2), pp. 288-303, 1995.

[23] Cramer, C. “Neural networks for image and video compression: A review,” European journal of operational research, 108, pp. 266-282, 1998.

[24] N. Jazdi, “Cyber physical systems in the context of Industry 4.0”, IEEE International Conference on Automation, Quality and Testing, Robotics, Cluj-Napoca, pp. 1-4, 2014.

[25] Manogaran G., Thota C., Lopez D., Sundarasekar R, “Big Data Security Intelligence for Healthcare Industry 4.0”. In: Thames L., Schaefer D. (eds) Cybersecurity for Industry 4.0. Springer Series in Advanced Manufacturing, Springer, Cham, pp. 103-126, 2017.

[26] H. Hamam, “A new representation of data through combinations”, ICCAT 2013,

[27] J. Said, R. Souissi, H. Hamam “A New Representation of Image Through Numbering Pixel Combinations” Journal of Information Security Research Volume: 4 , Issue: 1 (March 2013) Page: 46-51

[28] E.Cina, H.Hamam, “Increasing Compression Rate Of Images Through The Method Of Numbering Combinations” “Information Systems and Technology Innovation: toward a digital Economy

[29] Squeeze chart 2018: https://www.squeezechart.com

Contribution of individual authors to the creation of a scientific article (ghostwriting policy)

Elda Cina carried out: the Conceptualization, Data curation, Formal analysis, Funding acquisition, Investigation, Methodology, Resources, Supervision Validation, Visualization, Writing - original draft

Hiba Tabbara carried out: Resources, Investigation, Visualization, Writing - review & editing

Esmerald Aliaj carried out: Data curation, software development, validation.

Sources of funding for research presented in a scientific article or scientific article itself

The participation in the conference has been fully founded by American University of Middle East.

Creative Commons Attribution License 4.0 (Attribution 4.0 International, CC BY 4.0)