International Journal of Research in Advanced Engineering and Technology
ISSN: 2455-0876; Impact Factor: RJIF 5.44
Received: 23-11-2018; Accepted: 26-12-2018
www.newengineeringjournal.in
Volume 5; Issue 1; January 2019; Page No. 41-48

A comparative analysis of Huffman and LZW methods of color image compression-decompression

Mohd. Zoghoul[1], Hussein Hatamleh[2], Ziad Alqadi[3], Muhammed Mesleh[4], Belal Ayyoub[5]
[1, 3, 4, 5] Department of Computer Engineering, Al Balqa'a Applied University, Amman, Jordan
[2] Department of Mechatronics Engineering, Al Balqa'a Applied University, Amman, Jordan

Abstract
Data compression is a size reduction of data to be sent via network or to be stored on auxiliary storage for long time, thus data compression will save storage capacity, speed up file transfer, speedup data transmission by decreasing transferring time, and decrease costs for storage hardware and network bandwidth.
In this paper we will investigate Huffman and LZW methods of data compression-decompression. Different images in sizes and types will treated, compression, decompression times will be evaluated, compression ratio will obtained, the obtained results will be analyzed in order to do some judgments.

Keywords: compression, decompression, CR, CT, DT, throughput, speed up

1. Introduction

Data compression is a size reduction of data to be sent via network or to be stored on auxiliary storage for long time [1-60], thus data compression will save storage capacity, speed up file transfer, speedup data transmission by decreasing transferring time, and decrease costs for storage hardware and network bandwidth.
Data compression is very important and valuable task because of the following reasons:

- Data compression can dramatically decrease storage space required to store a file. For example, in a 3:1 compression ratio [11], a 30 megabyte (MB) file takes up 10 MB of space. As a result of compression, administrators spend less money and less time on storage.
- Data compression optimizes backup storage performance and has recently shown up in primary storage data reduction. Compression will be an important method of data reduction as data continues to grow exponentially.
- Data compression will improve the network bandwidth without paying extra costs for network enhancement
- Any type of file can be compressed; this will cover text file, digital images with small sizes and huge sizes, audio files and so on [6, 7].

Data compressing data can be a lossless or lossy task. Lossless data compression enables getting back or restoration of a compressed file to its original continents, without losing any piece of data, when the file is uncompressed [8, 9]. Lossless data compression-decompression is the typical approach with executables, as well as text and spreadsheet files, where the loss of words or numbers would change the information. Lossy data compression permanently eliminates redundant bits or bytes from the data, unimportant or imperceptible. Lossy data compression is useful with images, audio, graphics, video, where the removal of some data has little or no discernible effect on the representation of the original data contents [10]. Figure 1 shows the differences between these tasks:

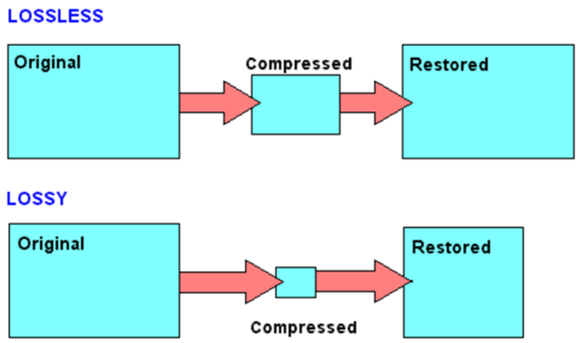

Fig 1: Lossless and lossy compression-decompression

Digital color images [12, 13, 14] are sometime require a lossless method for compression-decompression because they may be a covering image which contains a hidden secret data [15, 16], and here we must seek an efficient method of data compression-decompression [17, 18, 19] and here efficient mean [17]:

- Maximizing compression ratio (CR).
- Minimizing compression time (CT).
- Minimizing decompression time (DT).

2. Huffman compression-decompression method

Huffman method is one of the lossless popular methods used for data compression- decompression [20, 21]. Huffman compression task can be implemented applying the following steps:

1. Get the original input data.
2. Reshape the input data into one row.
3. Get the frequency (repetition) of each value in the input data, building a table for each value and the total repetition.
4. Sort the table according to repetition in descending way.

5. Use the sorted table to build a Huffman binary tree.
6. Decode the binary tree (left = 0, right=1).
7. Traverse binary tree to get the code for each value.
8. Use the codes to convert the input file to compressed file.

Here we will use these steps to compress the following data:
AABBBBBBBCCCDDDDDDDEFDCCCCDDDDDDDBBBBBBBBBBFFFFFFFFFFEAAABBBBBBBBEED

Figure 2 shows the Huffman tree for this file:

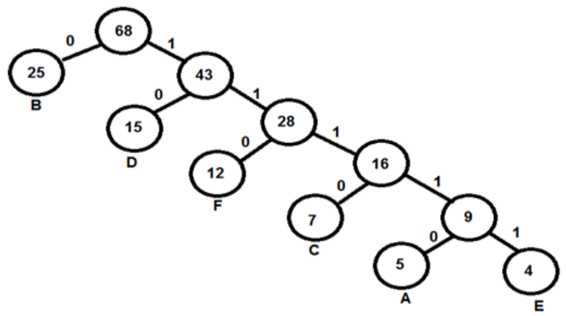

Fig 2: Huffman tree (example)

The Huffman tree here was built from left to right placing the mostly repeated value in the left side.
Using this tree we can get the code as shown in table 1:

Table 1: values coding (example)

Character	Frequency	Sorted	Frequency	Code	Compressed size(bit)
A	5	B	25	0	25
B	25	D	15	10	30
C	7	F	12	110	36
D	15	C	7	1110	28
E	4	A	5	11110	25
F	12	E	4	11111	20
Compressed file size (summation)					164
Compression ratio = 68*8/164					3.3171

From table 1 we can see that compressed file size is 164 bits and CR=3.3171, which means that we decrease the file size 3.3171 times and the compressed file is a stream of binary numbers as follows:

1111011110000000001110111011101010………………

Different ways are used to create Huffman binary tree, and the selected way can affect the compression ratio as shown in the following solution for the previous example:

- Finding the frequencies

Value	A	B	C	D	E	F
Frequency	5	25	7	15	4	12

- Building Huffman tree (see figure 3)

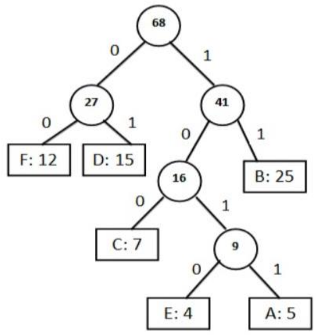

Fig 3: Huffman tree (another solution)

Table 2 shows the code for each value:

Table 2: values coding (example: another solution)

Character	frequency	Code	Compress size(bit)
A	5	1011	20
B	25	11	50
C	7	100	21
D	15	01	30
E	4	1010	16
F	12	00	24
Compressed file size(summation)			161
Compression ratio=68*8/161			3.3789

Here we increase CR to the value 3.3789.

3. LZW compression-decompression method

Lempel-Ziv-Welch method (LZW) of data compression-decompression is a lossless method of data compression-decompression [22, 23], the efficiency of this method depends on the frequency of groups or sets of values repeated in the input file (see figure 4), while Huffman method efficiency depends on the total repetition of each value in the input file.

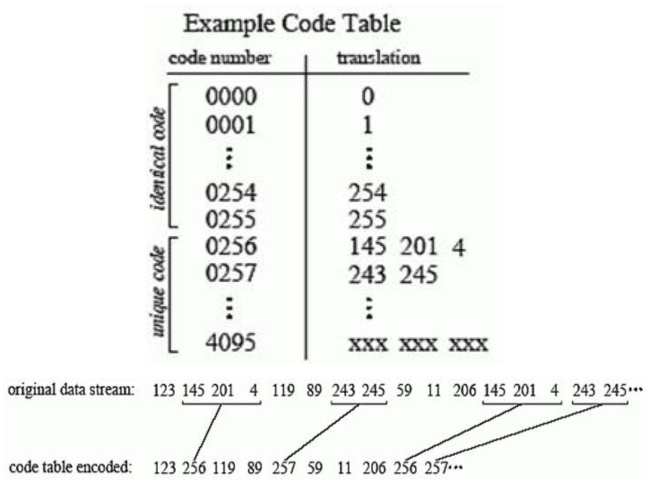

Fig 4: LZW compression

The process of LZW compression is described by the following algorithm:

```
Initialize table with single character strings
P = first input character
WHILE not end of input stream
     C = next input character
     IF P + C is in the string table
        P = P + C
     ELSE
        output the code for P
     add P + C to the string table
        P = C
      END WHILE
output code for P
```

While the process of decompression is described by the following algorithm:

```
read a character k;
  output k;
  w = k;
  while ( read a character k )
 /* k could be a character or a code. */
      {
      entry = dictionary entry for k;
      output entry;
      add w + entry[0] to dictionary;
      w = entry;
      }
```

The following example illustrates how LZW method works:

LZW compression

Input String: ^WED^WE^WEE^WEB^WET

w	k	Output	Index	Symbol
NIL	^			
^	W	^	256	^W
W	E	W	257	WE
E	D	E	258	ED
D	^	D	259	D^
^	W			
^W	E	256	260	^WE
E	^	E	261	E^
^	W			
^W	E			
^WE	E	260	262	^WEE
E	^			
E^	W	261	263	E^W
W	E			
WE	B	257	264	WEB
B	^	B	265	B^
^	W			
^W	E			
^WE	T	260	266	^WET
T	EOF	T		

LZW decompression

Input String (to decode): ^WED<256>E<260><261><257>B<260>T

w	k	Output	Index	Symbol
	^	^		
^	W	W	256	^W
W	E	E	257	WE
E	D	D	258	ED
D	<256>	^W	259	D^
^W	E	E	260	^WE
E	<260>	^WE	261	E^
^WE	<261>	E^	262	^WEE
E^	<257>	WE	263	E^W
WE	B	B	264	WEB
B	<260>	^WE	265	B^
^WE	T	T	266	^WET

The compression ratio here is calculated by dividing the input file size by compressed file size as illustrated below:

received	string table	decoding
a	–	a
b	table[256] = ab	b
c	table[257] = bc	c
256	table[258] = ca	ab
258	table[259] = abc	ca
257	table[260] = cab	bc
259	table[261] = bca	abc
262	table[262] = abca	abca
261	table[263] = abcab	bca
264	table[264] = bacb	bcab
260	table[265] = bcabc	cab
266	table[266] = cabc	cabc
263	table[267] = cabca	abcab
c	table[268] = abcabc	c

CR = (36*8)/(14*9) = 2.2857

4. Experimental analysis

The above explained methods of color image encryption-decryption were implemented using various images in sizes and types:

4.1 Implementing Huffman method

10 JPG color images were selected; table 3 shows the results of these implementations:

Table 3: Huffman JPG color image compression-decompression results

Image No	Dimension	Size (pixel)	Compressed file size(pixel)	CT	DT	CR
1	168x299x3	150696	140300	1.940000	15.075000	1.0741
2	168x299x3	150696	149370	1.930000	16.188000	1.0089
3	167x301x3	150801	138977	1.920000	14.779000	1.0851
4	183x275x3	150975	140007	1.965000	15.313000	1.0783
5	183x275x3	150975	144464	1.934000	15.608000	1.0451
6	168x301x3	151704	147864	1.985000	16.235000	1.0260
7	400x600x3	720000	668660	9.170000	73.436000	1.0768
8	422x750x3	949500	773310	11.757000	84.399000	1.2278
9	526x639x3	1008342	980874	12.764000	106.340000	1.0280
10	684x1024x3	2101248	1606280	26.338000	175.488000	1.3081

Another 10 PNG color images were selected; table 4 shows the results of these implementations:

Table 4: Huffman JPG color image compression-decompression results

Image No	Dimension	Size (pixel)	Compressed file size(pixel)	CT	DT	CR
1	187x269x1	50303	13809	0.684000	1.654000	3.6428
2	395x500x3	592500	94641	7.597000	10.704000	6.2605
3	423x476x3	604044	212533	7.788000	23.863000	2.8421
4	543x800x3	1303200	395437	16.764000	45.634000	3.2956
5	819x1024x3	2515968	1129203	31.870000	126.492000	2.2281
6	819x1024x3	2515968	1642867	32.134000	183.561000	1.5314
7	819x1024x3	2515968	1112694	31.746000	126.705000	2.2611
8	1835x1200x3	6606000	3603474	84.101000	404.324000	1.8332
9	1600x1600x3	7680000	2613496	97.712000	299.174000	2.9386
10	1854x1473x3	8192826	5703105	101.584000	614.526000	1.4366

Here we have to notice that if all the colors in the image are covered (colors are normalcy distributed), then the compression ratio will be decreased as shown in figures 5 and 6:

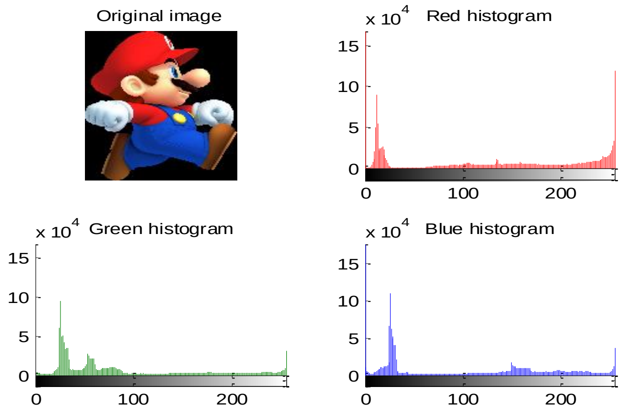

Fig 5: Colors are normally distributed CR=1.4366

Fig 6: Colors are not normally distributed CR=3.2956

4.2 Implementing LZW method

The same images were taken and compressed-decompressed using LZW method, tables 5 and 6show the experimental results:

Table 5: LZW JPG color image compression-decompression results

Image No	Dimension	Size (pixel)	Compressed file size(pixel)	CT	DT	CR
1	168x299x3	150696	125588	7.6390	5.4640	1.1999
2	168x299x3	150696	122472	8.1050	5.2330	1.2305
3	167x301x3	150801	106099	7.3530	4.6730	1.4213
4	183x275x3	150975	113884	7.8110	5.0450	1.3257
5	183x275x3	150975	116039	7.5800	4.9740	1.3011
6	168x301x3	151704	120413	8.0170	5.4250	1.2599
7	400x600x3	720000	523257	33.2890	22.7690	1.3760
8	422x750x3	949500	488543	38.0370	22.0950	1.9435
9	526x639x3	1008342	761316	45.9440	31.6930	1.3245
10	684x1024x3	2101248	1208907	86.7860	52.7720	1.7381

Table 6: LZW JPG color image compression-decompression results

Image No	Dimension	Size (pixel)	Compressed file size(pixel)	CT	DT	CR
1	187x269x1	50303	6554	1.8200	0.6870	7.6752
2	395x500x3	592500	31560	20.0790	4.0330	18.7738
3	423x476x3	604044	124838	20.5740	7.2180	4.8386
4	543x800x3	1303200	190780	41.0120	13.0120	6.8309
5	819x1024x3	2515968	690201	87.2560	35.5920	3.6453
6	819x1024x3	2515968	1247467	100.4410	56.7560	2.0169
7	819x1024x3	2515968	794914	89.9060	40.2990	3.1651
8	1835x1200x3	6606000	2116507	233.0590	108.1170	3.1212
9	1600x1600x3	7680000	1394693	258.7330	93.1600	5.5066
10	1854x1473x3	8192826	4158041	334.2900	196.0240	1.9704

4.3 Results comparisons

The obtained experimental results were used to calculate the compression throughput of each method (number of pixels compressed per second and it is equal the image size in pixels divided by the compression time in seconds), table 7 shows the throughputs for both methods:

Table 7: Compression throughput calculation

Image size	Huffman method		LZW method		Huffman enhancement
	CT	Throughput (pixel per second)	CT	Throughput (pixel per second)	
150696(jpg)	1.94	77678	7.639	19727	3.9376
150696	1.93	78081	8.105	18593	4.1995
150801	1.92	78542	7.353	20509	3.8296
150975	1.965	76832	7.811	19329	3.975
150975	1.934	78064	7.58	19918	3.9193
151704	1.985	76425	8.017	18923	4.0387
720000	9.17	78517	33.289	21629	3.6302
949500	11.757	80760	38.037	24963	3.2352
1008342	12.764	78999	45.944	21947	3.5995
2101248	26.338	79780	86.786	24212	3.2951
50303(png)	0.684	73542	1.82	27639	2.6608
592500	7.597	77991	20.079	29508	2.643
604044	7.788	77561	20.574	29360	2.6417
1303200	16.764	77738	41.012	31776	2.4464
2515968	31.87	78945	87.256	28834	2.7379
2515968	32.134	78296	100.441	25049	3.1257
2515968	31.746	79253	89.906	27984	2.8321
6606000	84.101	78548	233.059	28345	2.7711
7680000	97.712	78598	258.733	29683	2.6479
8192826	101.584	80651	334.29	24508	3.2908

Here we can see that Huffman method has a better throughput and the enhancement factor is always greater than 1.

Also we calculated the decompression throughput as shown in table 8:

Table 8: Decompression throughput calculation

Compressed Image size (Huffman)	Compressed Image size (LZW)	Huffman method		LZW method		LZW enhancement
		DT	Throughput (pixel per second)	DT	Throughput (pixel per second)	
140300(jpg)	125588	15.075	9306.8	5.464	22985	2.4697
149370	122472	16.188	9227.2	5.233	23404	2.5364
138977	106099	14.779	9403.7	4.673	22705	2.4145
140007	113884	15.313	9143	5.045	22574	2.469
144464	116039	15.608	9255.8	4.974	23329	2.5205
147864	120413	16.235	9107.7	5.425	22196	2.4371
668660	523257	73.436	9105.3	22.769	22981	2.5239
773310	488543	84.399	9162.5	22.095	22111	2.4132
980874	761316	106.34	9223.9	31.693	24022	2.6043
1606280	1208907	175.488	9153.2	52.772	22908	2.5027
13809(png)	6554	1.654	8348.9	0.687	9540	1.1427
94641	31560	10.704	8841.6	4.033	7825	0.885
212533	124838	23.863	8906.4	7.218	17295	1.9419
395437	190780	45.634	8665.4	13.012	14662	1.692
1129203	690201	126.492	8927.1	35.592	19392	2.1723
1642867	1247467	183.561	8950	56.756	21979	2.4558
1112694	794914	126.705	8781.8	40.299	19725	2.2461
3603474	2116507	404.324	8912.3	108.117	19576	2.1965
2613496	1394693	299.174	8735.7	93.16	14971	1.7138
5703105	4158041	614.526	9280.5	196.024	21212	2.2857

Here we can see that the decompression throughput of LZW method is better, that is because the size of LZW compressed file is smaller.

For both types of images LZW method has a better compression ratio as shown in table 9:

Table 9: Compression ratio comparisons

Image size	LZW CR	Huffman CR	LZW enhancement (increasing)
150696(jpg)	1.1999	1.0741	1.1171
150696	1.2305	1.0089	1.2196
150801	1.4213	1.0851	1.3098
150975	1.3257	1.0783	1.2294
150975	1.3011	1.0451	1.245
151704	1.2599	1.026	1.228
720000	1.376	1.0768	1.2779
949500	1.9435	1.2278	1.5829
1008342	1.3245	1.028	1.2884
2101248	1.7381	1.3081	1.3287
50303(png)	7.6752	3.6428	2.107
592500	18.7738	6.2605	2.9988
604044	4.8386	2.8421	1.7025
1303200	6.8309	3.2956	2.0727
2515968	3.6453	2.2281	1.6361
2515968	2.0169	1.5314	1.317
2515968	3.1651	2.2611	1.3998
6606000	3.1212	1.8332	1.7026
7680000	5.5066	2.9386	1.8739
8192826	1.9704	1.4366	1.3716

From the obtained results we can raise the following facts:
Both methods can be used to compress-decompress color images, and CR always greater than 1.
It is better to used LZW method because it always has a better CR.
Use Huffman method to compress a color image, when the colors are not normally distributed.
LZW method is more efficient when it deals with PNG imaged.
CR ratio of Huffman method increases when dealing with PNG image, but it always smaller than LZW CR.

5. Conclusion

Different images in sizes and types were compressed-decompressed using both Huffman and LZW methods, the experimental results showed that LZW method is more efficient because it always has a better compression ratio.
Huffman method efficiency will increases when it deals with images with unmoral distributed colors and with PNG images.
LZW method efficiency will increases when it deals PNG images.

6. References

1. Ziad A, Alqadi, Majed O, Al-Dwairi, Amjad A, Abu Jazar, Rushdi Abu Zneit. Optimized True-RGB color Image Processing. World Applied Sciences Journal. 2010; 8(10):1175-1182, ISSN 1818-4952, 2010.
2. Moustafa Z, Alqadi A. Color Image Reconstruction Using A New R'G'I Model. Journal of Computer Science. 2009; 5(4):250-254.
3. Matrouk K, Al-Hasanat A, Alasha'ary H, Al-Qadi, Al-Shalabi H. “Speech fingerprint to identify isolated word person”, World Applied Sciences Journal. 2014; 31(10):1767-1771.
4. Al-Azzeh J, Zahran B, Alqadi Z, Ayyoub B, Abu-Zaher M. “A Novel zero-error method to create a secret tag for an image”, Journal of Theoretical and Applied Information Technology. 2018; 96(13):4081-4091.
5. Prof. Ziad AA Alqadi, Prof Mohammed K, Abu Zalata, Ghazi M. Qaryouti, Comparative Analysis of Color Image Steganography, JCSMC. 2016; 5(11):37–43.
6. Jose M. “Hiding Image in Image Using LSB Insertion Method with Improved Security and Quality”, International Journal of Science and Research. 2014; 3(9):2281-2284.
7. Patel RM, Shah DJ. “Conceal gram: Digital image in image using LSB insertion method”, International Journal of Electronics and Communication Engineering & Technology. 2013; 4(1):230-2035.
8. Akhtar N, Johri P, Khan S. “Enhancing the security and quality of LSB based image steganography”, 5th International Conference on Computational Intelligence and Communication Networks, Mathura, India. 2013, 27-29.
9. Juneja M, Sandhu PS. “An improved LSB based Steganography with enhanced Security and Embedding/Extraction”, 3rd International Conference on Intelligent Computational Systems, Hong Kong China, January 26-27, 2013.
10. Alasha'ary K, Matrouk A, Al-Hasanat Z, lqadi A, Al-Shalabi H. Improving Matrix Multiplication Using Parallel Computing, International Journal on Information Technology (I.RE.I.T.) 1(6) ISSN 2281-2911, 2013.
11. Bilal Zahran, Ziad Alqadi, Jihad Nader, Ashraf Abu Ein. A COMPARISON BETWEEN PARALLEL AND SEGMENTATION METHODS USED FOR IMAGE ENCRYPTION-DECRYPTION, International Journal of Computer Science & Information Technology (IJCSIT), 8(5):201.
12. Alqadi ZA, Abu-Jazzar A. Analysis Of Program Methods Used For Optimizing Matrix Multiplication, Journal of Engineering. 2005; 15(1):73-78.
13. Alqadi ZA, Aqel M, El Emary IM. "Performance Analysis and Evaluation of Parallel Matrix Multiplication Algorithms", World Applied Sciences Journal, vol. 5(2), ISSN 1818-4952, 2008.
14. Alasha'ary H, Matrouk K, Al-Hasanat A, Alqadi Z, Al-Shalabi H. Improving Matrix Multiplication Using Parallel Computing, International Journal on Information Technology(I.RE. I.T.) Vol. 1, N. 6 ISSN 2281-2911, 2013.
15. Ziad Al-Qadi, Musbah Aqel. Performance analysis of parallel matrix multiplication algorithms used in image processing, World Applied Sciences Journal. 2009; 6(1):45-52.
16. Jamil Al-Azzeh, Bilal Zahran, Ziad Alqad. Salt, Pepper Noise. Effects and Removal International Journal on Electrical Engineering and Informatics 2(4),
17. Vimala T. “Salt and Pepper Noise Reduction Using Mdbutm Filter With Fuzzy Based Refinement. 2012; 2:5.
18. Jassim FA. Image Denoising Using Interquartile Range Filter with Local Averaging, International Journal of Soft Computing and Engineering (IJSCE). 2013; 2(6):424-428.
19. Jamil S. AL-Azzeh. Distributed Mutual Inter-Unit Test Method For D-Dimensional Mesh-Connected Multiprocessors With Round-Robin Collision Resolution: Jordanian Journal of Computers and Information Technology, 2019.
20. Jamil AL-Azzeh, Bilal Zahran, Ziad Alqadi, Belal Ayyoub, Muhammed Mesleh. A Novel Based On Image Blocking Method To Encrypt-Decrypt Color

JOIV: International Journal on Informatics Visualization, 2019.
21. Jamil AL-Azzeh. Evaluation Method for SDN Network Effectiveness in Next Generation Cellular Networks: International Journal of Communication Networks and Information Security, 2018.
22. Jamil AL-Azzeh. Improved testability method for mesh-connected VLSI multiprocessors: Jordanian Journal of Computers and Information Technology, 2018.
23. Jamil AL-Azzeh. A Novel Zero-Error Method to Create a Secret Tag for an Image; Journal of Theoretical and Applied Information Technology, 2018.
24. Jamil AL-Azzeh. Qualitative risk analysis of software development; Asian Journal of Information Technology, 2018.
25. Bilal Zahran, Jamil AL-Azzeh. A Modified LBP Method To Extract Features From Color Images: Journal of Theoretical and Applied Information Technology, 2018.
26. Jamil AL-Azzeh. Information Technologies for Supporting Administrative Activities of Large Organizations; DESIDOC Journal of Library & Information Technology. 2018; 38:3.
27. Jamil AL-Azzeh. A Distributed Multiplexed Mutual Inter-Unit in-Operation Test Method for Mesh-Connected VLSI Multiprocessors; Jordan Journal of Electrical Engineering. 2017; 10:5.
28. Jamil AL-Azzeh. Fault-Tolerant Routing in Mesh-Connected Multicomputer based on Majority-Operator-Produced Transfer Direction Identifiers; Jordan Journal of Electrical Engineering. 2017; 3:2.
29. Jamil AL-Azzeh. Analysis of Self-Similar Traffic Models in Computer Networks; International Review on Modelling and Simulations. 2017; 10:5.
30. Jamil AL-Azzeh. Statistical Analysis of Methods Used to Enhanced Color Image Histogram; XX International Scientific and Technical Conference; Russia, 2017, 24-26.
31. Mazen Abuzaher, Jamil AL-Azzeh. JPEG Based Compression Algorithm; International Journal of Engineering and Applied Sciences. 2017; 4:4.
32. Jamil AL-Azzeh. ZigBee, Bluetooth and Wi-Fi Complex Wireless Networks Performance Increasing; International Journal On Communications Antenna and Propagation. 2017; 7:1.
33. Jamil AL-Azzeh. Implementing Built-In Test in Analog and Mixed-Signal Embedded-Core-Based System-On-Chips; Asian Journal of Information Technology, Medwell Journals, 2016. (SJR indicator = 0.11).
34. Jamil AL-Azzeh. Creating a Color Map to be used to Convert a Gray Image to Color Image; International Journal of Computer Applications. 2016; 153(2):0975-8887.
35. Jamil AL-Azzeh. Analysis of Second Order Differential Equation Coefficients Effects on PID Parameters International Journal on Numerical and Analytical Methods in Engineering (IRENA). 2016; 4:2.
36. Jamil AL-Azzeh. Automated Demodulation of Amplitude Modulated Multichannel Signals with Unknown Parameters Using 3D Spectrum Representation Research Journal of Applied Sciences, Engineering and Technology, Maxwell Scientific Publication, 2016.
37. Jamil Al-Azzeh S. Adaptive Regulation of Radiated Power Radio Transmitting Devices in Modern Cellular Network Depending on Climatic Conditions. Contemporary Engineering Sciences, 2016, 9.
38. Mazin Al Hadidi, Jamil AL-Azzeh. Methods of Risk Assessment for Information Security Management, International Review on Computers and Software (I.RE. CO.S.), Vol. 11, N. 2 ISSN 1828-6003 (impact factor = 6.14)**,** 2016.
39. Jamil AL-Azzeh. Bidirectional Virtual Bit-slice Synchronizer: A Scalable Solution for Hardware-level Barrier Synchronization. Research Journal of Applied Sciences, Engineering and Technology**.** Maxwell Scientific Publication Corp. 2015; 11(8):902-909.
40. Jamil AL-Azzeh. The Organization of Built-in Hardware-Level Mutual Self-Test in Mesh-Connected VLSI Multiprocessors; International Journal on Information Technology (I.RE.I.T.) 3, Praise worthy Prize, 2015.
41. Jamil AL-Azzeh. an approach to achieving increased fault-tolerance and availability of multiprocessor-based computer systems; Australian Journal of Basic and Applied Sciences, 2014.
42. Jamil AL-Azzeh. Computer simulation of vibration robot created for the wall movement; Research Journal of Applied Sciences. 2014; 9:597-602,
43. Jamil AL-Azzeh. Review of Methods of Distributed Barrier Synchronization of Parallel Processes in Matrix VLSI Systems, International Review on Computers and Software (IRECOS), Praise worthy Prize. 2013; 8A(4):42- 46,
44. Jamil AL-Azzeh. Fastest Color Model for Image Processing Using Embedded Systems Australian Journal of Basic and Applied Sciences. 2013; 7(14):83-89.
45. Jamil AL-Azzeh. Using Virtual Network to Solve Freight Company Problems; World Applied Sciences Journal. 2013; 27(6):754-758 (SJR indicator = 0.17).
46. Jamil AL-Azzeh. Detection of eyes using FCM, International Review on Computers and Software (IRECOS), Praise Worthy Prize, 2012; 7A(4):1428-1434 (impact factor = 6.14).
47. Jamil AL-Azzeh. An optical character recognition, Contemporary Engineering Sciences. 2012; 5(11):521-529.
48. Jamil AL-Azzeh. Analysis of Matrix Multiplication Computational Methods European Journal of Scientific Research. 2014; 121(3):258-266.
49. Ziad A. Al-Qadi, Musbah J. Aqel Performance analysis of parallel matrix multiplication algorithms used in image processing: World Applied Sciences Journal. 2009; 6(1):45-52.
50. Mohammed Abuzalata, Ziad Alqadi, Jamil Al-Azzeh. Qazem Jaber Modified Inverse LSB Method for Highly Secure Message Hiding; International Journal of Computer Science and Mobile Computing. 2019; 8(2):93-103.
51. Qazem Jaber, Rashad J, Rasras Mohammed Abuzalata, Ziad Alqadi, Jamil Al-Azzeh. Comparative Analysis of Color Image Encryption-Decryption Methods Based on Matrix Manipulation: International Journal of Computer Science and Mobile Computing. 2019; 8:2.
52. Jamil Al-Azzeh, Ziad Alqadi, Mohammed Abuzalata. Performance Analysis of Artificial Neural Networks

used for Color Image Recognition and Retrieving: International Journal of Computer Science and Mobile Computing. 2019; 8:2.

53. Rashad J. Rasras, Mohammed Abuzalata, Ziad Alqadi, Jamil Al-Azzeh Qazem Jaber. Comparative Analysis of Color Image Encryption-Decryption Methods Based on Matrix Manipulation International Journal of Computer Science and Mobile Computing. 2019; 8(3):14-26.
54. Jamil Al-Azzeh, Bilal Zahran, Ziad Alqadi, Belal Ayyoub, Muhammed Mesleh. A Novel Based on Image Blocking Method to Encrypt-Decrypt Color; International Journal on informatics visualization. 2019; 3:1.
55. Jamil Al-Azzeh, Zahran B, Alqadi Z, Ayyoub B, Mesleh M. A Novel Based on Image Blocking Method To Encrypt-Decrypt Color, International Journal on Informatics Visualization. 2019; 3:1.
56. Zahran B, AL-Azzeh J, Al Qadi Z, Al Zoghoul M, Khawatreh S. A Modified LBP Method to Extract Features from Color Images, Journal of Theoretical and Applied Information Technology (JATIT). 2018; 96:10.
57. AL-Azzeh J, Zahran B, Alqadi Z, Ayyoub B, Abu-Zaher M. A novel Zero-error Method to Create a Secret Tag for an Image, Journal of Theoretical and Applied Information Technology (JATIT). 2018; 96(13):4081-4091.
58. Jamil AL-Azzeh, Zahran B, Alqadi Z. Salt and Pepper Noise: Effects and Removal, International Journal on Informatics Visualization. 2018; 2(4):252-256.
59. Jihad Nader, Ziad Alqadi, Bilal Zahran. Analysis of Color Image Filtering Methods, International Journal of Computer Applications (IJCA). 2017; 174(8):12-17.
60. Ziad Alqadi, Bilal Zahran, Jihad Nader, Estimation and Tuning of FIR Lowpass Digital Filter Parameters, International Journal of Advanced Research in Computer Science and Software Engineering. 2017; 7(2):18-23.