PORT SAID ENGINEERING RESEARCH JOURNAL
Faculty of Engineering - Port Said University
Volume 21 No. 1 March 2017 pp. 95:108

Image Compression Using Advanced Optimization Algorithms

Mohamed E. Emara, Rehab F. Abdel-Kader, and Mohamed S. Yasein

ABSTRACT

In this paper a new image compression schema that uses three-dimensional discrete cosine transform and relies on two-dimensional discrete wavelet transform, for image classification, is proposed. The proposed technique utilizes a modified quantization table and a method for converting a three-dimensional image cube into a one-dimensional array, which provides better coding efficiency in the run length coding step. To ensure faster performance, this technique provides proposed parallel computation system. Several images have been used to test the proposed algorithm. Experimental results demonstrate that the proposed algorithm outperforms previous compression methods in terms of peak-signal-to-noise ratio with a lower compression bit rate.

1. INTRODUCTION

Image coding and compression techniques have become a highly active research area over the past decades [1]. The main purpose of image compression is to reduce the redundancy [2] of the image in order to store or transmit data in an efficient form. One common characteristic in most images is that neighboring pixels are highly correlated and consequently contain a lot of redundant information. To find an image representation in which pixels are less-correlated, various image compression schemes that utilize image transformations can be used. There are two compression techniques that used for image compression (Lossless and Lossy compression [3]). When the reconstructed image after compression become numerically identical to the original image, it called lossless image compression. Because it depends on achieving a simple ratio of compression. And when the reconstructed image after compression become more deterioration to the original image, it called lossy image compression. Because it depends on achieving a much higher ratio of compression.

The common model of lossy image compression system consists of three phases:

1- Image Encoder.
2- Image Quantization.
3- Image Decoder.

To process the compression, it applying a linear transform to make the pixels of image less-correlated. Applying the quantization process to the coefficients that resulting from the transform. And then comes the role of the decoding Process that used for find the accurate probabilities for each quantized value and make the convenient code by these probabilities.

Electrical Engineering Department, Faculty of Engineering, 42523, Port-Said, Egypt
mohamad_emara@himc.psu.edu.eg,
rehabfarouk@eng.psu.edu.eg,
myasein@eng.psu.edu.eg

This process to compare the output code stream with the input code stream, and the ratio of compression depends on this process result. Discrete Cosine Transformation (DCT) [4] is the most easily and widely used transform in image compression schemes. It has excellent compaction for highly correlated data. It provides a good tradeoff between information-packing ability and computational complexity. There are a lot of fields that used the DCT. In image compression, JPEG [5] compression is the well-known algorithm that used the DCT and achieved high compression with less data loss. And also in the video compression, MPEG [6] compression is one of the best algorithm to compress videos in low storage with high quality. In watermarking field [7], there are a lot of researches that used the DCT to achieve better results. The image processing with parallel computation [8] is proposed in many techniques [9], depending on the multicore systems instead of single processor system to overcome the consumption time problem. There are a lot of fields that used the parallel computation. In [10], it discussed the segmentation technique using a CPU and GPU proposed parallel technique. In [11-12], the parallel processing is used with video compression field. In [13-15], the image compression field used the CPU and GPU parallel computing in several ways to show how it is effective in image.

In this paper, a new image compression technique using 3D-DCT is proposed. The proposed technique utilizes a modified quantization table and a method for converting a 3D image cube into an 1D array, which provides better coding efficiency in the run length coding step. In order to improve the performance of the image compression algorithm, a Two-Dimensional Discrete Wavelet Transform (2D-DWT) based classification is used. Initially, the image is analyzed by 2D-DWT [16] to determine the type of the image. The given 2D image is converted into 3D cubes of sizes that are determined based on the image type (low or high detail). 3D-DCT is then applied on the formatted cubes. Several images have been used to test the proposed algorithm. A proposed parallel computation is used to improve the time consumption of the proposed image compression in two ways (CPU parallel computation, GPU parallel computation). Test results demonstrate that the proposed algorithm outperforms previous compression methods in terms of Peak-Signal-to-Noise Ratio (PSNR) with a lower compression bit rate. The paper is organized as follows. In section 2, the 2D-DCT and 3D-DCT compression techniques are previewed. In section 3, the proposed image compression algorithm is described in detail. In section 4, the proposed parallel computation technique is described in detail. In section 5, the experimental results of testing the proposed method is presented and discussed. Finally, conclusions are drawn in Section 5.

2. 2D-DCT AND 3D-DCT BASED TECHNIQUES

In this section, the 2D-DCT and the 3D-DCT compression techniques are discussed.

2.1. 2D-DCT Based Techniques

JPEG is designed to support a wide variety of applications for continuous-tone images. JPEG compression can be characterized as a lossy DCT-based coding technique [17]. The DCT transform is used to concentrate the total energy, contained initially in the original signal, in a small number of the transformed coefficients. 2D-DCT is typically used in image compression because image data are naturally two dimensional. In general, increasing the dimension of any signal makes it better to get smaller distances in the new multi-dimensional constellation and it from the concept of Trellis Coding [18] That prove that increasing the constellation size reduces Euclidean distances between the constellation points but sequence coding offers a coding gain that overcomes the power disadvantage of going to the higher constellation. When the distances get closer, DCT coefficients get smaller. In Image compression, the 2D-DCT is applied on 2D images and DCT coefficients can be determined by (1).

$$F(u,v) = C(u)C(v)K\sum_{x=0}^{N-1}\sum_{y=0}^{M-1} f(x,y)\, A_{ux}\, A_{vy} \qquad (1)$$

, where A_{ij} and K determined by (2), (3)

$$A_{ij} = \cos\left[\frac{(2j+1)i\,\pi}{2N}\right] \qquad (2)$$

$$K = \sqrt{\frac{8}{MxNxL}} \qquad (3)$$

and x = 0,1,...,N−1; y = 0,1,...,M−1 ; C(u) and C(v) are determined by (4).

$$C(\varepsilon) = \begin{cases} \frac{\sqrt{2}}{2} & if \quad \varepsilon = 0 \\ 1 & otherwise \end{cases} \qquad (4)$$

2.2. 3D-DCT Based Techniques

The 3D-DCT has been used in many techniques. Some of these techniques focus on integral images [19-20], where images elements are placed along the third dimension before applying 3D-DCT. This is commonly used in video compression [21-22], in which the temporal domain is used as the third dimension. In visual tracking [23], 3D-DCT is frequently used to represent the object appearance model that is robust to variations in illumination, pose, etc. Furthermore, 3D-DCT is used in many image processing applications, such as video watermarking, denoising, fingerprinting, and [24- 26]. 3D-DCT sequential coding is used for specific classes of images like medical images [27]. In [28], 3D spiral JPEG is used, where it depends on spiral scanning to format the multi-dimensional constellation to get a more effective compression scheme [29]. For hyperspectral space images, different techniques are proposed, as in [30-31]. In video compression, 3D-DCT is used by taking the basis as the two-dimensional frames and the temporal dimension (the sequence of frames). DCT coefficients can be determined by (5).

$$F(u,v) = C(u)C(v)C(p)K\sum_{x=0}^{N-1}\sum_{y=0}^{M-1}\sum_{z=0}^{L-1} f(x,y,z)\, A_{ux}\, A_{vy} A_{pz} \qquad (5)$$

, where A_{ij} and K determined by (6), (7)

$$A_{ij} = \cos\left[\frac{(2j+1)i\,\pi}{2N}\right] \qquad (6)$$

$$K = \sqrt{\frac{8}{MxNxL}} \qquad (7)$$

and x = 0,1,...,N−1; y = 0,1,...,M−1; z = 0,1,...,L−1 ; *C(u)* and *C(v)* are determined by (8).

$$C(\varepsilon) = \begin{cases} \frac{\sqrt{2}}{2} & if \quad \varepsilon = 0 \\ 1 & otherwise \end{cases} \qquad (8)$$

2.3. JPEG Compression Architecture

The input image is first divided into 8×8 blocks; then 2D DCT is applied to each block. DCT coefficients are then quantized using an 8×8 quantization table, as described in the JPEG standard [32]. The quantization is performed by dividing each element of the transformed original data matrix by the corresponding element in the quantization matrix Q and rounding to the nearest integer value as defined by (9).

$$D_{quant}(i,j) = round\left(\frac{D_{DCT}(i,j)}{Q(i,j)}\right) \qquad (9)$$

, where Dquant is the quantized coefficient, DDCT is the DCT coefficient, and Q is the quantization matrix. Further compression is achieved by applying an appropriate scaling factor. In order to reconstruct the data, the rescaling and the de-quantization is performed. The de-quantized matrix is then transformed back using the inverse-DCT. The entire procedure is shown in Figure 1.

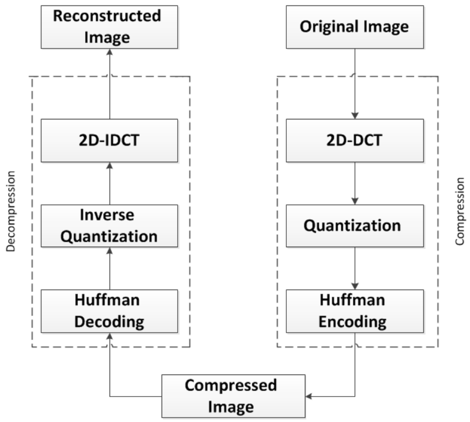


Figure 1. JPEG Compression Block Diagram

3. THE PROPOSED IMAGE COMPRESSION ALGORITHM

The main goal of the proposed compression algorithm is to achieve a high compression ratio with minimum information loss. The block diagram for the full proposed System is shown in Figure 7. In the following subsections, the main steps of the proposed 3D-DCT based compression technique are described.

3.1. 3D Cube Formation

A 2D input image is first mapped into a set of 3D data cubes. This is done by grouping NxN blocks. These blocks are processed from left-to-right and top-to-bottom. Eight NxN blocks are used to construct a 3D data cube as shown in Figure 2.

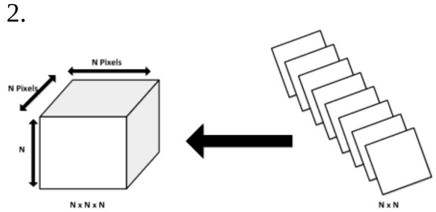
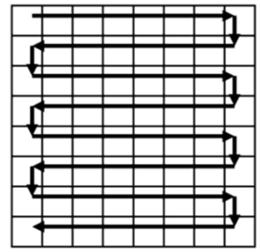
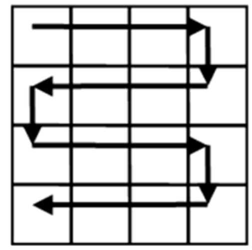


Figure 2. 3D Cube Formation Process

In our technique, we use two cube-dimensions: 8x8x8 cube and 4x4x4 cube, depending of the type of the image, as discussed in the following sections. The image scanning process of images to construct 3D cubes is illustrated in Figure 3.



(a)



(b)

Figure 3. (a) 8x8 Scanning for 8x8x8 first item of cube. (b) 4x4 Scanning for 4x4x4 first item of cube.

3.2. Image classification

When correlation between pixels (The similarity between blocks [33]) is high, DCT coefficients get smaller and consequently, it yields a better compression. Images can be classified into two main types: high-detail and low-detail images. It is clear that the similarity ratio in low-detail images is more than high–detail images. Based on that, we suggest that the 8x8x8 cube size would be more suitable for the low detail images because it has high similarity ratio, while the 4x4x4 cube size would be more suitable for the high-details images because of its low similarity ratio. In the proposed algorithm, an image classifier on decreasing the dimension of the image by using DWT and using the details images to determine its type and then decide the suitable cube size. The result of 2D-DWT is a decomposed image into four quadrants, as shown in Figure 4.

LL: The upper left quadrant and it is denoted by LL and represents the approximated version of the original at half the resolution.

HL/LH: The lower left and the upper right blocks. The LH block contains vertical edges. In contrast, the HL blocks shows horizontal edges.

HH: The lower right quadrant. We can interpret this block as the area, where we find edges of the original image in diagonal.

Figure 4. 2D-DWT

The GxG image (512 x512 or any other size) is decomposed by 1-level 2D-DWT, then take the inverse DWT with only the LH sub band (it is chosen experimentally based on the vertical edges of image) and the other sub bands set to zero. We take the 2D-DFT (Discrete Fourier Transform) [34] to compute the mean of the inversed image. The computed mean is compared to a certain threshold T (decided experimentally). If it is lower than T, then it is a low details image. Otherwise it is a high details image. The block diagram for this step is shown in Figure 5.

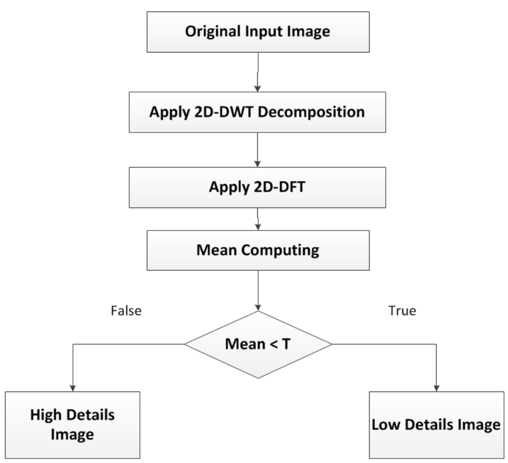


Figure 5.2D-DWT Control System Block Diagram

3.3. 3D-DCT Transformation

The 3D-DCT, as in in eq.3, is performed on each cube.

3.4. 3D Quantization

The quantization matrix that is used is MxMxM matrix and can be determined by (10).

$$Q(i,j,k) = 3x + 6y + 1z \tag{10}$$

Although it is not optimized, based on our experiments this formula has provided the best results. The proposed quantization table can be multiplied by a scalar to get varying levels of compression rates and picture quality.

3.5. 3D Zig-Zag Scanning

It is necessary to convert a 3D cube into an 1D vector, before the variable length Huffman coding is applied. The 3D scan order for 8x8x8 cube can be done with four techniques: (3D-Zig-Zag,3D-Horizontal,3D-Vertical and 3D-Hilbert Scanning) as illustrated in Figure 6.

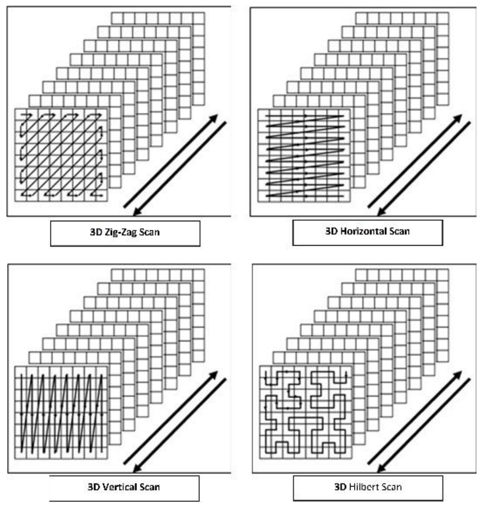


Figure 6.3D Zig-Zag Scanning

3.6. Huffman Encoding

Huffman coding is combined with reducing the image redundancies using DCT to help in compressing the image data to a better level.

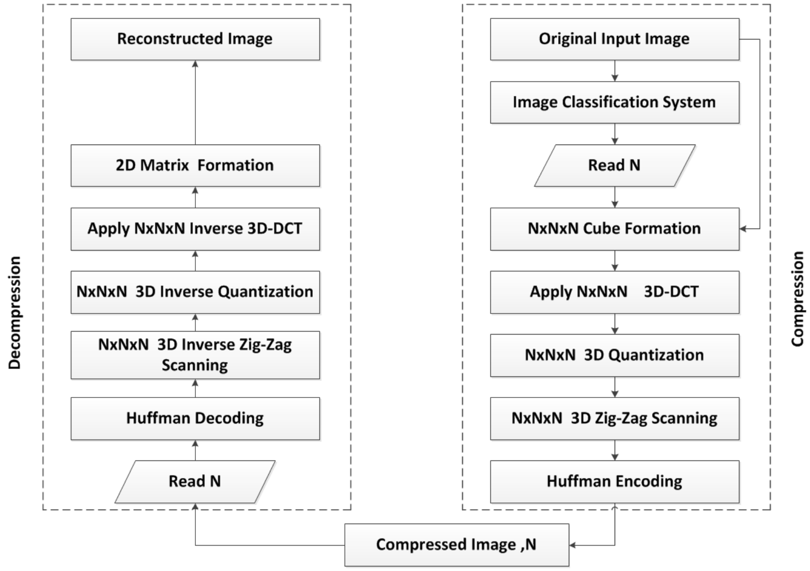


Figure 7. Encoder and Decoder Block Diagram

4. THE PROPOSED PARALLEL COMPUTATION TECHNIQUES

We used two techniques of the parallel computation (SPMD, CUDA techniques).

4.1. SPMD (Single Program Multiple Data):

The main idea of SPMD technique depend on that you have one block code and have different data are used in this code. In SPMD technique, there are number of workers that would be defined. So, by running identical code on all workers, each worker can have different, unique data for that code.

This proposed image compression using SPMD executed in the following steps:

1- Divide the original image blocks into four sub-images.
2- The proposed image compression algorithm (Single Code) runs on the four sub-images simultaneously in parallel.
3- The four decompressed sub-images would be reassembled in the final decompressed image.

The block diagram for the full proposed System with SPMD is shown in Figure.8.

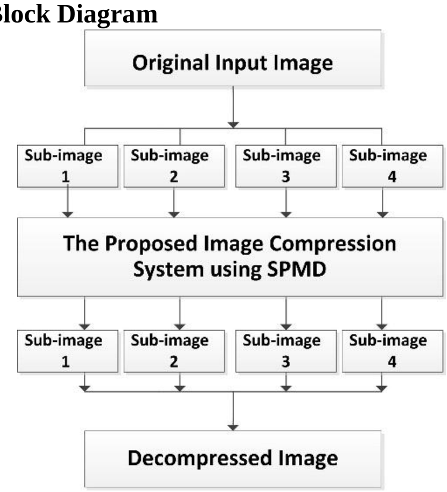


Figure 8. The Full Proposed Image Compression by SPMD Block Diagram

4.2. CUDA Parallel Processing:

This part will display the details of the acceleration using CUDA with MATLAB code. Using NVIDIA graphics processor unit (GPU), the CUDA programming re-define some phases of MATLAB code into C-like language to re-evaluate it into parallelization process. The improvement of the proposed image compression algorithm consumption time is evaluated in the experimental result.

4.2.1. 3D Cube Formation Parallelization

In the 3D Cube formation code, there are a high number of iteration (for loops). Every iteration creates one cube of NxNxN blocks, so we will parallelize these iterations in threads as shown in Figure.9.

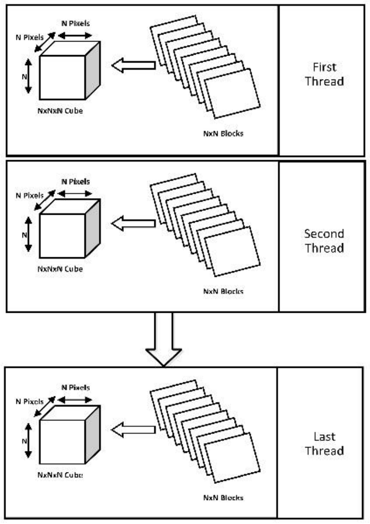


Figure 9. 3D Cube Formation Parallelization Block Diagram

4.2.2. 3D-DCT and 3D-Quantization Parallelization

In this part, every iteration has NxNxN cube and perform the main process of the proposed image compression when we apply the 3D-DCT and 3D-Quantization. So, we will parallelize these iterations in threads as shown in Figure 10.

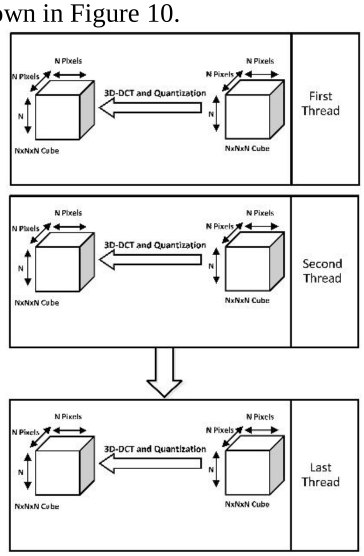


Figure 10. 3D-DCT-Quantization parallelization block diagram

4.2.3. Proposed Scanning Parallelization

The proposed scanning depends on performing four different scanning (Zigzag, Horizontal, Vertical and Hilbert Scanning) to get the better result. So, we will parallelize these four methods in threads as shown in Figure 11.

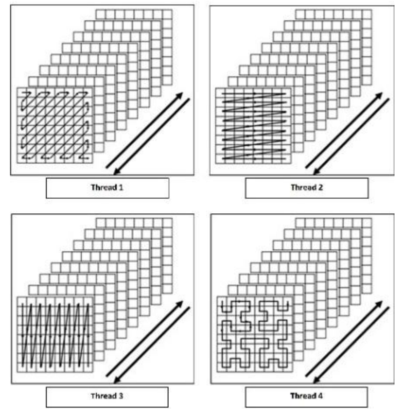


Figure 11. Proposed Scanning Parallelization Block Diagram

5. PERFORMANCE EVALUATION AND EXPERIMENTAL RESULTS

5.1. The proposed Image compression algorithm

The performance of the algorithm is evaluated in terms of the picture quality and compression ratio. To evaluate the picture quality in image compression systems, reliable quality measures should be used. A set of objective picture quality measures for image compression systems were investigated in [35] and emphasized the correlation between these measures and the subjective picture quality measures. The most commonly used measure is Peak Signal to Noise Ratio (PSNR) and the Compression Ratio (CR) can be determined by (11), (13).

$$PSNR = 10x\ \log_{10} \frac{255^2}{MSE} \tag{11}$$

, where the Mean Square Error MSE determined by (12).

$$MSE = \frac{\sum_{M,N} [I_1(m,n) - I_2(m,n)]^2}{MxN} \tag{12}$$

, where I1 is the original image, I2 is the compressed image and MxN is the total number of pixels in the original image.

$$CR = \frac{B_0}{B_1} \tag{13}$$

, where, B0 is the number of bits before compression, and B1 is the number of bits after compression. Alternatively, the bit rate, which measures the number of bits per pixel (BPP) can be determined by (14).

$$BPP = \frac{B_1}{MxN} \tag{14}$$

, where B1 is number of bits after compression and MxN is the total number of pixels in an image.

5.1.1. Bit rate

The proposed algorithm was tested using several test images of dimension 512 x512. The results are displayed in Tables 1 and 2 in terms of PSNR and BPP. Results obtained using the proposed 3D-DCT algorithm is compared with a standard JPEG compression system as shown in Table 1 Low-detail images such as (Lena, Cameraman, Pepper, Elaine and Barbara) and high-detail images such as (Baboon, Bridge, Hildebrandt, City, Cartoon) were compressed using 4x4x4 and 8x8x8 cubes. It can be seen in Table.1 that results of using 8x8x8 Cubes and 4x4x4 Cubes show that the 8x8x8 cube formation gives better bit rate than the JPEG algorithm in the case of low-detail images (have a low bit rate and a high PSNR). Also, it can be seen that the 4x4x4 cube formation gives better bit rate than the JPEG algorithm in the case of high-detail images (have a high bit rate and a high PSNR). Figure 15 shows some images that were used in the experimental testing for different PSNR. Lena and Baboon images were selected as examples for low and high-detail images and tested with different PSNR values and the results are shown in using Figure 12 and Figure 13.

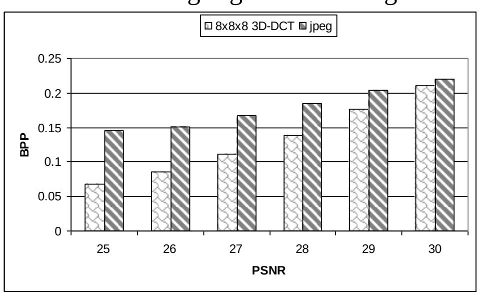


Figure 12. PSNR vs BPP in Lena image

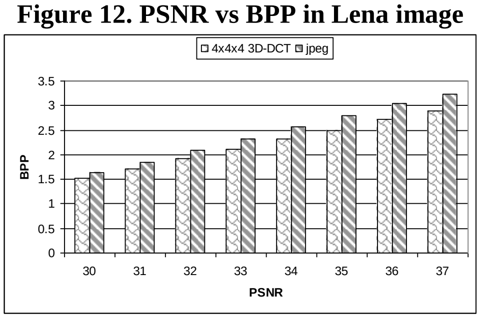


Figure 13. PSNR vs BPP in Baboon image

In Figure.12, it can be noticed that the bit rate improves with high PSNR in case of 8x8x8 cube. While, in Figure.13, it can be shown that the bit rate improves for high PSNR in case of the 4x4x4 cube formation. Table 2 shows a comparison of the results obtained using the proposed 3D-DCT algorithm, JPEG standard algorithm, and the Spiral JPEG algorithm.

Table 2. Comparison between JPEG, 3D-Spiral JPEG, and the two cube for 3D-DCT

Image	PSNR	JPEG (BPP)	Spiral (BPP)	8x8x8 Cube (BPP)	4x4x4 Cube (BPP)
Lena	27.65	0.17	0.13	0.128	0.153
Baboon	37.68	3.38	3.13	3.10	3.05
Flower	29,73	0.127	0.04	0.036	0.046
House	24.41	0.17	0.11	0.11	0.16

The Quantization equation that in the proposed algorithm is derived to deal with the cube formation method. By changing the quantization equation that used in the 3D-Spiral compression algorithm and using the proposed quantization equation. From Table2, it can be noticed that the 8x8x8 cube formation have slightly better bit rate than the 3D-Spiral JPEG algorithms, in the case of low-detail images like Lena, Flower. Furthermore, it can be seen that the 4x4x4 cube formation have slightly better bit rate than the 3D-Spiral JPEG algorithm, in the case of high- detail images like Baboon.

5.1.2. Time Comparison

In terms of time, when using the two methods, we observed that the compression system with 8x8x8 cube formation way is faster than the compression system with 4x4x4 cube formation way and this is due to the large size of blocks that used in the formation of the cube, the more the number of cubes, the less time the compression process. And it also proved by results for different images as in Figure 14.

5.1.3. Blocking artifact effects

As a result of compression, the decompressed images may exhibit various kinds of distortion artifacts [36] such as blocking, blurring and ringing. The human visual sensitivity to different types of artifacts is very different. The blocking effect is usually the most significant among them, especially at low bit rate compression. The reduced blocking artifact effect in the proposed algorithm can be illustrated in Figure 16.

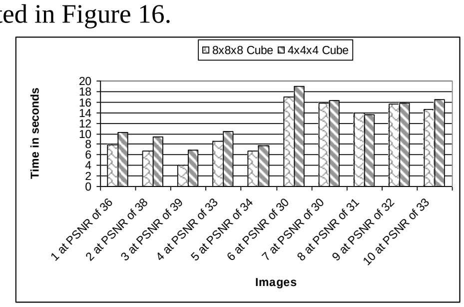
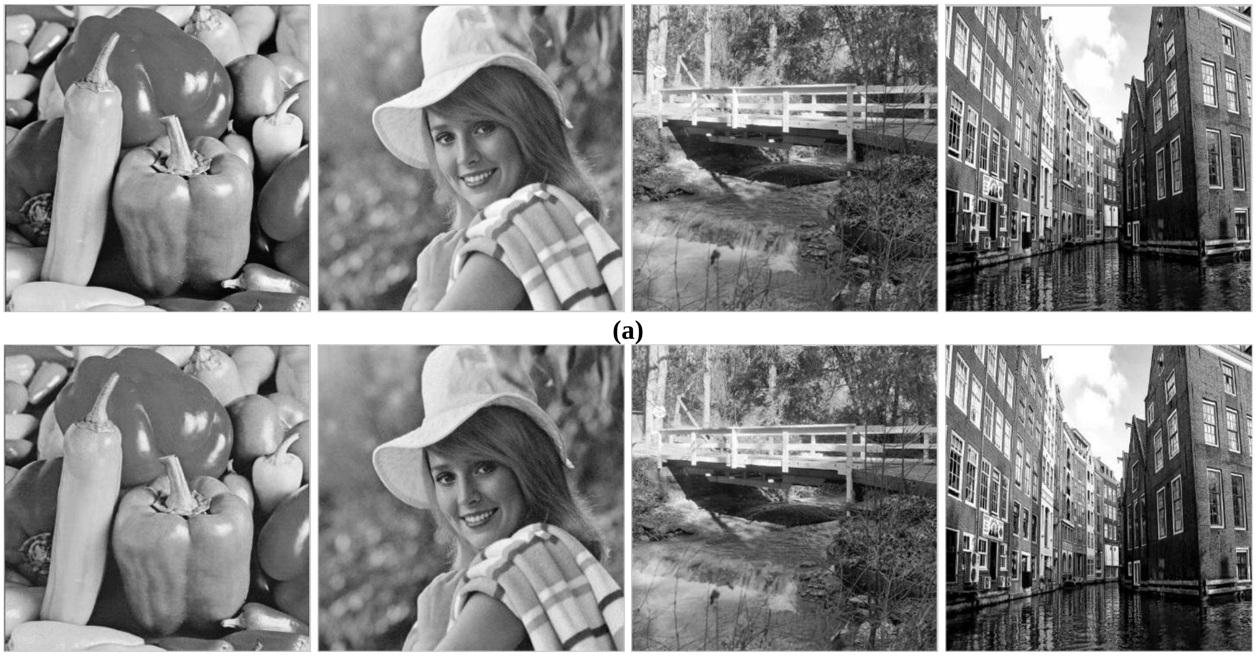


Figure 14. Time comparison at different images for the two proposed ways of compression

Table 1. Comparison between JPEG and the proposed 3D-DCT algorithm (8x8x8 and 4x4x4 cube)

Image	Low/High N= 8 or 4	PSNR	JPEG (BPP)	8x8x8 Cube (BPP)	4x4x4 Cube (BPP)
Lena	N=8	30	0.22	0.21	0.24
		27	0.165	0.11	0.13
		26	0.151	0.085	0.11
Cameraman	N=8	30	0.20	0.18	0.21
		27	0.15	0.10	0.13
		25	0.14	0.06	0.09
Pepper	N=8	28	0.17	0.14	0.17
		26	0.15	0.09	0.12
		25	0.13	0.07	0.10
Elaine	N=8	30	0.22	0.21	0.26
		28	0.16	0.12	0.16
		26	0.13	0.07	0.10
Barbara	N=8	29	0.57	0.56	0.63
		27	0.42	0.39	0.46
		25	0.27	0.25	0.30
Baboon	N=4	35	2.79	2.59	2.50
		33	2.32	2.17	2.11
		30	1.63	1.56	1.52
		28	1.18	1.16	1.14
Bridge	N=4	37	3	2.87	2.78
		35	2.57	2.48	2.39
		32	1.87	1.84	1.81
Hildebrandt	N=4	37	3.04	3.02	2.76
		35	2.60	2.64	2.38
		33	2.15	2.22	2
City	N=4	37	3.28	3.15	2.90
		35	2.88	2.78	2.55
		33	2.46	2.40	2.20
Cartoon	N=4	37	2.73	2.59	2.42
		35	2.25	2.20	1.99
		33	1.73	1.74	1.55



(a)

(b)

Figure 15. (a) Original Images (Pepper, Elaine, Bridge, City), (b) Decompressed images (Pepper at PSNR=33.7 dB, Elaine at PSNR= 32.4 dB, Bridge at PSNR= 31.2 dB, City at PSNR=30.6)

(a) lena (Proposed) (b) lena (JPEG) (c) Cameraman (Proposed) (d) Cameraman (JPEG)

Figure 16. Comparisons of (a) Proposed-Lena: PSNR=27 for 0.11bpp, (b) JPEG-Lena: PSNR=27 for 0.165bpp, (c) Proposed-Cameraman: PSNR=30 for 0.18bpp, (d) JPEG-Cameraman: PSNR=30 for 0.20bpp

5.2. 2-DWT Based Classification

In order to verify the effectiveness of 2D-DWT- based classification step in the proposed algorithm, the system is tested using 100 different images (50 of low details and 50 of high details).

The resulting error percentage of the classification results has not exceeded 2%.

Figure 17 and Figure 18 show examples of images that used in the testing (The original images after and before the 3D-DWT decomposition for low details and high details types).

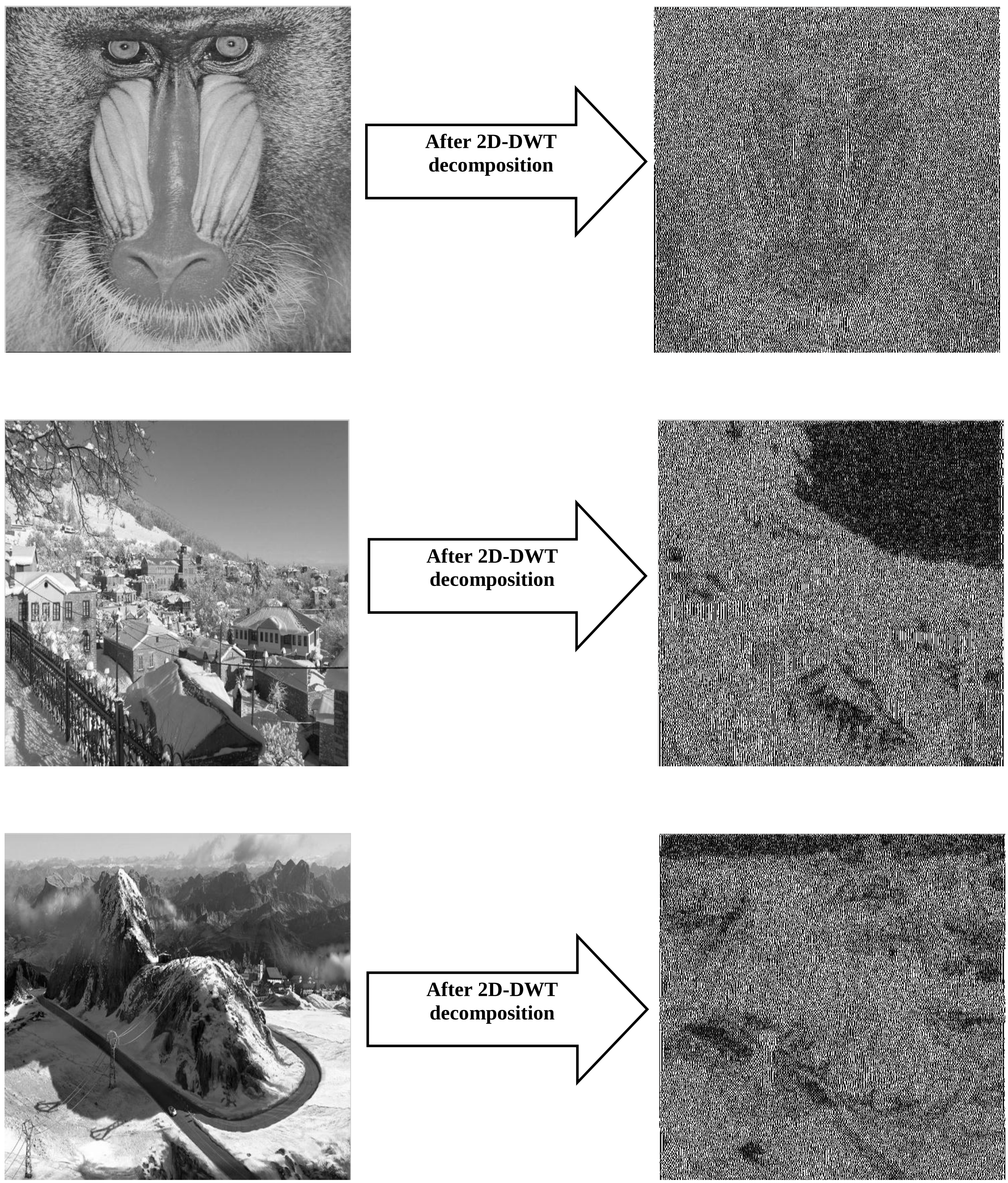


Figure 17. Some high Details original images and the DWT decomposition output

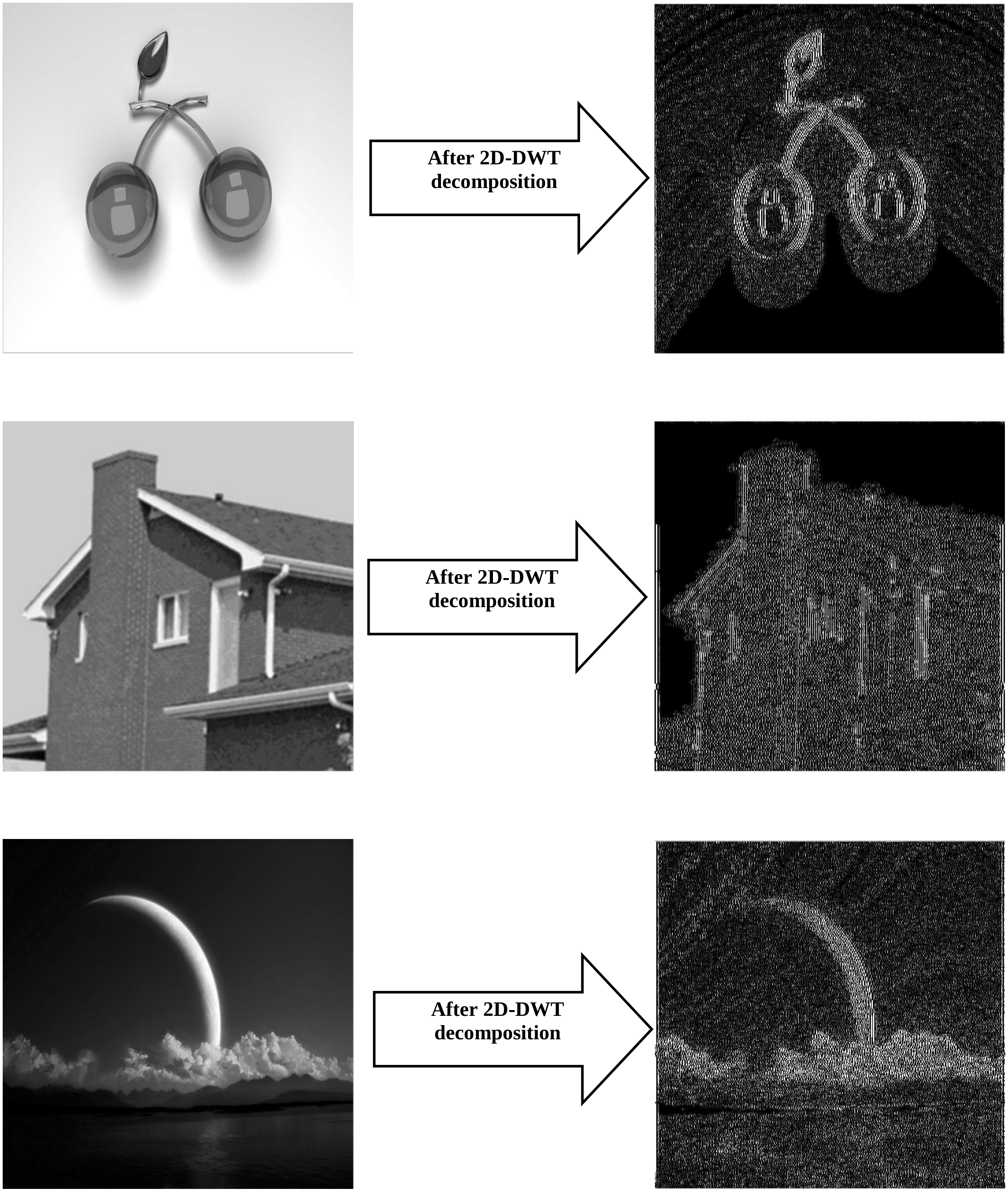


Figure 18. Some low details original images and the DWT decomposition output

5.3. Parallel Computation

5.3.1. SPMD

The proposed algorithm was tested using several test images of dimension 512 x512. The results are displayed in Table 3 and Figure 19 in terms of Q (quality of compression), the consumption time in terms of (serial time, SPMD time, and CUDA time) in seconds and the efficiency of parallelization.

5.3.2. CUDA Parallelization

The CUDA parallelization was tested on some phases of the algorithm (phase 1: 3D cube formation, phase 2: 3D-DCT-quantization and phase 3: proposed scanning). The results are displayed in Table 3, Tables 4 and Figure 20 in terms of Q (quality of compression), the consumption time in seconds and the efficiency of parallelization.

Table 3. Comparison between the execution time of the serial proposed algorithm, the parallel SPMD with proposed algorithm, and the parallel CUDA with the executed four phases of the algorithm

	Images											
	Baboon			Lena			Cameraman			Pepper512		
Q	**Serial Time**	**SPMD Time**	**CUDA Time**	**Serial Time**	**SPMD Time**	**CUDA Time**	**Serial Time**	**SPMD Time**	**CUDA Time**	**Serial Time**	**SPMD Time**	**CUDA Time**
10	2.95	1.629	2.306	2.335	1.412	1.750	2.748	1.351	2.163	2.448	1.401	1.863
20	5.988	2.505	5.344	3.516	1.859	2.931	3.393	1.768	2.808	3.614	1.805	3.029
30	9.132	3.343	8.488	4.546	2.173	3.961	4.325	2.104	3.740	4.681	2.090	4.096
40	11.932	4.122	11.288	5.535	2.614	4.950	5.210	2.439	4.625	5.668	2.413	5.083
50	14.552	5.023	13.908	6.424	2.911	5.839	6.008	2.807	5.423	6.588	2.718	6.003
60	17.569	5.868	16.925	7.529	3.168	6.944	6.910	3.100	6.325	7.731	2.944	7.146
70	22.064	7.059	21.420	9.129	3.790	8.544	8.236	3.588	7.651	9.455	3.548	8.870
80	29.717	9.247	29.073	12.115	4.758	11.530	10.561	4.430	9.976	12.809	4.380	12.228
90	45.569	13.242	44.925	20.303	7.221	19.718	16.788	6.346	16.203	23.153	6.904	22.568

Table 4. Comparison between the time of the serial algorithm and the parallel GPU CUDA for specific phases of the proposed algorithm

Time	**Image Size**								
	256 x 256			**512 x 512**			**1024 x 1024**		
	Phase 1	**Phase 2**	**Phase 3**	**Phase 1**	**Phase 2**	**Phase 3**	**Phase 1**	**Phase 2**	**Phase 3**
Serial Time	0.004	0.0365	0.1242	0.0176	0.1458	0.4468	0.0669	0.5669	1.7777
CUDA Parallel Time	0.0049	0.0074	0.0112	0.0056	0.0101	0.0103	0.0067	0.0123	0.0104

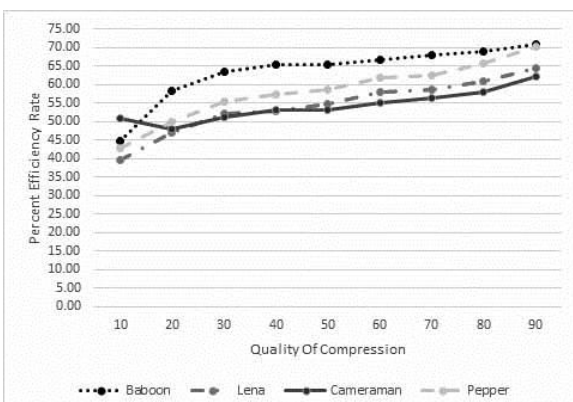


Figure 19. The efficiency of SPMD for different quality

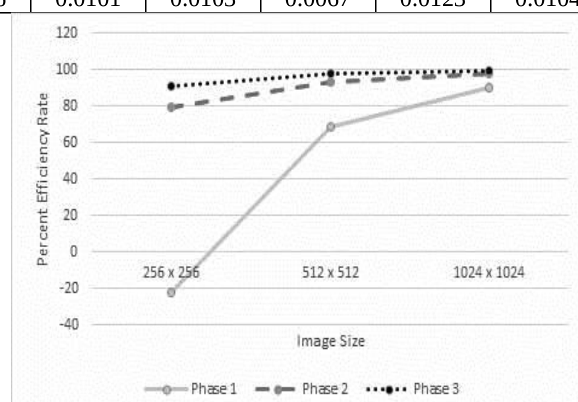


Figure 20. The efficiency of GPU CUDA for different image sizes

6. CONCLUSION

In this paper a new image compression technique that uses 3D-DCT and relies on 2D-DWT based classification has been proposed. The proposed technique has two cube formation methods (8x8x8 and 4x4x4 cube formation), to be used depending on the image type that classified by the 2D-DWT Classification Technique. Several images have been used to test the proposed algorithms. The results show that the proposed algorithm outperforms previous compression methods in terms of Peak-Signal-to-Noise Ratio (PSNR) with a lower bit rate and it has less blocking artifacts than JPEG compression. To overcome the time consumption problem, this paper proposed two techniques with parallel computation and the results show the time consumption is decreased.

7. REFERENCES

[1] R. C. Gonzalez and R. E. Woods, Digital image processing, 3rd ed. Upper Saddle River, N.J: Prentice Hall, pp. 547-635, 2008.

[2] K. Sayood, Introduction to data compression, 3rd ed. Amsterdam; Boston: Elsevier, pp. 6-10, 2006.

[3] K. Sayood, Introduction to data compression, 3rd ed. Amsterdam; Boston: Elsevier, pp. 1-5, 2006.

[4] M. Gupta, & A. K. Garg, "Analysis of Image Compression Algorithm Using DCT," International Journal of Engineering Research and Applications (IJERA), vol. 2, Issue 1, pp.515-521, 2012.

[5] G. K. Wallace, "The JPEG still picture compression standard," IEEE Transactions on Consumer Electronics, vol. 38, no. 1, pp. xviii-xxxiv, Feb. 1992.

[6] T. Sikora, "MPEG digital video-coding standards," IEEE Signal Processing Magazine, vol. 14, no. 5, pp. 82-100, Sep. 1997.

[7] S. An and C. Wang, "A computation structure for 2-D DCT watermarking," IEEE International Midwest Symposium on, pp. 577-580, 2009.

[8] Rajkumar Buyya, High Performance Cluster Computing: Programming and Applications, Prentice Hall PTR,Vol. 2, pp. 4-27,1999.

[9] Sanjay Saxena, Shiru Sharma and Neeraj Sharma," Parallel Image Processing Techniques, Benefits and Limitations", Research Journal of Applied Sciences, Engineering and Technology,pp. 223-238, 2016.

[10] R. Farias, R. Marroquim, and E. Clua , "Parallel Image Segmentation Using Reduction-Sweeps on Multicore Processors and GPUs", XXVI Conference on Graphics, Patterns and Images (SIBGRAPI),pp. 139-146, 5-8 Aug. 2013.

[11] S. Bozóki, S. J. P. Westen, R. L. Lagendijk and J. Biemond," Parallel Algorithms for MPEG Video Compression with PVM", International Conference HPCN challenges in telecomp and telecom,1996.

[12] Ishfaq Ahmad, Yong He, Ming L. Liou," Video compression with parallel processing",Journal of Parallel computing in image and video processing,Vol. 28,pp. 1039-1078,2002.

[13] K. S. Priyadarshini, G. S. Sharvani," A Survey on Parallel Computing of Image Compression Algorithms", Proceedings of the International Conference, Computational Systems for Health & Sustainability,pp. 178-183, April, 2015.

[14] B. R. Naidu, M. S. P. Babu," A novel framework for JPEG image compression using baseline coding with parallel process",IEEE International Conference on Computational Intelligence and Computing Research,pp. 1-7, 18-20 Dec. 2014.

[15] Jun Wang , H. K. Huang,"Three-dimensional medical image compression using a wavelet transform with parallel computing", Proceedings of the SPIE, Volume 2431, pp. 162-172 ,1995.

[16] K. Fatima, V. G. Sarvepalli, and Z. N. Nakhi, "A Novel Architecture for the Computation of the 2D-DWT," International Conference on Computational Intelligence and Security, pp. 531-535,2007.

[17] Z.-N. Li and M. S. Drew, Fundamentals of multimedia. Upper Saddle River, NJ: Pearson Prentice Hall, pp. 225-280, 2004.

[18] M. Ergen, Mobile Broadband. Boston, MA: Springer US, pp. 130-132,2009.

[19] J.-I. Jeon and H.-S. Kang, "3D DCT Based Compression Method for Integral Images," in Advances in Visual Computing, vol. 6454, pp. 659-668,2010.

[20] A. Mehanna, A. Aggoun, O. Abdulfatah, M. R. Swash, and E. Tsekleves, "Adaptive 3D-DCT based compression algorithms for integral images," Broadband Multimedia Systems and Broadcasting (BMSB), pp. 1-5, 2013.

[21] M. C. Lee, R. K. W. Chan, and D. A. Adjeroh, "Quantization of 3D-DCT Coefficients and Scan Order for Video Compression," Journal of Visual Communication and Image Representation, vol. 8, no. 4, pp. 405-422, Dec. 1997.

[22] T. Haiyan, S. Wenbang, G. Bingzhe, and Z. Fengjing, "Research on Quantization and Scanning Order for 3-D DCT Video Coding," Computer Science and Electronics Engineering (ICCSEE) ,vol. 1, pp. 200-204,2012.

[23] X. Li, A. Dick, C. Shen, A. van den Hengel, and H. Wang, "Incremental Learning of 3D-DCT Compact Representations for Robust Visual Tracking," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 35, no. 4, pp. 863-881, Apr. 2013.

[24] Jingbing Li, Huaiqiang Zhang, and Chunhua Dong, "Multiple video watermarks based on 3D-DWT and 3D-DCT robust to geometrical attacks," Automatic Control and Artificial Intelligence (ACAI), pp. 1372-1376, 2012.

[25] Joachimiak, M., Rusanovskyy, D., Hannuksela, M. M., & Gabbouj, M. Multiview., "3D video denoising in sliding 3D DCT domain," In Signal Processing Conference (EUSIPCO), Proceedings of the 20th European,2012.

[26] Hualing Mao, Guorui Feng, Xinpeng Zhang, and Heng Yao, "A robust and fast video fingerprinting based on 3D-DCT and LSH,", International Conference on Multimedia Technology (ICMT), pp. 108-111, 2011.

[27] X. Li and B. Furht, "An Approach to Image and Video Compression using Three-Dimensional DCT," Proceedings of the Visual 2003 Conference, Miami, Florida, September 2003.

[28] M. A. Engin and B. Cavusoglu, "New Approach in Image Compression: 3D Spiral JPEG," IEEE Communications Letters, vol. 15, no. 11, pp. 1234-1236, Nov. 2011.
[29] L. Blaszak and M. Domanski, "Spiral coding order of macroblocks with applications to SNR-scalable video compression," International Conference on Image Processing, Vol. 3, pp. III-688, 2005.
[30] A. Mulla, J. Baviskar, A. Baviskar, and C. Warty, "Image compression scheme based on zig-zag 3D-DCT and LDPC coding," International Conference on Advances in Computing, Communications and Informatics (ICACCI), pp. 2380-2384, 2014.
[31] A. Karami, S. Beheshti, and M. Yazdi, "Hyperspectral image compression using 3D discrete cosine transform and support vector machine learning," International Conference on Information Science, Signal Processing and their Applications (ISSPA), pp. 809-812, 2012.
[32] Z.-N. Li and M. S. Drew, Fundamentals of multimedia. Upper Saddle River, NJ: Pearson Prentice Hall, pp. 281-315, 2004.
[33] H. Palangi, A. Ghafari, M. Babaie-Zadeh, and C. Jutten, "Image Coding and Compression with Sparse 3D Discrete Cosine Transform," Independent Component Analysis and Signal Separation: 8th International Conference, vol. 5441, pp.532-539, 2009.
[34] R. C. Gonzalez and R. E. Woods, Digital image processing, 3rd ed. Upper Saddle River, N.J: Prentice Hall, pp. 247-275, 2008.
[35] M. Mrak, S. Grgic, and M. Grgic, "Picture quality measures in image compression systems,", EUROCON, vol. 1, pp. 233-236, 2003.
[36] Leida Li, Weisi Lin, and Hancheng Zhu, "Learning Structural Regularity for Evaluating Blocking Artifacts in JPEG Images," IEEE Signal Processing Letters, vol. 21, no. 8, pp. 918-922, Aug. 2014.