Image compression based on 2D Discrete Fourier Transform and matrix minimization algorithm

RASHEED, Mohammed H, SALIH, Omar M, SIDDEQ, Mohammed M and RODRIGUES, Marcos <http://orcid.org/0000-0002-6083-1303>

Array 6 (2020) 100024

Contents lists available at ScienceDirect

Array

journal homepage: www.elsevier.com/journals/array/2590-0056/open-access-journal

Image compression based on 2D Discrete Fourier Transform and matrix minimization algorithm

Mohammed H. Rasheed [a], Omar M. Salih [a], Mohammed M. Siddeq [a,*], Marcos A. Rodrigues [b]

[a] Computer Engineering Dept., Technical College/Kirkuk, Northern Technical University, Iraq
[b] Geometric Modeling and Pattern Recognition Research Group, Sheffield Hallam University, Sheffield, UK

ARTICLE INFO

Keywords:
DFT
Matrix minimization algorithm
Sequential search algorithm

ABSTRACT

In the present era of the internet and multimedia, image compression techniques are essential to improve image and video performance in terms of storage space, network bandwidth usage, and secure transmission. A number of image compression methods are available with largely differing compression ratios and coding complexity. In this paper we propose a new method for compressing high-resolution images based on the Discrete Fourier Transform (DFT) and Matrix Minimization (MM) algorithm. The method consists of transforming an image by DFT yielding the real and imaginary components. A quantization process is applied to both components independently aiming at increasing the number of high frequency coefficients. The real component matrix is separated into Low Frequency Coefficients (LFC) and High Frequency Coefficients (HFC). Finally, the MM algorithm followed by arithmetic coding is applied to the LFC and HFC matrices. The decompression algorithm decodes the data in reverse order. A sequential search algorithm is used to decode the data from the MM matrix. Thereafter, all decoded LFC and HFC values are combined into one matrix followed by the inverse DFT. Results demonstrate that the proposed method yields high compression ratios over 98% for structured light images with good image reconstruction. Moreover, it is shown that the proposed method compares favorably with the JPEG technique based on compression ratios and image quality.

1. Introduction

The exchange of uncompressed digital images requires considerable amounts of storage space and network bandwidth. Demands for efficient image compression result from the widespread use of the Internet and data sharing enabled by recent advances in digital imaging and multimedia services. Users are creating and sharing images with increased size and quantity and expect quality image reconstruction. It is clear that sharing multimedia-based platforms such as Facebook and Instagram lead to widespread exchange of digital images over the Internet [1]. This has led to efforts to improve and fine-tune present compression algorithms along with new algorithms proposed by the research community to reduce image size whilst maintaining the best level of quality. For any digital image, it can be assumed that the image in question may have redundant data and can be neglected to a certain extent. The amount of redundancy is not fixed, but it is an assumed quantity and its amount depends on many factors including the requirements of the application to be used, the observer (viewer) or user of the image and the purpose of its use [2,3]. Basically, if the purpose of an image is to be seen by humans then we can assume that the image can have a variable high level of redundant data. Redundant data in digital images come from the fact that pixels in digital images are highly correlated to a level where reducing this correlation cannot be noticed by the human eye (Human Visual System) [4,5]. Consequently, most of these redundant, highly correlated pixels can be removed while maintaining an acceptable level of human visual quality of the image. Therefore, in digital images the Low Frequency Components (LFC) are more important as they contribute more to define the image contents than High Frequency Components (HFC). Based on this, the intension is to preserve the low frequency values and shorten the high frequency values by a certain amount, in order to maintain the best quality with the lowest possible size [6,7].

Image frequencies can be determined through a number of transformations such as the Discrete Cosine Transform (DCT), Discrete Wavelet Transform (DWT) and Discrete Fourier Transform (DFT) [8]. In this study we will use DFT as a first step in the process to serialize a digital image for compression. Since its discovery, the DFT has been used in the

* Corresponding author.
E-mail addresses: mhrjabary@gmail.com (M.H. Rasheed), omar.alsabaawi@gmail.com (O.M. Salih), mamadmmx76@gmail.com (M.M. Siddeq), M.Rodrigues@shu.ac.uk (M.A. Rodrigues).

https://doi.org/10.1016/j.array.2020.100024
Received 21 November 2019; Received in revised form 11 February 2020; Accepted 6 March 2020
Available online 8 March 2020

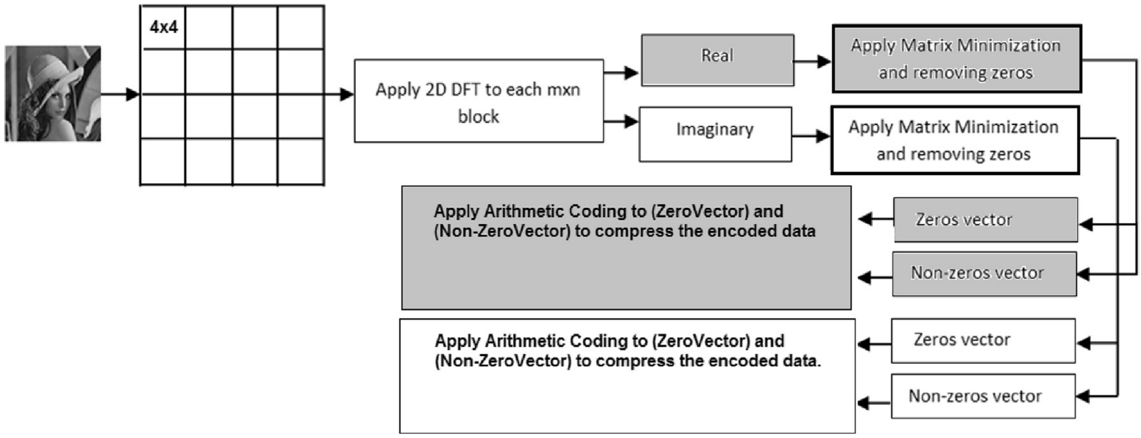


Fig. 1. The proposed compression method.

Matrix = 4x4	15 18 26 48 14 19 27 31 16 21 32 31 16 25 35 32	Convert Matrix to frequency domain by DFT	Real= (Matrix): 406 -59 -44 -59 7 10 -21 0 8 5 -14 5 7 0 -21 10
			Imaginary= (Matrix): 0 59 0 -59 17 14 3 -26 0 21 0 -21 -17 26 -3 -14

Fig. 2. DFT applied to a 4 × 4 matrix of data.

Real=	406 -59 -44 -59 7 10 -21 0 8 5 -14 5 7 0 -21 10	After quantization (real and imaginary is divided by Q = 20)	Qr=	20 -3 -2 -3 0 1 -1 0 0 0 -1 0 0 0 -1 1
Imaginary=	0 59 0 -59 17 14 3 -26 0 21 0 -21 -17 26 -3 -14		Qi=	0 3 0 -3 1 1 0 -1 0 1 0 -1 -1 1 0 -1

Fig. 3. Quantization and rounding off the real and imaginary components.

LFC-Matrix = [20,... etc] (DC value for each block 4x4 saved in LFC-Matrix)

HFC_{Real} = [0 -3 -2 -3 0 1 -1 0 0 0 -1 0 0 0 -1 1]
HFC_{Imag}=[0 3 0 -3 1 1 0 -1 0 1 0 -1 -1 1 0 -1]

Fig. 4. Each block (4 × 4) is divided to real and imaginary matrices (after applying DFT). The real matrix contains DC value at first location, these DC values are saved in a new matrix. The rest of high-frequency coefficients are saved in a different matrix as shown in contents of the LFC-Matrix, HFC_{Real} and HFC_{Imag}.

field of image processing and compression. The DFT is used to convert an image from the spatial domain into frequency domain, in other words it allows us to separate high frequency from low frequency coefficients and neglect or alter specific frequencies leading to an image with less information but still with a convenient level of quality [8–10].

We propose a new algorithm to compress digital images based on the DFT in conjunction with the Matrix Minimization method as proposed in Ref. [10,11]. The main purpose of matrix minimization is to reduce High Frequency Components (HFC) to 1/3 of its original size by converting each three items of data into one, a process that also increases redundant coefficients [11,12]. The main problem with Matrix Minimization is that it has a large probability data called Limited-Data [13,14,16]. Such probabilities are combined within the compressed file as indices used later in decompression.

Our previous research [13,14] used the DCT combined with Matrix Minimization algorithm yielding over 98% compression ratios for structured light images and 95% for conventional images. The main justification to use DFT in the proposed method is to demonstrate that the Matrix Minimization algorithm is very effective in connection with a discrete transform and, additionally, to investigate the DFT for image compression.

The contribution of this research is to reduce the relatively large probability table to two values only, minimum and maximum, rather than keeping the entire lookup table (referred to as Limited-Data in our previous research [10,11,12and13]). The main reason is to increase compression ratios by reducing the size of the compressed file header.

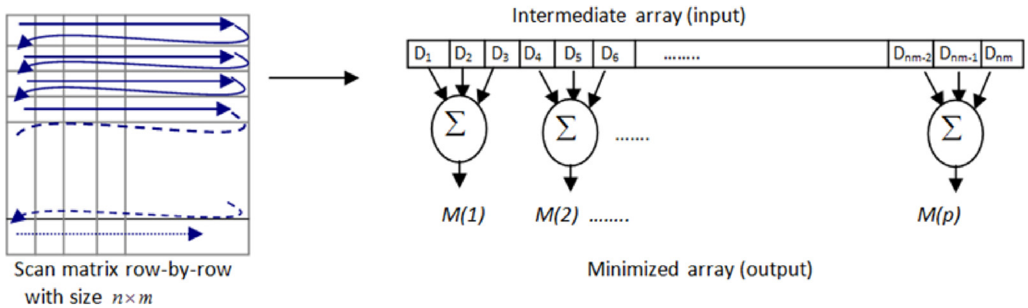


Fig. 5. The Matrix Minimization method for an m x n matrix [10–12].

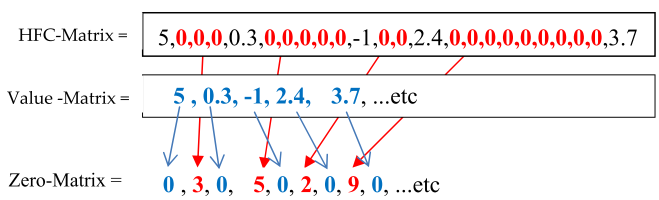


Fig. 6. Separating zeros and nonzero from HFC matrix and coding zero and nonzero values into Zero and Value matrices.

The proposed compression algorithm is evaluated and analyzed through measures of compression ratios, RMSE (Root Mean Square Error) and PSNR (Peak Signal-to-Noise Ratio). It is demonstrated that the proposed method compares well with the popular JPEG technique.

2. The proposed compression algorithm

The proposed compression method is illustrated in Fig. 1. Initially, an original image is subdivided into non-overlapping blocks of size M x N pixels starting at the top left corner of the image. The Discrete Fourier transform (DFT) is applied to each M x N block independently to represent the image in the frequency domain yielding the real and imaginary components. The Matrix Minimization algorithm is applied to each component and zeros are removed. The resulting vectors are subjected to Arithmetic coding and represent the compressed data.

To illustrate the process for each M x N (M = N = 4) block in the original image, we represent a 4 × 4 block in Fig. 2 below:

A uniform quantization is then applied to both parts, which involves dividing each element by a factor called quantization factor Q followed by rounding the outcomes which results in an increase of high frequency coefficients probability thus reducing the number of bits needed to represent such coefficients. The result of this operation is that the compression ratio increases. Fig. 3 illustrate the quantization and rounding off steps. For more information, the uniform quantization (Qr and Qi) are selected heuristically.

Up to this point, two matrices (Qr and Qi) have been generated per block representing the real and the imaginary parts respectively. Regarding the real part, all low coefficient values (i.e. the DC values) are detached and saved into a new matrix called Low Frequency Coefficients (LFC-Matrix) and its substituted with a zero value in the quantized matrix. It is important to note that DC values are only found in the real parts which highly contribute to the main details and characteristics of the image. The generated LFC-Matrix size consists of all the DC values of the entire image can be considered small compared to all other High Frequency Coefficients (HFC-Matrix) and can be represented with few bytes. Fig. 4 illustrates the content of the generated three matrices.

Since the size of the LFC-Matrix is small compared to HFC-Matrices, it is very obvious that HFC matrices for both real and imaginary parts need to be reduced to get a reasonable compression. Therefore, the algorithm called Matrix-Minimization suggested by Siddeq and Rodrigues [10] is applied. The algorithm is used to reduce the size of HFC matrices by contracting every three coefficients to a single equivalent value, which can be traced back to their original values in the decompression phase. The contraction is performed on each three consecutive coefficients using Random-Weight-Values. Each value is multiplied by a different random number (Ki) and then their summation is found, the value generated is considered a contracted value of the input values. Fig. 5 illustrates the Matrix Minimization applied to M x N matrix [11,12].

It is important to note that in the decompression phase a search algorithm is required to find the three original values that are used to find the contracted value, therefore, the minimum and maximum values of the m x n block are stored. The idea behind this is to limit the range of values required to recover the original three values that made the contracted value hence increase the speed of the search algorithm at decompression stage.

Because in previous work the range of the search space are limited in the array for easy searching and this was encoded in the header file to be used at decompression stage. However, it is possible that complex images may generate large arrays which, in turn, will impair compression (make it more computationally demanding). For this reason, we suggested another method in this paper using DFT and reduced limited search area (i.e. search area contains just two values [MIN, MAX]). Such bounding makes searching for the sought values easier and faster. Any further detailed information about Matrix Minimization can be found in

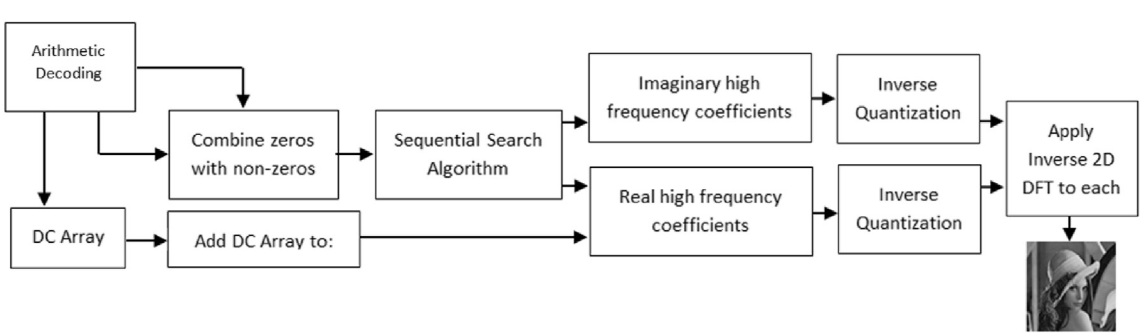


Fig. 7. Decompression steps.

260 KB Quantization value=10 | 138.2 KB Quantization value=25 | 88.1 KB Quantization value=45

(a) Decompressed Lena image, dimension = 1024 x 1024

201 KB Quantization value=25 | 108.4 KB Quantization value=60 | 71.4 KB Quantization value=100

(b) Decompressed Lion image, dimension = 1200 x 1200

228 KB Quantization value=10 | 91.7 KB Quantization value=30 | 47.8 KB Quantization value=60

(c) Decompressed Apple image, dimension = 1200 x 1200

Fig. 8. (a), (b) and (c) Lena, Lion and Apple images status are compressed by our proposed method using different quantization values.

references [11,12,16]. These three references show with examples how the Matrix Minimization works with keys and how the limited search is used for decoding.

After the Matrix-Minimization algorithm has been applied, the produced HFC-Matrix for both real and imaginary parts are examined and it is possible to see a high probability in the number of zero values than any other values in the matrix. Therefore, separating zero from non-zero values will remove redundant data and hence increase the efficiency of the arithmetic coding compression [9,10,13,14].

The implementation of the method is by isolating all zero values from the matrix while preserving all non-zero values in a new array called **Value Matrix**. The total number of zeros removed between each non-zero value in the **HFC-Matrix** is counted during the process. A new array called **Zero Matrix** is then created in which we append a zero value whenever we have a non-zero value at the same index in the original HFC-Matrix followed by an integer that represents the total number of zeros between any two non-zero values. Fig. 6 demonstrates the process of separating zeros and non-zero values [14–16].

The zero values in the Zero-Matrix reflect the actual non-zero values in sequences in the original matrix. Likewise, the integer values reflect the total number of zeros that come thereafter. Finally, the two matrices are ready for compression by a coding method which in our case is arithmetic coding [6,7]. It is important to note that the proposed method described above is also applied to the LFC-Matrix which contains the low

Fig. 9. (a), (b) and (c) Boeing 777, Girl and Baghdad colour images are compressed by our proposed method using different quantization values.

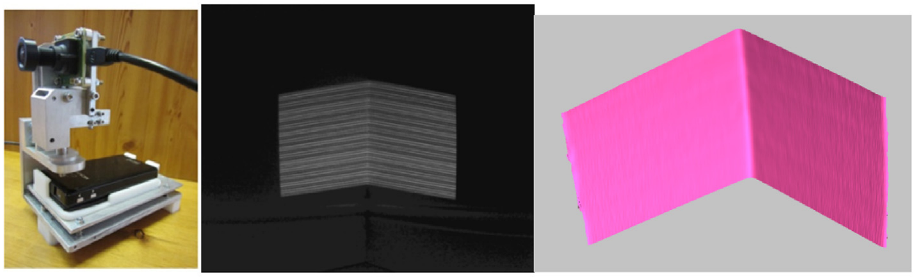
frequency coefficients values of the real part. Up to this point, the Value-Matrix and Zero-Matrix in our case are considered headers and used in the decompression process to regenerate the original HFC and LFC matrices.

3. The decompression algorithm

The decompression algorithm is a counter compression operation which performs all functions of the compression but in reverse order. The steps to decompression start by decoding the LFC-Matrix, Value-Matrix and Zero-Matrix using arithmetic decoding followed by reconstructing a unified array based on Value and Zero matrices and reconstruct the HFC-Matrix for both parts. Siddeq and Rodrigues proposed a novel algorithm called Sequential Search Algorithm [10–13], which is based on three pointers working sequentially to regenerate the three values that constitute the contracted values with assistance of the MIN and MAX values which were preserved during the compression process. The MIN and MAX values are considered to be the limited space search values used to restore the actual HFC for both parts (real and imaginary) [14–18]. Finally, an inverse quantization and DFT is applied to each part to reconstruct the compressed digital image. Fig. 7 illustrates the decompression steps.

4. Experimental results

Experimental results shown here demonstrate the efficacy of the



(a) (b) (c)

Fig. 10. (a) The 3D Scanner developed by the GMPR group, (b) a 2D picture captured by the camera, (c) 2D image converted into a 3D surface patch.

Corner (1280x1024) Face1 (1392x1040) Face2 (1392x1040)

Fig. 11. Original 2D images with different dimensions used by our proposed compression method.

Table 1
Results for grey images.

Image	Image Size (MB)	Quantization	After Compression (KB)	(Bit/ Pixel) bpp	RMSE	PSNR
Lena	1.0	10	260	0.253	1.2	47.3
		25	138.2	0.134	2.4	44.3
		45	88.1	0.086	3.9	42.2
Lion	1.37	25	201	0.143	2.5	44.1
		60	108.4	0.077	5.0	41.1
		100	71.4	0.05	8.1	39.0
Apples	1.37	10	228	0.162	1.2	47.3
		30	91.7	0.065	2.6	43.9
		60	47.8	0.034	4.6	41.5

proposed compression technique. Our proposed method was implemented in MATLAB R2014a running on an Intel Core i7-3740QM microprocessor (8-CPUs). For clarity, we divide the results into two parts:

- The method applied to general 2D images of different sizes and assess their visual quality with RMSE [1,3]. Also, we applied Peak Signal-to-Noise Ratio (PSNR) for measuring image quality. This measurement widely used in digital image processing [23]. Tables 1 and 2 show the first part of results by applying the proposed compression/decompression method to six selected images whose details are shown in Figs. 8 and 9.
- We apply the proposed compression technique to structured light images (i.e. a type of image used for reconstruct 3D surfaces - see Section 5).

5. Results for structured light images and 3D surfaces

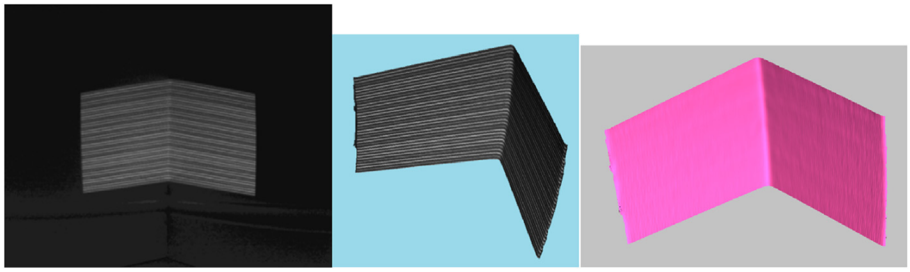
A 3D surface mesh reconstruction method was developed by Rodrigues [8,19] with a team within the GMPR group at Sheffield Hallam University. The working principle of the 3D mesh scanner is that the scene is illuminated with a stripe pattern whose 2D image is then captured by a camera. The relationship between the light source and the camera determines the 3D position of the surface along the stripe pattern. The scanner converts a surface to a 3D mesh in a few milliseconds by

Table 3
Compressed 2D structured light images.

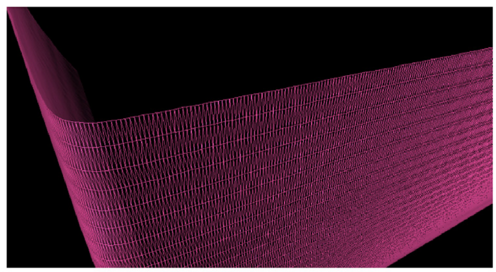
Image	Image Size (MB)	Quantization	After Compression (KB)	(bit/ Pixel) bpp	RMSE	PSNR
Corner	1.25	60	35.4	0.027	4.7	41.4
		100	17.8	0.013	15.5	36.2
Face1	1.37	100	34.0	0.024	8.4	38.8
		160	18.1	0.012	11.5	37.5
Face2	1.37	50	46.2	0.032	6.7	39.8
		150	20.1	0.014	9.9	38.1

Table 2
Results for colour images.

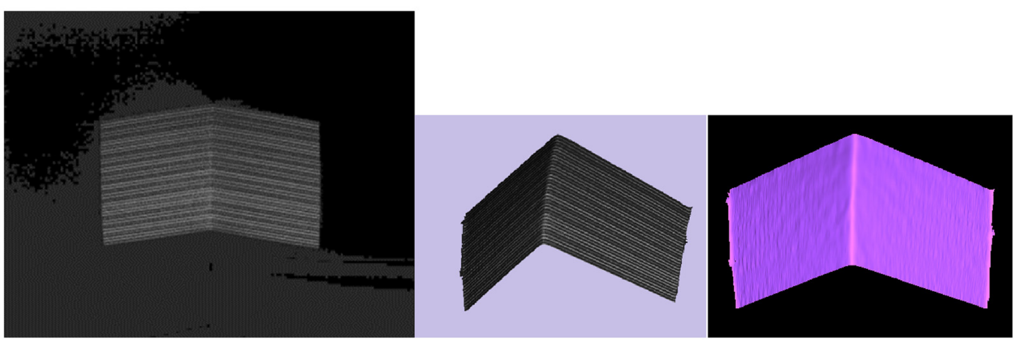
Image	Image Size (MB)	Quantization for each layer in R,G,B	After Compression (KB)	(Bit/Pixel) bpp	RMSE	PSNR
Boeing 777	6.15	10	437.4	0.069	2.1	44.9
		25	182.8	0.029	3.9	42.2
Girl	4.29	10	641.1	0.145	3.9	42.2
		25	315.6	0.071	5.5	40.7
Baghdad	8.58	25	426.3	0.097	4.4	41.6
		35	309.8	0.07	5.6	40.6



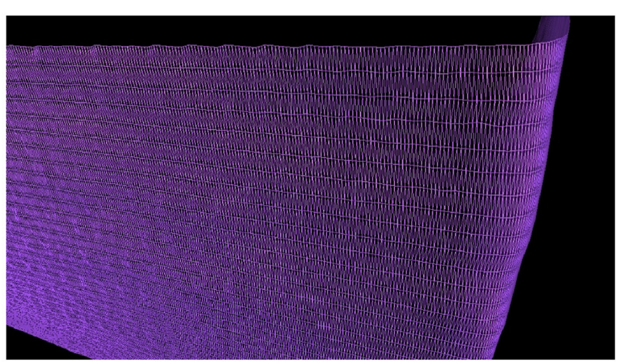
(left) 2D decompressed Corner image with RMSE=4.7, (middle and right) converted the 2D Corner's image to 3D surface by GMPR method at compressed size = 35.4 KB.



(a): Zooming in the decompressed 2D Corner's image (RMSE=4.7) converted to 3D mesh reconstructed successfully by the GMPR method without significant distortions. (Compressed size was 35.4 KB)

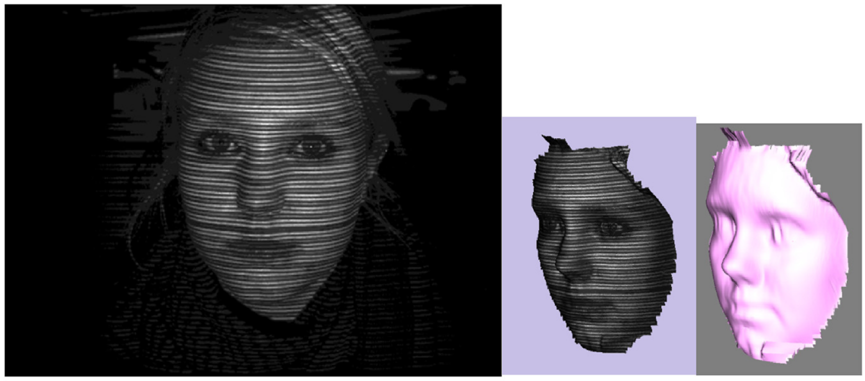


(left) 2D decompressed Corner image with RMSE=15.5, (middle and right) converted the 2D Corner's image to 3D surface by the GMPR method at compressed size = 17.8 KB.

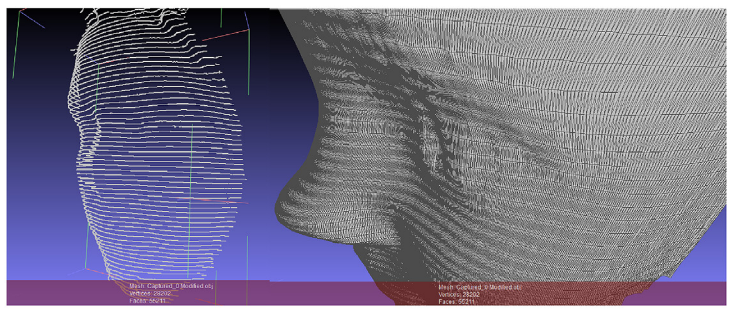


(b): Zooming in to the decompressed 2D Corner's image (RMSE=15.5) converted to 3D mesh reconstructed successfully by the GMPR method without significant distortions at higher compression ratio (compressed size was 17.8 KB)

Fig. 12. (a) and (b): shows the 2D decompressed for Corner's image, that used in 3D application to reconstruct 3D mesh surface. The 3D mesh (3D vertices and triangles) is successfully reconstructed without significant distortion at high compression ratios up to 98.6%.

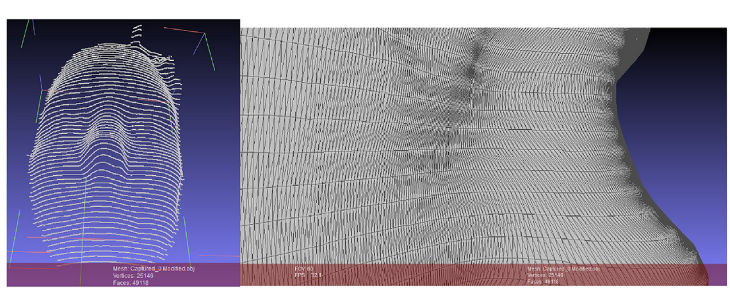


(left) 2D decompressed Face1 image with RMSE=8.4, (middle and right) converted the 2D Face1 image to 3D surface by the GMPR method at compressed size =34 KB.



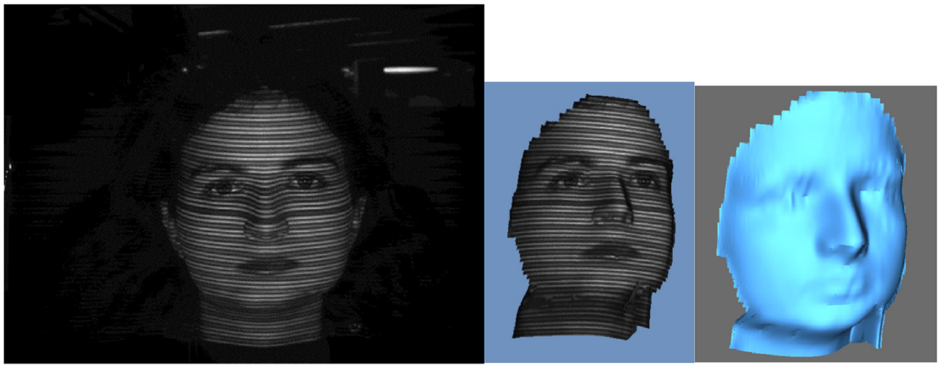
(a): Zooming in the decompressed 2D Face1 image (RMSE=8.4) used to reconstruct 3D vertices and mesh successfully without significant distortion (3D surface visualized by MeshLab software)

(left) 2D decompressed Face1 image with RMSE=11.5, (middle and right) converted the 2D Face1 image to 3D surface by the GMPR method at compressed size =18.1 KB.

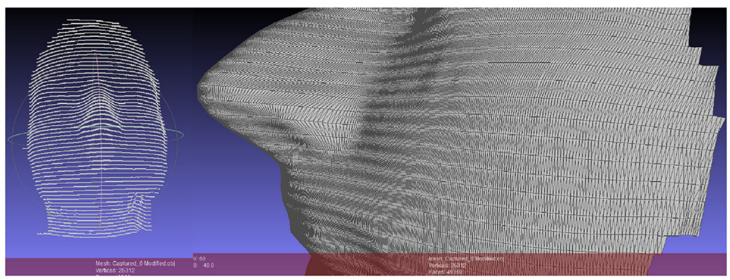


(b): Zooming in the decompressed 2D Face1 image (RMSE=11.5) used to reconstruct 3D vertices and mesh successfully without significant distortion (3D surface visualized by MeshLab software)

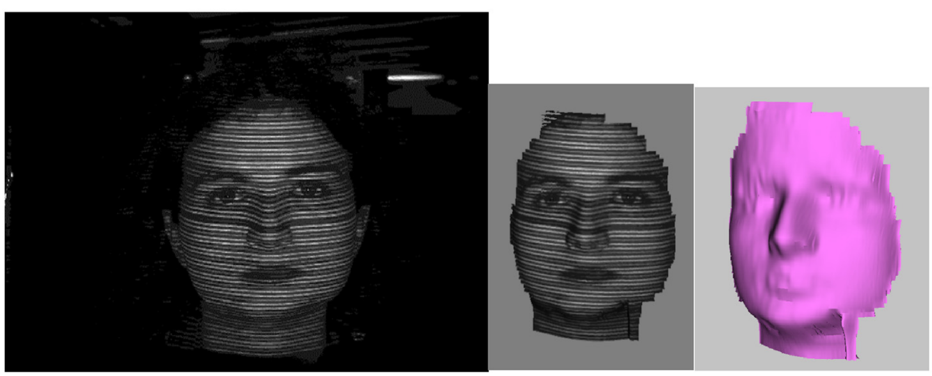
Fig. 13. (a) and (b): shows decompressed for Face1 2D image, that used in the 3D application to reconstruct 3D mesh surface. The 3D mesh is successfully reconstructed without significant distortion at high compression ratios of 98.6%.



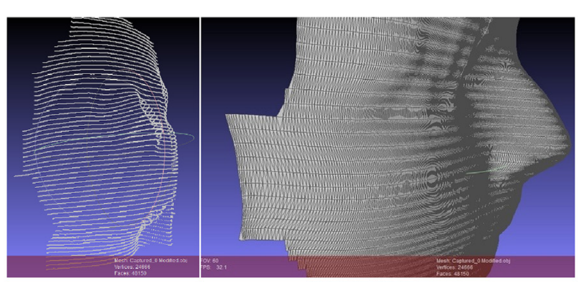
(left) 2D decompressed Face2 image with RMSE=6.7, (middle and right) converted the 2D Face2 image to 3D surface by GMPR group at compressed size =46.2 KB.



(a): Zooming in the decompressed 2D Face2 image (RMSE=6.7) used to reconstruct 3D vertices and mesh successfully without significant distortion (3D surface visualized by MeshLab software)



(left) 2D decompressed Face2 image with RMSE=9.9, (middle and right) converted the 2D Face2 image to 3D surface by GMPR group at compressed size =20.1 KB.



(a): the decompressed 2D Face2 image (RMSE=9.9) used to reconstruct 3D vertices and mesh successfully without significant distortion (3D surface visualized by MeshLab software)

Fig. 14. (a) and (b): shows decompressed for Face2 2D image, that used in 3D application to reconstruct 3D mesh surface. The 3D mesh was successfully reconstructed without significant distortion at high compression ratios of 98.5%.

Table 4
Comparative analysis of using DFT alone and our proposed method (DFT and Matrix Minimization) based on image quality and compressed size.

Image	Size (MB)	Quantization Factor	Compressed size DFT alone		RMSE	PSNR	Compressed Size DFT + MM		RMSE	PSNR
			KB	bpp			KB	bpp		
Conventional images										
Lena	1.0	45	721	0.7	2.1	44.9	88	0.085	3.9	42.2
Lion	1.37	100	808	0.57	3.59	42.5	71	0.05	8.1	39.0
Apples	1.37	60	576	0.41	2.3	44.5	47	0.033	4.6	41.5
Boeing	6.15	25	2200	0.35	1.8	45.5	182	0.028	3.9	42.2
Girl	4.29	25	2350	0.54	3.59	42.5	315	0.071	5.5	40.7
Bagdad	8.58	35	3500	0.4	2.3	44.5	309	0.035	5.6	40.6
Structured light images										
Corner	1.25	100	615	0.48	2.9	43.5	17	0.013	15.5	36.2
Face1	1.37	160	624	0.44	4.8	41.3	18	0.012	11.5	37.5
Face2	1.37	150	508	0.36	4.1	42.0	20	0.014	9.9	38.1

Table 5
Comparative analysis of compression using JPEG and our approach based on image quality and compression size.

Image	Size (MB)	Compressed Size by JPEG		RMSE	PSNR	Compressed Size by DFT + MM		RMSE	PSNR
		KB	bpp			KB	bpp		
Conventional images									
Lena	1.0	64	0.062	1.9	45.3	88	0.085	3.9	42.2
Lion	1.37	56	0.039	8.8	38.6	71	0.05	8.1	39.0
Apples	1.37	48	0.034	3.2	43.0	47	0.033	4.6	41.5
Boeing	6.15	210	0.033	8.7	38.7	182	0.028	3.9	42.2
Girl	4.29	347	0.078	9.8	38.2	315	0.071	5.5	40.7
Bagdad	8.58	279	0.031	3.5	42.6	309	0.035	5.6	40.6
Structured light images									
Corner	1.25	26	0.02	14.3	36.5	17	0.013	15.5	36.2
Face1	1.37	23	0.016	16.5	35.9	18	0.012	11.5	37.5
Face2	1.37	27	0.019	13.1	36.9	20	0.014	9.9	38.1

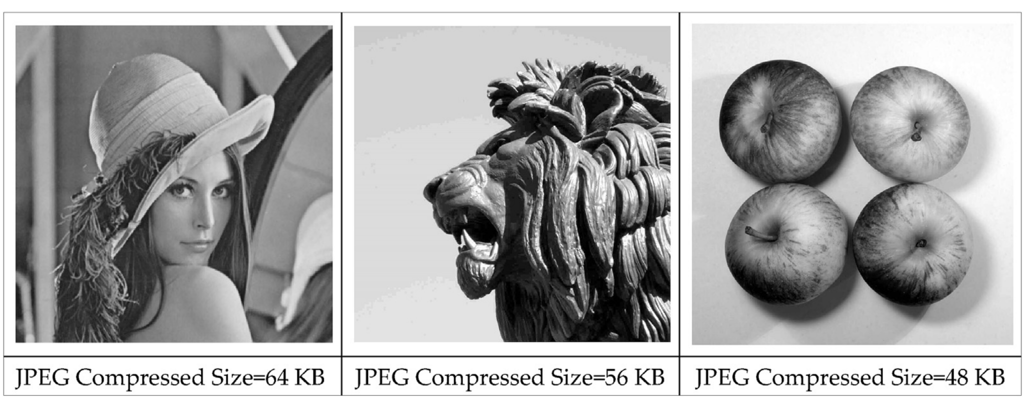


Fig. 15. Compressed and decompressed greyscale images by JPEG, the quality of the decompressed images varies compared with our approach according to RMSE and PSNR.

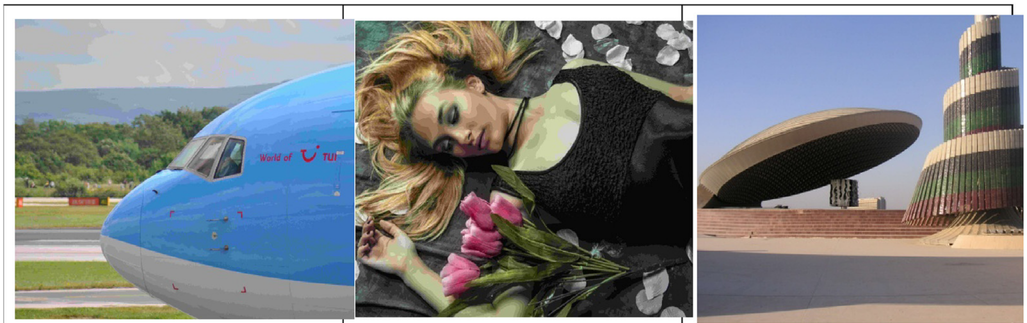
using a single 2D image [19,20] as shown in Fig. 10.

The significance of using such 2D images is that, if the compression method is lossy and results in a noisy image, the 3D algorithms will reconstruct the surface with very noticeable artefacts, that is, the 3D surface becomes defective and degraded with problem areas easily noticeable. If, on the other hand, the 2D compression/decompression is of good quality, then the 3D surface is reconstructed well and there are no visible differences between the original reconstruction and the reconstruction with the decompressed images.

Fig. 10 (left) depicts the GMPR scanner together with an image captured by the camera (middle) which is then converted into a 3D surface and visualized (right). Note that only the portions of the image that contain patterns (stripes) can be converted into 3D; other parts of the image are ignored by the 3D reconstruction algorithms [21,22]. The original images used in this research are shown in Fig. 11 (Corner, Face1 and Face2). The three images shown in Fig. 11 were compressed by the method described in this paper whose compressed sizes with RMSE and PSNR are shown in Table 3. After decompression, the images were subjected to 3D reconstruction using the GMPR method and compared with 3D reconstruction of the original images. The reconstructed 3D surfaces are shown in Figs. 12–14.

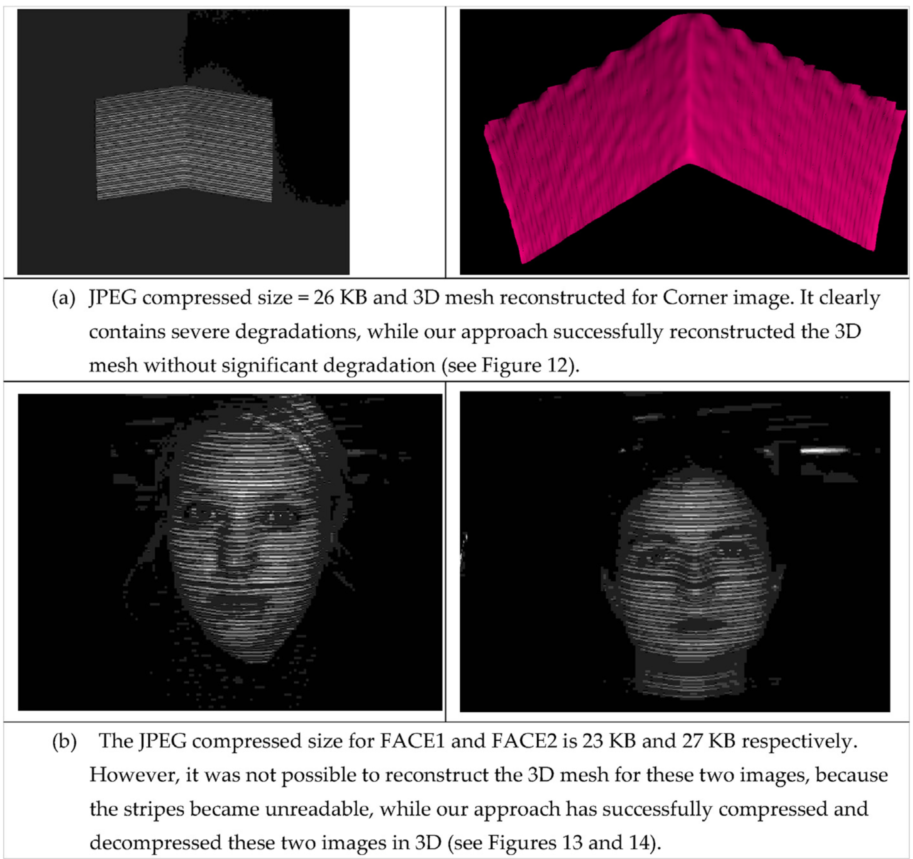
6. Discussion and comparative analysis

Our literature survey did not show results for image compression using the DFT alone. The reason is that by applying a DFT, it yields two sets of coefficients, real and imaginary. If one wishes to keep those for faithful image reconstruction, then it is not possible to achieve high compression ratios. We applied the DFT as described in this paper resulting in images with good visual quality and low compression complexity. A comparative analysis between compression ratios for DFT alone and DFT followed by the Matrix Minimization algorithm show



JPEG Compressed Size=210 KB | JPEG Compressed Size=347 KB | JPEG Compressed Size=279 KB

Fig. 16. Compressed and Decompressed colour images by JPEG, the decompressed images (Boeing and Girl) have lower quality compared with our approach according to RMSE and PSNR. However, our approach couldn't reach to JPEG level of compression for Bagdad's image on the right.



(a) JPEG compressed size = 26 KB and 3D mesh reconstructed for Corner image. It clearly contains severe degradations, while our approach successfully reconstructed the 3D mesh without significant degradation (see Figure 12).

(b) The JPEG compressed size for FACE1 and FACE2 is 23 KB and 27 KB respectively. However, it was not possible to reconstruct the 3D mesh for these two images, because the stripes became unreadable, while our approach has successfully compressed and decompressed these two images in 3D (see Figures 13 and 14).

Fig. 17. 3D reconstruction from JPEG compressed images. In (a) reconstruction was possible but with significant artefacts. In (b) 3D reconstruction was not possible as images were too deteriorated.

Table 6
Comparative analysis between pervious work [12] (Matrix Minimization algorithm) and our approach based on time execution.

Image	Size (MB)	Previous work (Matrix Minimization algorithm)			The proposed algorithm		
		Compressed size (KB)	Bits/Pixel (bpp)	Decompression time (seconds)	Compressed size (KB)	Bits/pixel (bpp)	Decompression time (seconds)
Lena	1.0	120	0.117	102	88	0.085	25
Lion	1.37	98	0.069	240	71	0.05	40
Apples	1.37	92	0.065	90	47	0.033	15
Boeing	6.15	240	0.038	420	182	0.028	150
Girl	4.29	399	0.090	330	315	0.071	114
Bagdad	8.58	673	0.076	720	309	0.035	198
Corner	1.25	56	0.043	84	17	0.013	22
Face1	1.37	46	0.032	144	18	0.012	59
Face2	1.37	38	0.027	174	20	0.014	66

Table 7
Comparative analysis between pervious work [12] (Matrix Minimization) and our approach based on image quality and compression sizes.

Image	Size (MB)	Previous work (Matrix Minimization algorithm)				The proposed algorithm			
		Compressed Size (KB)	Bits/Pixel (bpp)	RMSE	PSNR	Compressed Size (KB)	Bits/Pixel (bpp)	RMSE	PSNR
Lena	1.0	120	0.117	6.8	39.8	88	0.085	3.9	42.2
Lion	1.37	98	0.069	10.1	38.0	71	0.050	8.1	39.0
Apples	1.37	92	0.065	7.1	39.6	47	0.033	4.6	41.5
Boeing	6.15	240	0.038	10.2	38.0	182	0.028	3.9	42.2
Girl	4.29	399	0.090	8.4	38.8	315	0.071	5.5	40.7
Bagdad	8.58	673	0.076	5.9	40.4	309	0.035	5.6	40.6
Corner	1.25	56	0.043	16.0	36.0	17	0.013	15.5	36.2
Face1	1.37	46	0.032	14.4	36.5	18	0.012	11.5	37.5
Face2	1.37	38	0.027	11.2	37.6	20	0.014	9.9	38.1

enormous differences as shown in Table 4.

The results demonstrate that our proposed method of using a DFT in conjunction with the Matrix Minimization algorithm has the ability to compress digital images up to 98% compression ratios. It is shown that the DFT alone cannot compress images with similar ratios and quality. Although it can be seen from Table 4 that our proposed method (DFT + Matrix Minimization algorithm) increases the overall RMSE and, while some image details are lost, reconstructed images are still high quality.

Additionally, the proposed method is compared with JPEG technique [23–25] which is a popular technique used in image and video compression. Also, the JPEG is used in many areas of digital image processing [26]. The main reason for comparing our method with JPEG is because JPEG is based on DCT and Huffman coding. Table 5 shows the analytical comparison between the two methods.

In above Table 5 it shown that our proposed method is better than JPEG technique to compress structured light images, while for conventional images it can be stated that both methods are roughly equivalent as image quality varies in both methods. The following Figs. 15–17 show comparisons between our approach the and JPEG technique for the images shown in Tables 4 and 5

Concerning the compression of structured light images for 3D mesh reconstruction, the comparison of our method with JPEG shows enormous potential for our approach as depicted in Figs. 11–14. Trying to compress the same images using JPEG and then using the decompressed image to generate the 3D mesh clearly shows the problems and limitations of JPEG. This is illustrated in Fig. 17, which shows the JPEG technique on two structured light images for 3D mesh reconstruction.

Comparative analysis focused on our previous work on the Matrix Minimization algorithm based on two discrete transforms DWT and DCT, as suggested by Siddeq and Rodrigues [9–11] performing compression and encryption at the same time. However, complexity of compression and decompression algorithms is cited as a disadvantage of previous work. Table 6 shows the decompression time for the Matrix Minimization algorithm [12] (previous work) compared with our proposed approach. The advantages of the proposed over previous work are summarized as follows:

- The complexity of the decompression steps is reduced in the proposed approach. This is evident from execution times quoted in Table 6 as the current approach runs faster than previous work on the same hardware.
- The header file information of current approach is smaller than previous work leading to increased compression ratios.

It is important to stress the significant novelties of the proposed approach which are the reduced number of steps at decompression stage and smaller header information resulting in faster reconstruction from data compressed at higher compression ratios. Table 7 shows that our proposed image compression method has higher compression ratios and better image quality (i.e. for both types conventional and structured light images) as measured by RMSE and PSNR.

7. Conclusion

This research has demonstrated a novel approach to compress images in greyscale, colour and structured light images used in 3D reconstruction. The method is based on the DFT and the Matrix-Minimization algorithm. The most important aspects of the method and their role in providing high quality image with high compression ratios are highlighted as follows.

- After dividing an image into non-overlapping blocks (4×4), a DFT is applied to each block followed by quantizing each part (real and imaginary) independently. Meanwhile, the DC value (Low Frequency Coefficients) from each block are stored in a new matrix, while the rest of the values in the block are the High Frequency Coefficients.
- The Matrix-Minimization algorithm is applied to reduce the high-frequency matrix to 1/3 of its original size, leading to increased compression ratios.
- The relatively large probability table of previous method was reduced to two values, minimum and maximum leading to higher compression ratios and faster reconstruction.

Results demonstrate that our approach yields better image quality at higher compression ratios while being capable of accurate 3D reconstruction of structured light images at very high compression ratios. Overall, the algorithm yields a best performance on colour images and structured light images used in 3D reconstruction than on standard grey images.

On the other hand, the compression steps introduced by the MM algorithm, especially at decompression stage, make the compression algorithm more complex than, for instance, standard JPEG. In general, it can be stated that decompression is slower than compression due to the search space to recover the original Low and High Frequency coefficients. In addition, arithmetic coding and decoding is applied to three sets of data (DC values, in addition to real and imaginary frequency coefficients) adding significantly more computation steps leading to increased execution time.

Conflicts of interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

CRediT authorship contribution statement

Mohammed H. Rasheed: Conceptualization, Methodology. **Omar M. Salih:** Data curation, Writing - original draft. **Mohammed M. Siddeq:** Visualization, Software. **Marcos A. Rodrigues:** Supervision, Writing - review & editing.

Acknowledgments

We grateful acknowledge the Computing, Communication and

Cultural Research Institute (C3RI) and the Research and Innovation Office at Sheffield Hallam University for their support.

References

[1] Richardson IEG. Video codec design. John Wiley &Sons; 2002.
[2] Sayood K. Introduction to data compression. 2nd ed. Academic Press, Morgan Kaufman Publishers; 2001.
[3] Rao KR, Yip P. *Discrete cosine transform*: algorithms, advantages, applications. San Diego, CA: Academic Press; 1990.
[4] Gonzalez Rafael C, Woods RichardE. Digital image processing. Addison Wesley publishing company; 2001.
[5] Yuan Shuyun, Hu Jianbo. Research on image compression technology based on Huffman coding. J Vis Commun Image Represent February 2019;59:33–8.
[6] Li Peiya, Lo Kwok-Tung. Joint image encryption and compression schemes based on 16×16 DCT. J Vis Commun Image Represent January 2019;58:12–24.
[7] M. Rodrigues, A. Robinson and A. Osman. Efficient 3D data compression through parameterization of free-form surface patches, In: Signal process and multimedia applications (SIGMAP), proceedings of the 2010 international conference on. IEEE, 130-135.
[8] Siddeq MM, Al-Khafaji G. Applied minimize-matrix-size algorithm on the transformed images by DCT and DWT used for image compression. Int J Comput Appl 2013;70:15.
[9] Siddeq MM, Rodrigues MA. A new 2D image compression technique for 3D surface reconstruction. In: 18th international conference on circuits, systems, communications and computers. Greece: Santorin Island; 2014. p. 379–86.
[10] Siddeq MM, Rodrigues MA. A novel image compression algorithm for high resolution 3D reconstruction. 3D Research 2014;5(2). https://doi.org/10.1007/s13319-014-0007-6. Springer.
[11] M.M. Siddeq and Rodrigues Marcos. Applied sequential-search algorithm for compression-encryption of high-resolution structured light 3D data. In: Blashki, Katherine and Xiao, Yingcai, (eds.)MCCSIS : multi conference on computer science and information systems 2015. IADIS Press, 195-202.
[12] Siddeq MM, Rodrigues Marcos. A novel 2D image compression algorithm based on two levels DWT and DCT transforms with enhanced minimize-matrix-size algorithm for high resolution structured light 3D surface reconstruction. 3D Research 2015; 6(3):26. https://doi.org/10.1007/s13319-015-0055-6.
[13] Siddeq Mohammed, Rodrigues Marcos. A novel high frequency encoding algorithm for image compression. EURASIP J Appl Signal Process 2017;26. https://doi.org/10.1186/s13634-017-0461-4.
[14] Siddeq Mohammed, Rodrigues Marcos. DCT and DST based image compression for 3D reconstruction. 3D Research 2017;8(5):1–19.
[15] Sheffield Hallam University, Mohammed M Siddeq and Marcos A Rodrigues. Image data compression and decompression using minimize size matrix algorithm. WO 2016/135510 A1. Patent 2016.
[16] M.M. Siddeq and Rodrigues Marcos. Novel 3D compression methods for geometry, connectivity and texture. 3D Research, 7 (13). 2016
[17] Siddeq MM, Rodrigues Marcos. 3D point cloud data and triangle Face compression by a novel geometry minimization algorithm and comparison with other 3D formats. In: Proceedings of the international conference on computational methods. vol. 3. California USA: University of California; 2016. p. 379–94.
[18] Siddeq MM, Rodrigues AM. A novel hexa data encoding method for 2D image crypto-compression. Multimed Tool Appl 2019. https://doi.org/10.1007/s11042-019-08405-3. Springer.
[19] Rodrigues M, Kormann M, Schuhler C, Tomek P. Robot trajectory planning using OLP and structured light 3D machine vision. Heidelberg: Springer; 2013. p. 244–53. Lecture notes in Computer Science Part II. LCNS, 8034 (8034).
[20] Rodrigues M, Kormann M, Schuhler C, Tomek P. Structured light techniques for 3D surface reconstruction in robotic tasks. In: Kacprzyk J, editor. Advances in intelligent systems and computing. Heidelberg: Springer; 2013. p. 805–14.
[21] Rodrigues M, Kormann M, Schuhler C, Tomek P. An intelligent real time 3D vision system for robotic welding tasks. Mechatronics and its applications. IEEE Xplore; 2013. p. 1–6.
[22] Wang, Zhou; Bovik, A.C.; Sheikh, H.R.;Simoncelli, E.P. *Image quality assessment*: 2004 from error visibility to structural similarity". IEEE Trans Image Process. 13(4): 600–612.
[23] Adler A, Boublil D, Zibulevsky M. Block-based compressed sensing of images via deep learning. In: 2017 IEEE 19th international workshop on multimedia signal processing. Luton: MMSP; 2017. p. 1–6.
[24] BAl-Ani MuzhirShaban, Hammouri Talal Ali. Video compression algorithm based on frame difference approaches. Int J Soft Comput November 2011;2(No.4).
[25] Adler A. Covariance-assisted matching pursuit. IEEE Signal Process Lett Jan. 2016; 23(1):149–53.
[26] Dar Y, Elad M, Bruckstein AM. Optimized pre-compensating compression. IEEE Trans Image Process Oct. 2018;27(10):4798–809.