Image Compression for Quality 3D Reconstruction

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

Citation:

RODRIGUES, Marcos, SALIH, Omar M, RASHEED, Mohammed H and SIDDEQ, Mohammed M (2020). Image Compression for Quality 3D Reconstruction. Journal of King Saud University (Computer and Information Sciences). [Article]

Image Compression for Quality 3D Reconstruction

Omar M. Salih[1], Mohammed H. Rasheed[1], and Mohammed M. Siddeq[1], Marcos A. Rodrigues[2]
omar.alsabaawi@ntu.edu.iq , mohammed.rasheed@ntu.edu.iq , mamadmmx76@gmail.com
[1]*Computer Engineering Dept., Technical College/Kirkuk,*
Northern Technical University, IRAQ
[2]*GMPR-Geometric Modelling and Pattern Recognition Research Group,*
Sheffield Hallam University, Sheffield, UK

Abstract
A 3D mesh can be reconstructed from multiple viewpoint images or from a single structured light image. Lossy compression of such images by standard techniques such as JPEG at high compression ratios lead to 3D reconstruction being adversely affected by artifacts and missing vertices. In this paper we demonstrate an improved algorithm capable of high compression ratios without adversely affecting 3D reconstruction and with minimum data loss. The compression algorithm starts by applying block DCT over the input image, and the transformed data being quantized using an optimized quantization matrix. The quantized coefficients of each block are arranged as a 1D array and saved with other block's data in a larger matrix of coefficients. The DC coefficients are subject to a first order difference whose values are referred to as residual array. The AC coefficients are reduced by eliminating zeros and saving the non-zero values in a reduced coefficients array using a mask of 0 (for a block of zeros) and 1 (for a block of non-zeros). Finally, arithmetic coding is applied to both coefficients and residual arrays. At decompression stage, the coefficients matrix is regenerated by scanning the coefficients array and examining the headers to substitute zero and non-zero data. This matrix is then added to the residual array to obtain the original DC values. The IDCT is then applied to obtain the original image. The proposed algorithm has been tested with images of varying sizes in the context of 3D reconstruction. Results demonstrate that our proposed algorithm is superior to traditional JPEG at higher compression ratios with high perceptual quality of images and the ability to reconstruct the 3D models more effectively, both for structured light images and for sequences of multiple viewpoint images.

Keywords: 3D Mesh Reconstruction, DCT, JPEG, Image Compression, structured light images

1. Introduction

In recent years, the demand for geometry modelling has grown for a wide range of applications such as aerospace models, medicine, computer science, automotive CAD datasets, engineering, virtual reality, entertainment, and so on. Modelling 3D objects can be classified into two primary categories namely surface modelling and solid modelling. Surface modelling is the branch in which we can represent and deal with 2D surfaces embedded in the 3D space. A popular polynomial discrete representation method for 3D surfaces is mesh generation from scanned images. The popularity comes from their algorithmic simplicity, ease of calculation on Graphics Processing Unit (GPU), and rendering and displaying efficiency [1]. A 3D mesh consists of three basic elements: edges, vertices, and faces. The edges basically are lines connecting vertices. Faces are closed surfaces created by edges. Vertices are formed by connecting neighboring edges. In regular meshes, all faces have the same size, and also all vertices have the same valence. Among many representation methods, triangular meshes may be considered as the most common method due to their simple and efficient modeling [2]. Current 3D applications devices have high technology graphics cards for rendering 3D meshes with the aid of special software for editing and visualizing data. Most of these applications need to save as well as transmit, share and exchange 3D data over networked environments and the Internet. On the other hand, the large size of 3D mesh data requires a higher cost storage size and imposes hard limits on bandwidth due to a large amount of data [3]. This calls for active research on more efficient 3D data compression and reconstruction algorithms to satisfy the demands of 3D modeling. In the literature (e.g. [3], [4], [5], [6] and [7]), many different techniques have been proposed for 3D mesh data compression and reconstruction based on high degree polynomial interpolation or partial differential equations. An approach confines the concept that the actual geometry of a 3D solid object is the 2D surface. In other words, it means compressing the 2D surface image to a high compression ratio with minimum loss without adversely altering 3D reconstruction. In this way, we can decrease 3D computational complexity, sharing time, bandwidth limit and storage requirements [8].

Image compression is a process of reducing the amount of data and can be lossy or lossless. Lossless compression ensures a complete retrieval of the original data, while in lossy compression some of the information from the original image is lost. The key feature of lossy compression is getting higher compression ratios at the expense of losing some information. Just because some information is lost it does not mean the output is necessarily distorted. For this reason, lossy compression can cut image file sizes in half as compared to lossless compression without a perceptible deterioration in quality [9]. Neighboring pixels in the spatial domain are highly correlated, which means they include redundant information causing the energy to distribute evenly across the image, therefore, making it difficult to compress. These lead to a good reason for transforming spatial domain images into a different representation in the frequency domain. The frequency domain can decorrelate data and compact the energy in the image by concentrating it into a small number of significant values and the data becomes easier to compress [10]. The two most extensively used transforms for image compression are discrete cosine transform (DCT) and discrete wavelet transform (DWT) which both are used in JPEG and JPEG-2000 standards, respectively. The DCT is usually applied to small, regular blocks of images (e.g. 8x8 blocks) while the DWT is usually applied to larger image segments or to complete images [7].

The focus of this paper is on compression of 3D data reconstructed from multiple viewpoint images or from a single structured light image. Because 3D structures are obtained from images, we naturally focus on quality image compression. Structured light images contain patterns of light and shadows on the surface of the 3D object and captured by the sensor at very high resolutions allowing 3D reconstruction. Related to the proposed algorithm in this paper, previous works on compressing 2D image data appropriate for 3D reconstruction from structured light images are limited by Siddeq and Rodrigues and first proposed in 2014 [11] and [12]. They applied a single level DWT on the input image, and then a DCT on the LL sub-band producing the DC and AC components. A single level DWT is applied again to the DC component, and then the resulted LL2 sub-band is transformed again by DCT. An algorithm named "matrix minimization algorithm" was applied to the AC-matrix and other sub-bands achieving compression ratios of up to 98%. In the same year [8], they developed a technique at the decompression stage by proposing an algorithm named "Limited-Data Sequential Search Algorithm" where similar transformations are applied to the data blocks followed by arithmetic coding at compression stage. The results show a better image quality at higher compression ratios than both JPEG and JPEG2000. However, the overall complexity of those algorithms increased computation time at both compression and decompression stages. In [13], Siddeq and Rodrigues proposed a two-level DWT followed by a DCT to generate the DC and AC components. A second DCT was applied to the AC components generating two arrays: zeros and non-zeros arrays. A minimize-matrix size (MMS) algorithm then applied to the AC-coefficients and the other high frequencies followed by arithmetic coding. A novel fast match search decompression algorithm (FMS) was proposed to reconstruct all high-frequency components by calculating all compressed data probabilities using a binary search algorithm to estimate the data with the aid of a lookup table. The results showed higher compression ratios than JPEG and JPEG2000 with better visual image quality and accurate 3D reconstructing. On the other hand, the complexity of the FMS-based binary search algorithm lead to a remarkable increase in the execution time compared with previous works. In 2017 [7] they introduced a new method for compression based on two transforms: 1D-DCT applied to each row and 1D-DST (Discrete Sine Transform) applied to each column of the input image followed by quantization for the high frequencies and then coded by matrix minimization algorithm followed by arithmetic coding. A binary search based on fast matching search algorithm was used at decompression yielding up to 99% compression ratio.

It is clear from previous research that effective image compression is based on a combination of multi transformations of DWT, DCT, and DST in conjunction with a number of algorithms such as matrix minimization with sequential and binary data search. Such algorithms are used to increase compression ratios with accurate surface reconstruction, however, they increase the overall complexity which in turn increases the computation time accordingly.

The contribution of this paper is on developing new methods of 3D surface compression from multiple viewpoint images and from structured light images. We introduce a new method for lossy image compression based on JPEG alone with optimization of quantization leading to the creation of two components: the DC-components and AC-coefficients that represent low and high frequencies, respectively. The DC and AC coefficients are gathered into one matrix called coefficients matrix which is subject to a number of operations for increasing high frequencies and eliminating zeros to attain high compression ratios and a precise 3D reconstruction. We rigorously tested the algorithms whose results are compared with the traditional JPEG algorithm in terms of 2D and 3D RMSE and PSNR.

2. The JPEG Technique

JPEG is a lossy/lossless compression algorithm standard of the ISO/CCITT committee known as JPEG (Joint Photographic Experts Group) for the compression of both grayscale and colour images. It was first established in 1987 and produced its first draft in 1991 and became widely used especially in web pages. JPEG has an important feature allowing the user to adjust the compression ratio via quality parameter [14, 15]. The problem with JPEG is the degradation of image quality when low-quality parameters are selected to get a higher compression ratio [16]. The JPEG algorithm is illustrated in Figure 1 and demonstrated through the following steps:

1- The image is divided into non-overlapping 8×8 blocks called data units, each data unit is compressed separately. If the image dimensions are not a multiple of eight, the last row and column are padded with zeros [14, 17].
2- Each 8×8 block is transformed by DCT to produce new 8×8 data blocks in the frequency domain. The frequency components of each data block consist of a single DC coefficient at the top-left corner that represents the low frequency which is the average value of the 8×8 data block, and the remaining 63 elements called AC coefficients which represent the higher frequency. The key feature of DCT is de-correlating image data yielding negligible data at the right bottom of the 8×8 block, this process increases the compression ratio while maintaining important data approximately similar to the original data as the human eye cannot identify those differences [18, 19].
3- Each of the 64 coefficients in the 8×8 data block are quantized by the quantization coefficient (QC), this is accomplished by dividing all the 64 coefficients by a separate number and rounding to the nearest integer. This causes irretrievable information loss. Larger QCs numbers cause more lost information [20].
4- The quantized 64 coefficients for every block are zigzag scanned to convert the matrix into a one-dimensional array of 64 elements. This array is encoded using Run Length Encoding (RLE) and Huffman coding [21].

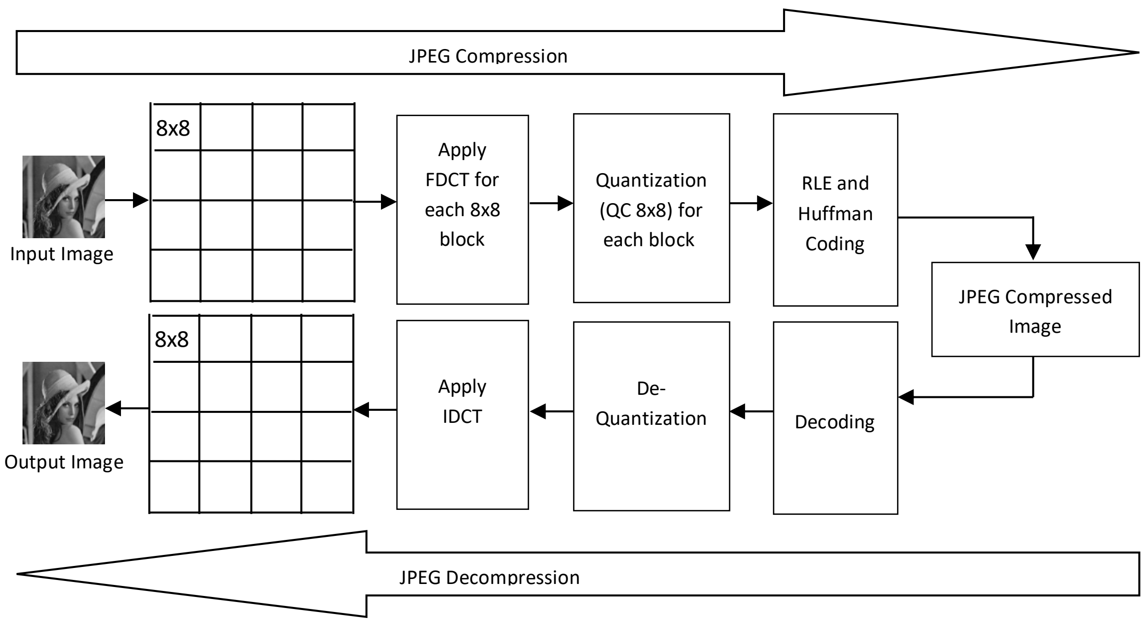


Figure 1: JPEG Compression and Decompression

3. The New Proposed Algorithm

The proposed compression and decompression stages for 3D mesh reconstruction are described in the following sections.

3.1 Compression

First, the image is subdivided into non-overlapping m×n blocks. For a direct comparison with JPEG we use the standard block size of 8×8 pixels starting from the upper left corner of an image. The image may be padded with zeros if the image dimensions are not a multiple of eight. The DCT is applied to each 8×8 block and saved as an array of 64 elements, the first element represents a low frequency called the DC component while the remaining 63 elements are called the AC coefficients which represent the high frequencies. Figure 2 shows the block diagram of the proposed compression algorithm. The DC coefficient and its neighboring AC coefficients are kept un-quantized by substituting 1s in their equivalent locations in the quantization matrix. This optimization can improve the reconstruction and preserves image quality for high compression ratios.

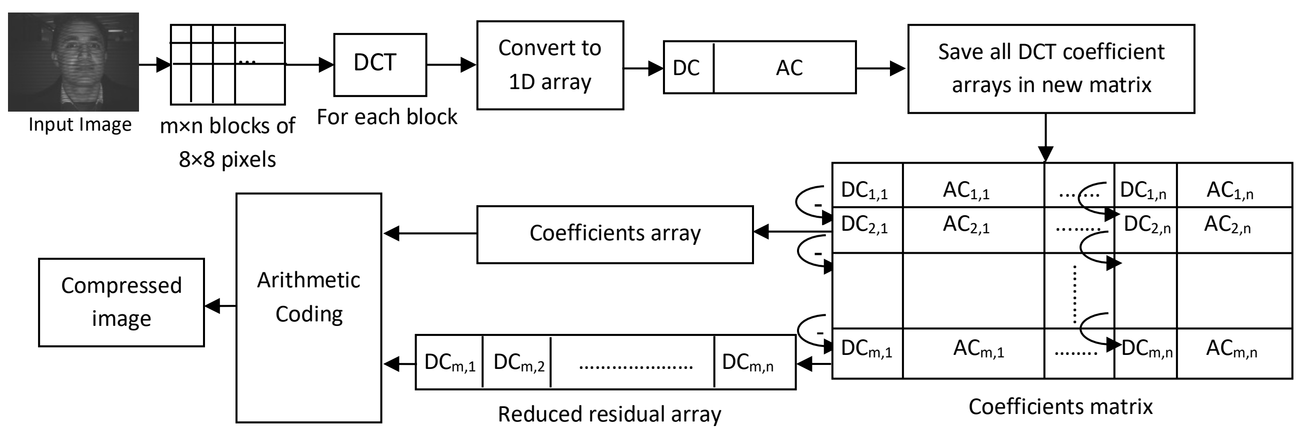


Figure 2: Proposed Compression Algorithm

The quantization coefficients for an 8×8 block used after the DCT step are shown in the following matrix:

$$Q = \begin{bmatrix} 1 & 1 & 1 & 1 & 24 & 40 & 51 & 61 \\ 1 & 1 & 1 & 19 & 26 & 58 & 60 & 55 \\ 1 & 1 & 16 & 24 & 40 & 57 & 69 & 56 \\ 1 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}$$

[need to explain here where those numbers come from, why this particular choice]

All quantized arrays are saved in a new matrix named coefficients matrix corresponding to the block indices of the original image. In every column, each DC coefficient is replaced by the difference between the current row DC value with the next row's value starting from the first row (i=1) down to (i=m-1), that is:

$$DC_{i,j} = DC_{i,j} - DC_{i+1,j} \tag{1}$$

Where i,j are the indices of the block's row and column (i=1:m, j=1:n), respectively. This makes the data more correlated and small as possible to get a high compression ratio. The last DC coefficients in each column are kept unchanged and moved into a new array called "reduced residual array" while substituting their original positions in the coefficients matrix by zeros. The coefficients matrix then transferred to a new array "coefficients array", this is done by scanning every block (64 values) in each row. If all the coefficients are zeros they will be substituted by a single zero in coefficients array, otherwise, substituting the same values in coefficients array led by "1" as a header if any value is not equal to zero. Figure 3 shows the coefficients array format. Then, both of the coefficients array and the reduced residual array are arithmetically codded yielding the compressed image file.

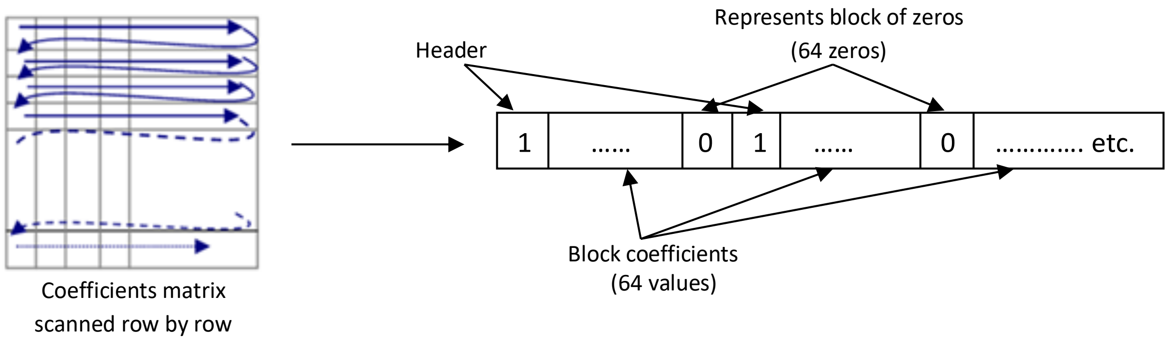


Figure 3: Coefficients array format

3.2 Decompression

The decompression steps are literally the reversing of the compression algorithm, Figure 4 illustrates the steps. First, the compressed image file is arithmetically decoded to obtain the coefficients array and the reduced residual array. The coefficients matrix is re-generated from the coefficients array by scanning and checking the headers (1s) and the zeros in the coefficients array. If a header of "1" is present, the next 64 values are moved to the coefficient matrix in the corresponding 8×8 block index, while if a zero is present, then 64 zeros are substituted.

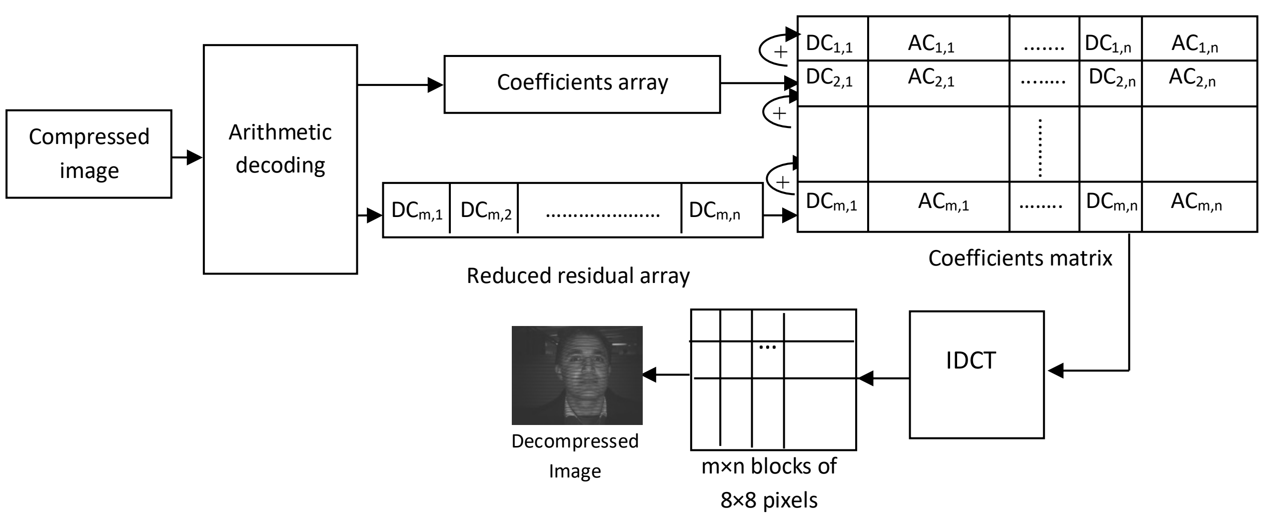


Figure 4: Developed JPEG Decompression

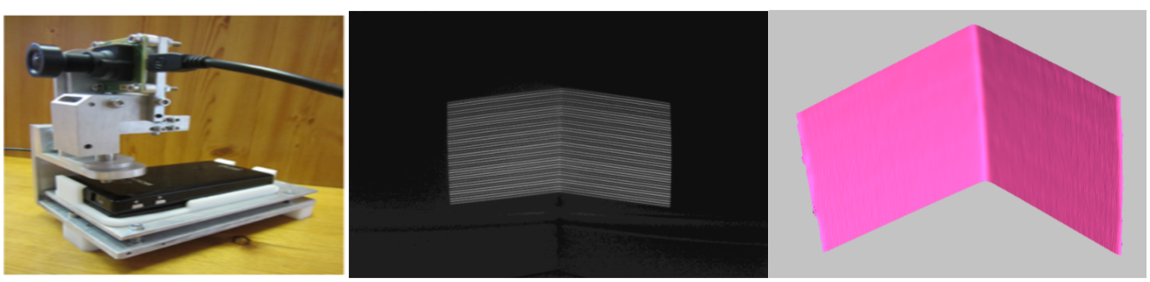
The current DC values in the coefficients matrix are actually the differences between every two consecutive values within the same column. The original DC coefficients are retrieved by: first, substituting the reduced residual array in the last row DC coefficients, and second, the remaining DC values in every column are replaced by the addition between the current row DC value with the next row's value starting from the row (i=m-1) up to the first (i=1), that is:

$$DC_{i,j} = DC_{i,j} + DC_{i+1,\,j} \tag{2}$$

Where i,j are the indices of the block's row and column (i=1:m, j=1:n), respectively. Finally, the IDCT (Inverse Discrete Cosine Transform) is applied to the coefficients matrix to recover the original image.

4. Experimental Results

The proposed algorithm is applied to grayscale structured light images. These images are input to a software developed by GMPR (Geometric Modeling and Pattern Recognition) Research Group at Sheffield Hallam University [22, 23] to create 3D mesh surfaces. The basic idea of the GMPR 3D scanner is to capture 2D images by projecting horizontal patterns of light on the surface of an object. The relationship between the capturing sensor and the lighting source determines the 3D surface positions along the stripe pattern. A 3D reconstruction algorithm then converts a 2D image into a 3D surface in a few milliseconds [24] as illustrated in Figure 5.



(a) The 3D mesh Scanner (b) A 2D image captured by the scanner (c) converting into a 3D surface

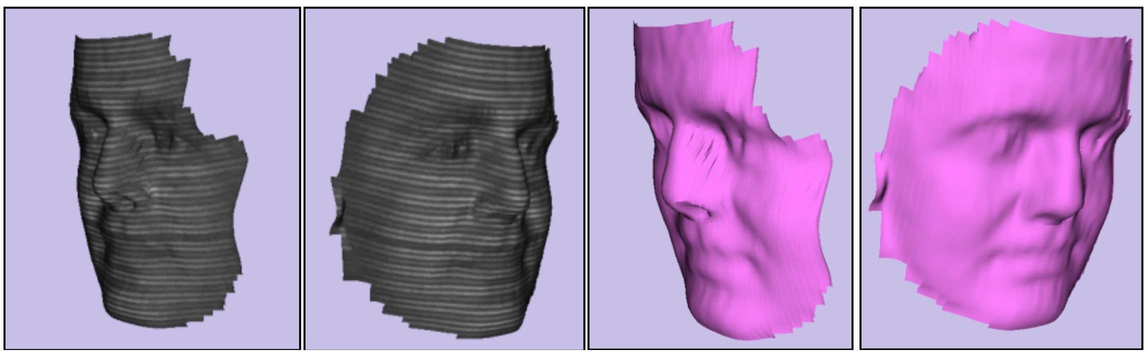
Figure 5: Converting a 2D image into a 3D surface using the GMPR technique

Table 1 shows the experimental results of three grayscale images compressed by the proposed algorithm as shown in Figures 6, 7, and 8.

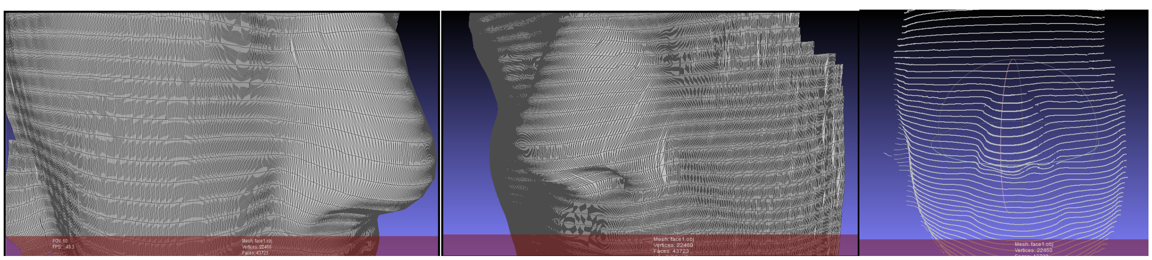
Table 1: Compressed 2D images using the proposed method

Image name	Original Size	Image dimension	Compressed Size	RMSE	PSNR
Face1	1.38 Mbytes	1932 x 1040	14 KB	3.3	42.9
Face2	1.38 Mbytes	1932 x 1040	17.9 KB	4.35	41.7
Face3	1.38 Mbytes	1932 x 1040	21.1 KB	8.3	38.9

(a) Original image "Face1" (b) Decompressed image "Face1" by our proposed algorithm



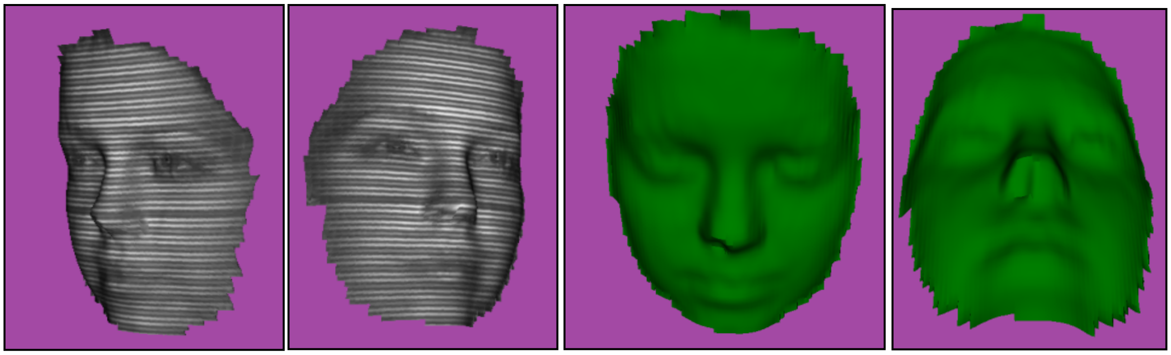
(c) The 3D object for the decompressed image "Face1" created by GMPR software



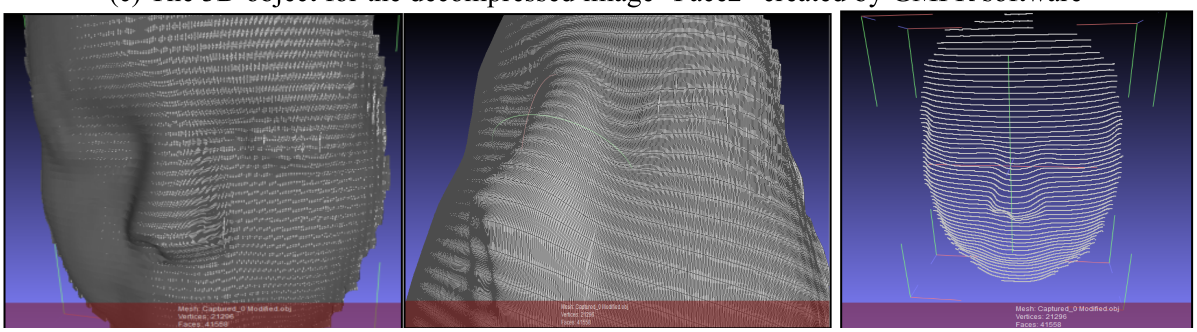
(d) Meshlab display of 3D mesh surface details created successfully for "Face1"

Figure 6: (a, b, c, and d) decompressed “Face1” image converted successfully to a 3D surface at higher compression ratio of 99%

(a) Original image "Face2" (b) Decompressed image "Face2" by our proposed algorithm



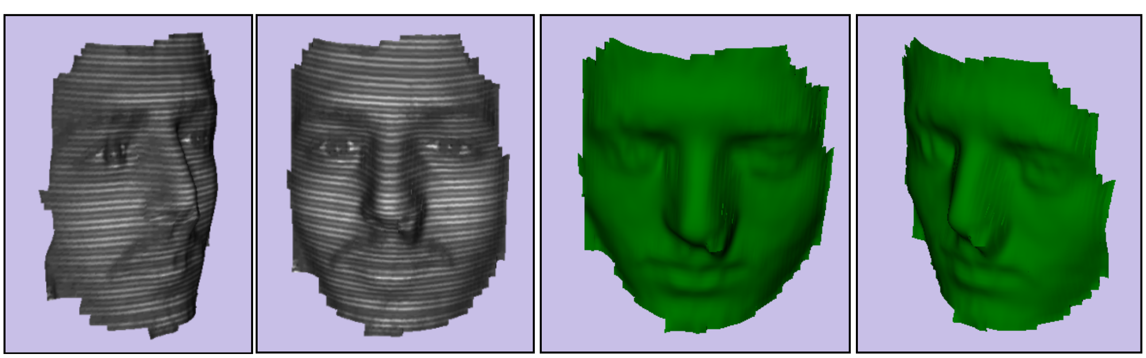
(c) The 3D object for the decompressed image "Face2" created by GMPR software



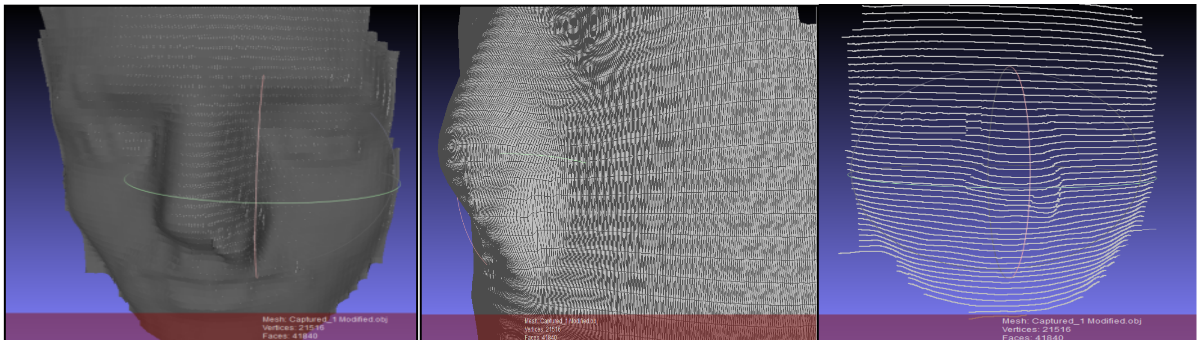
(d) Meshlab display of 3D mesh surface details created successfully for "Face2".

Figure 7: (a, b, c, and d) decompressed "Face2" image converted successfully to a 3D surface at higher compression ratio over 98%

(a) Original image "Face3"   (b) Decompressed image "Face3" by our proposed algorithm



(c) The 3D object for the decompressed image "Face3" created by GMPR software



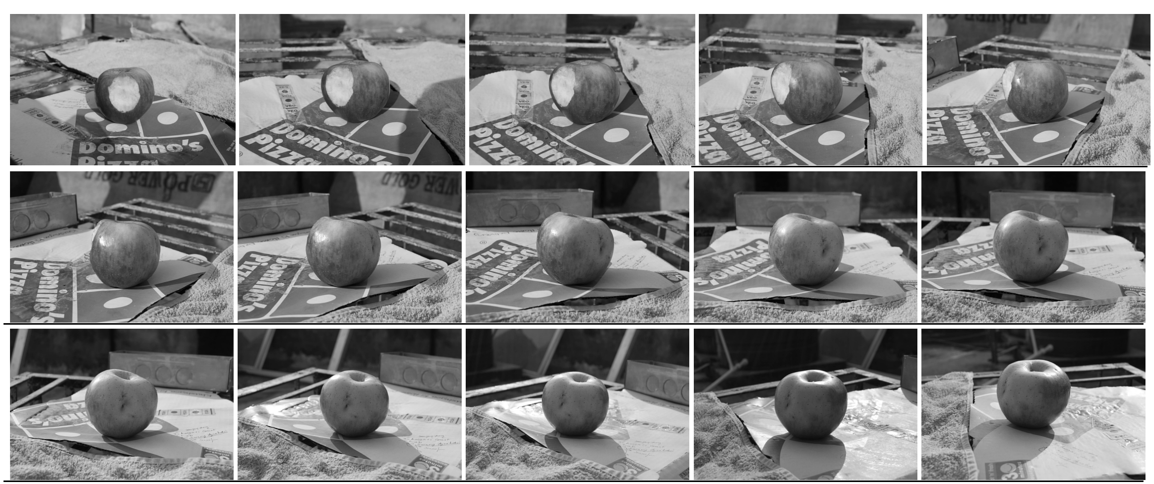
(d) Meshlab display of 3D mesh surface details created successfully for "Face3"

Figure 8: (a, b, c, and d) decompressed "Face3" image converted successfully to a 3D surface at higher compression ratio over 98%

Further to structured light images, the proposed algorithm is applied to three different sets of multiple viewpoint images to create 3D objects. The object is created by special software called 3DF Zephyr Software [25] and converted into a 3D Mesh by using 3D MeshMixer software from Autodesk [26]. Table 2 shows the experimental results of these 3 sets of images compressed by our proposed algorithm as shown in Figures 9, 10, and 11.

Table 2. Compressed 2D multiple viewpoint images by the proposed method at high compression ratios

Sequence	Number of Frames	Total frame size	Total compressed size	Average RMSE	Average PSNR
Apple	48	114.8 MB	3.1 MB	5.8	41.3
Soft Toy	42	100.38 MB	2.39 MB	11.0	38.2
Face	20	152 MB	2.25 MB	8.1	39.6



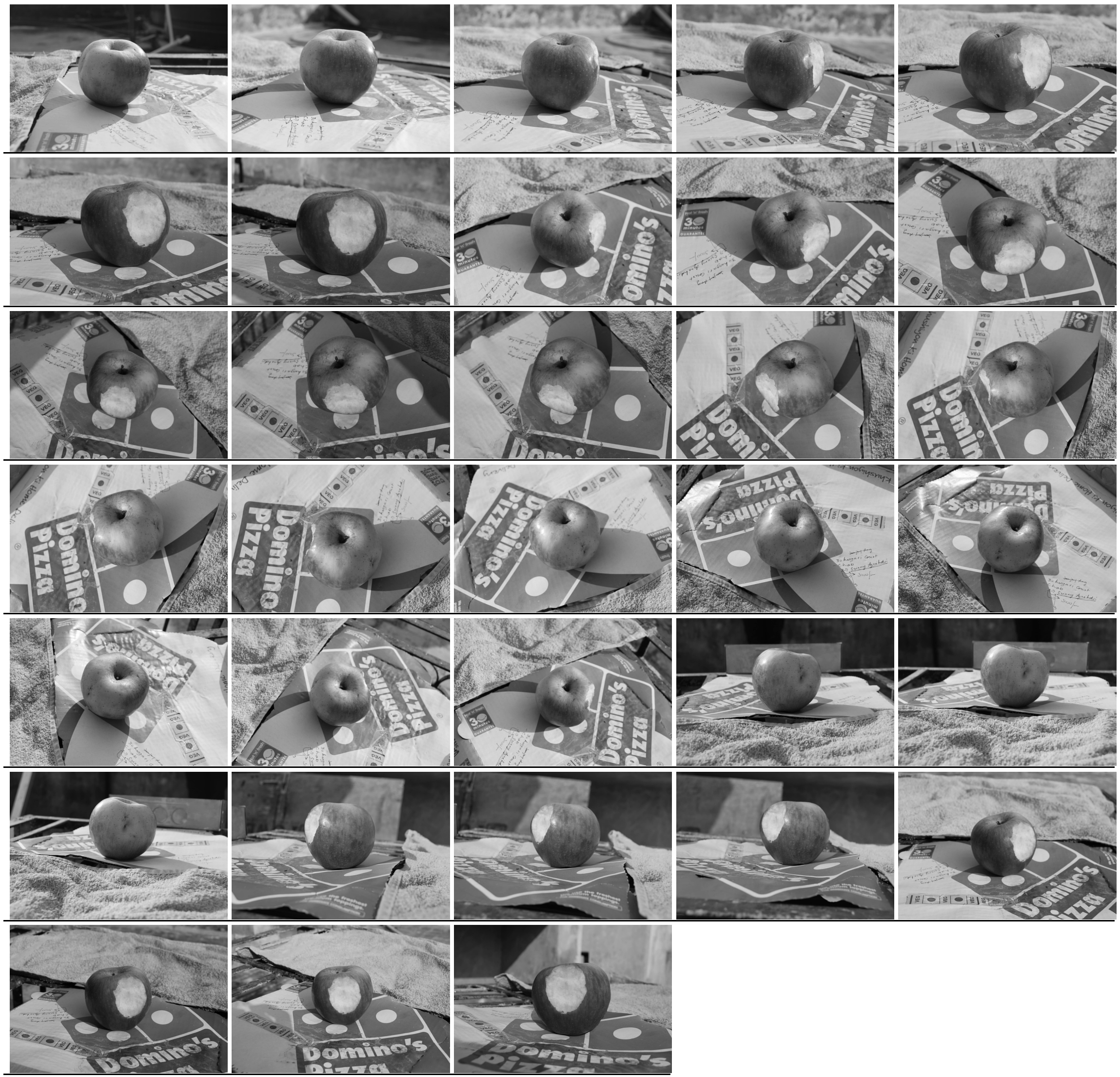
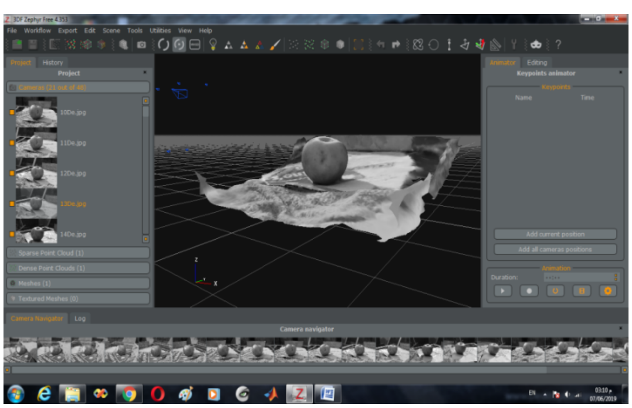
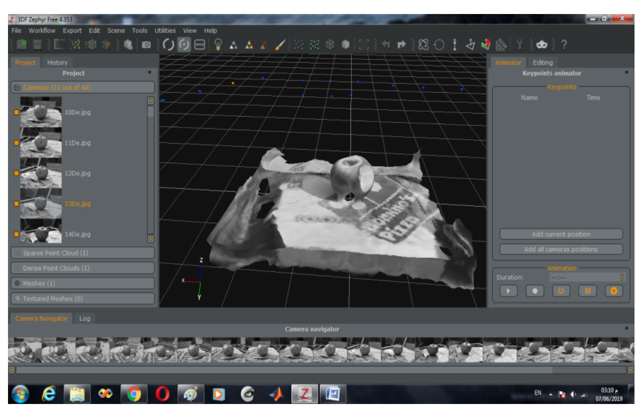
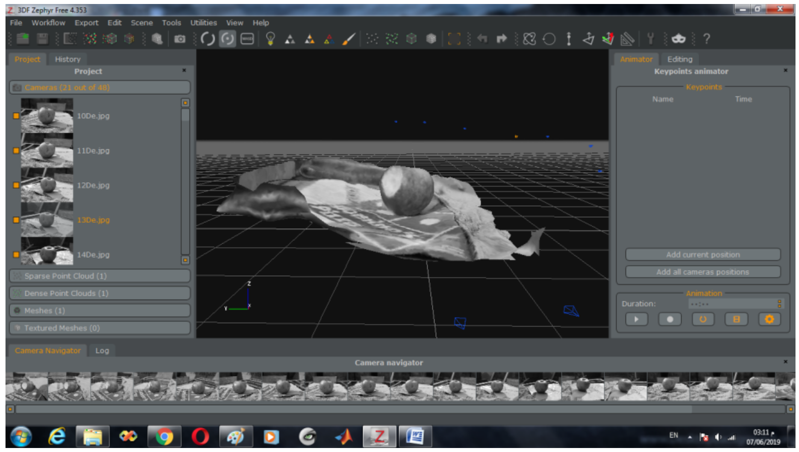


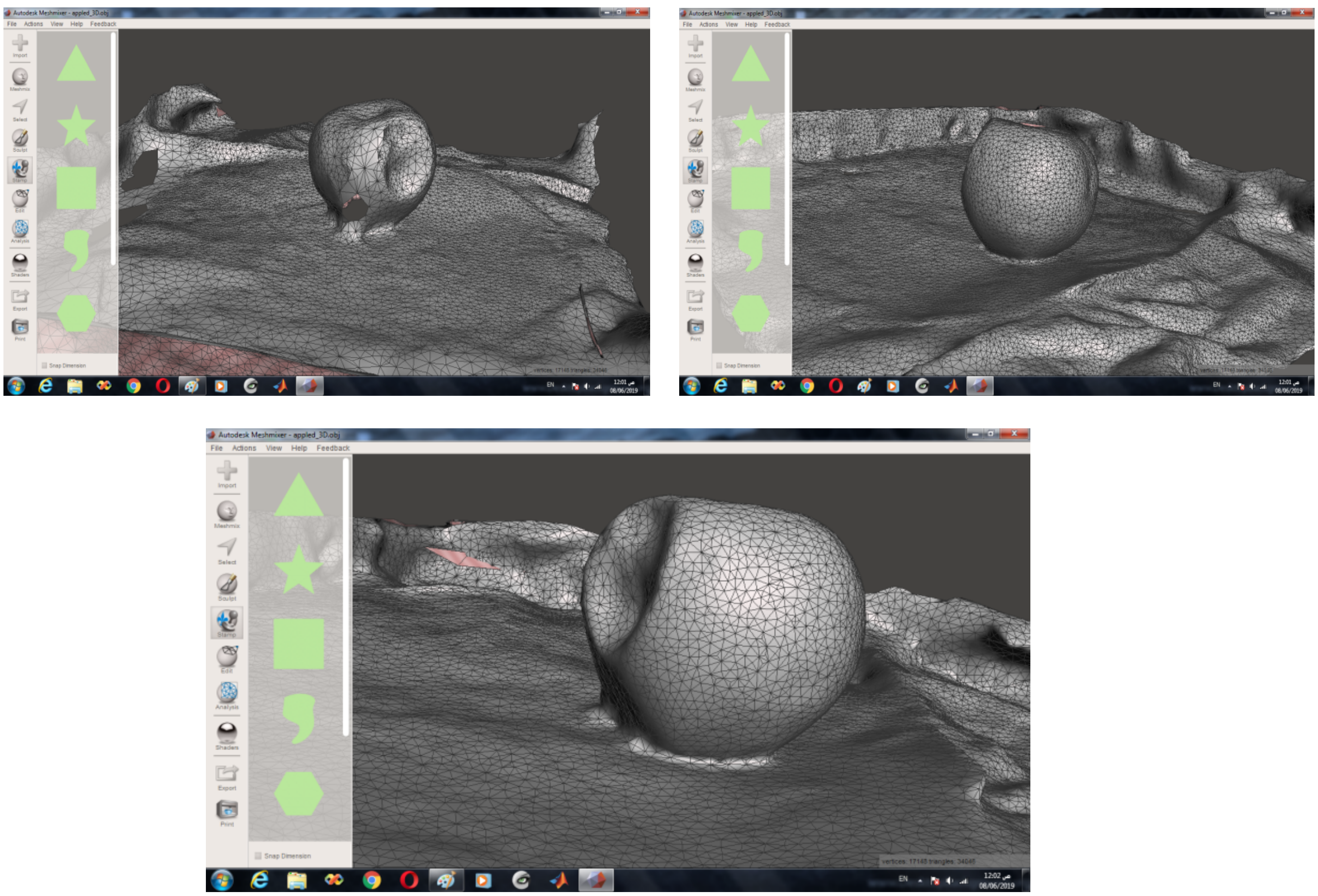
Figure 9: Decompressed 48 frames of Apple images, dimensions for each image =1936 x 1296







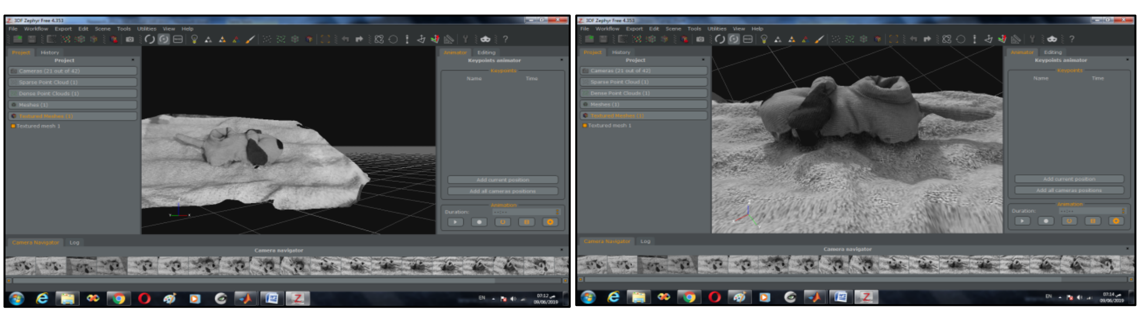
(a) Apple frames converted to 3D object by using 3DF Zephyr Software

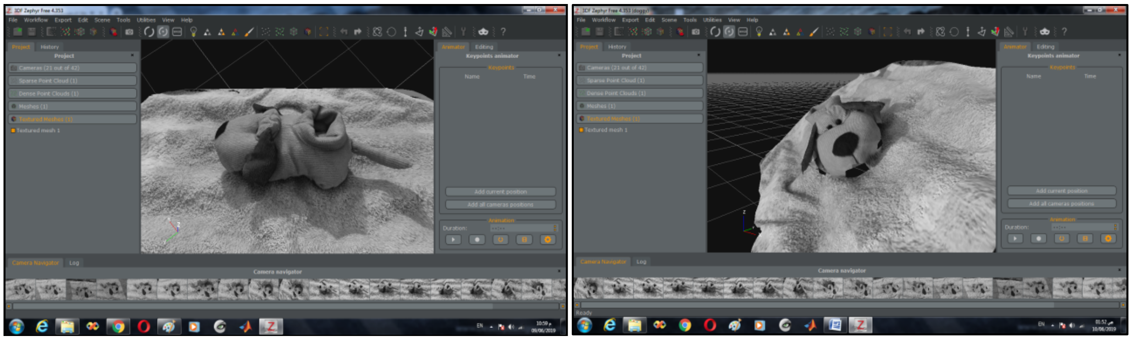


(b) 3D Mesh for the 3D apple shown by 3D MeshMixer software

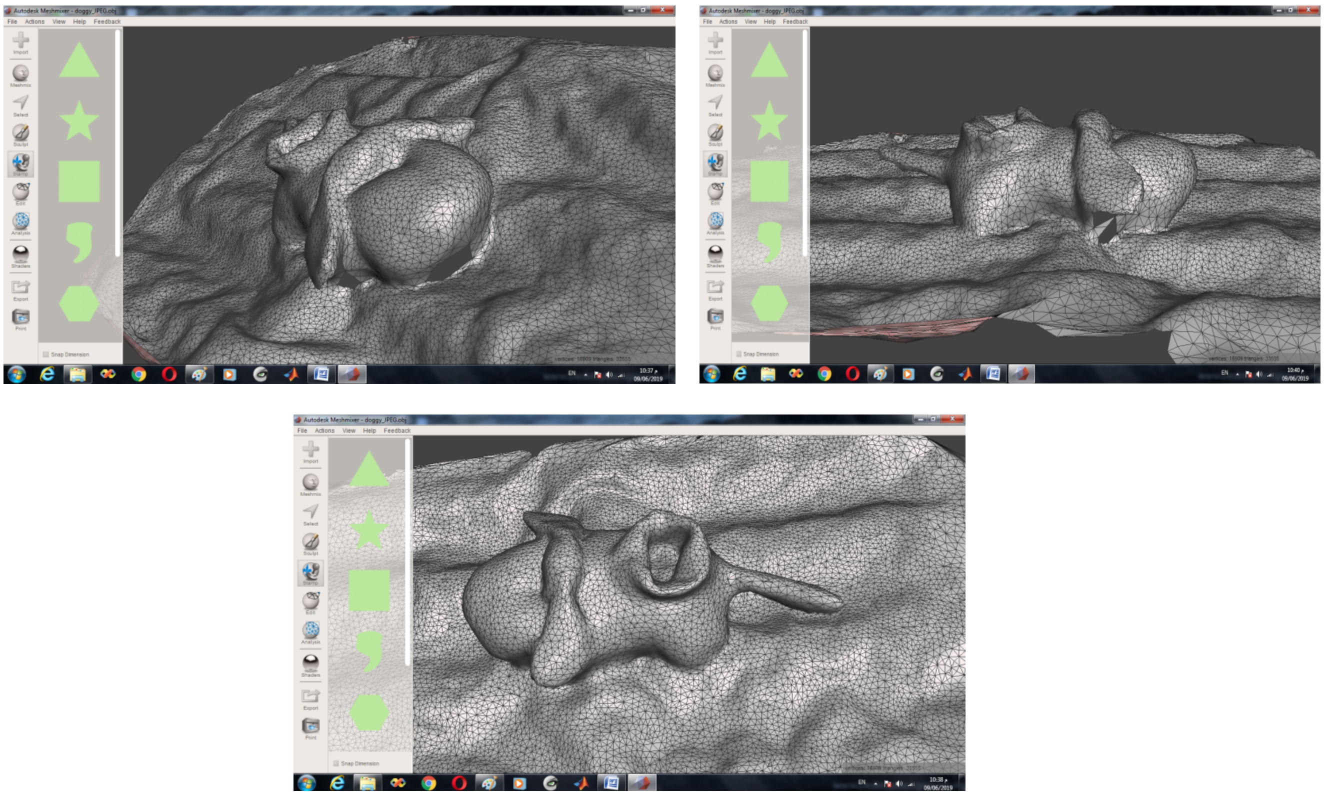
Figure 10: (a) and (b) show the 3D object and mesh surface for the decompressed Apple images by our proposed image compression at higher compression ratio up to 97.2%

Figure 11: Decompressed 42 frames of Soft Toy images, dimensions for each image = 1936 x 1296.





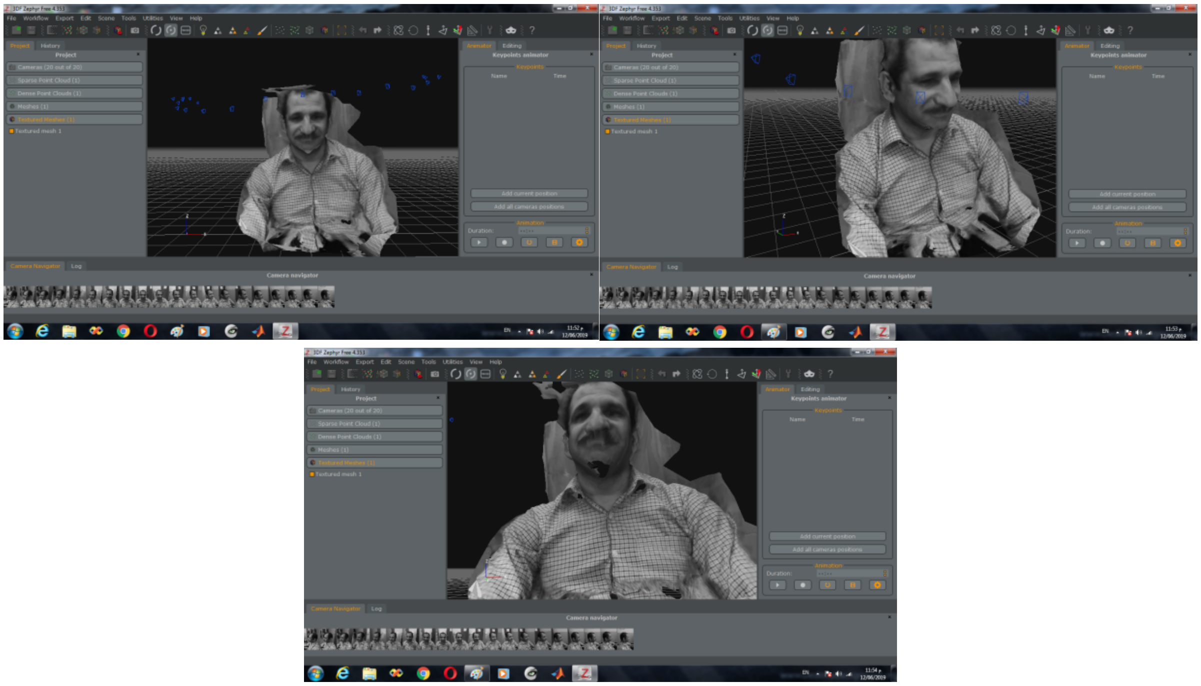
(a) Soft Toy frames converted to 3D object by using 3DF Zephyr Software



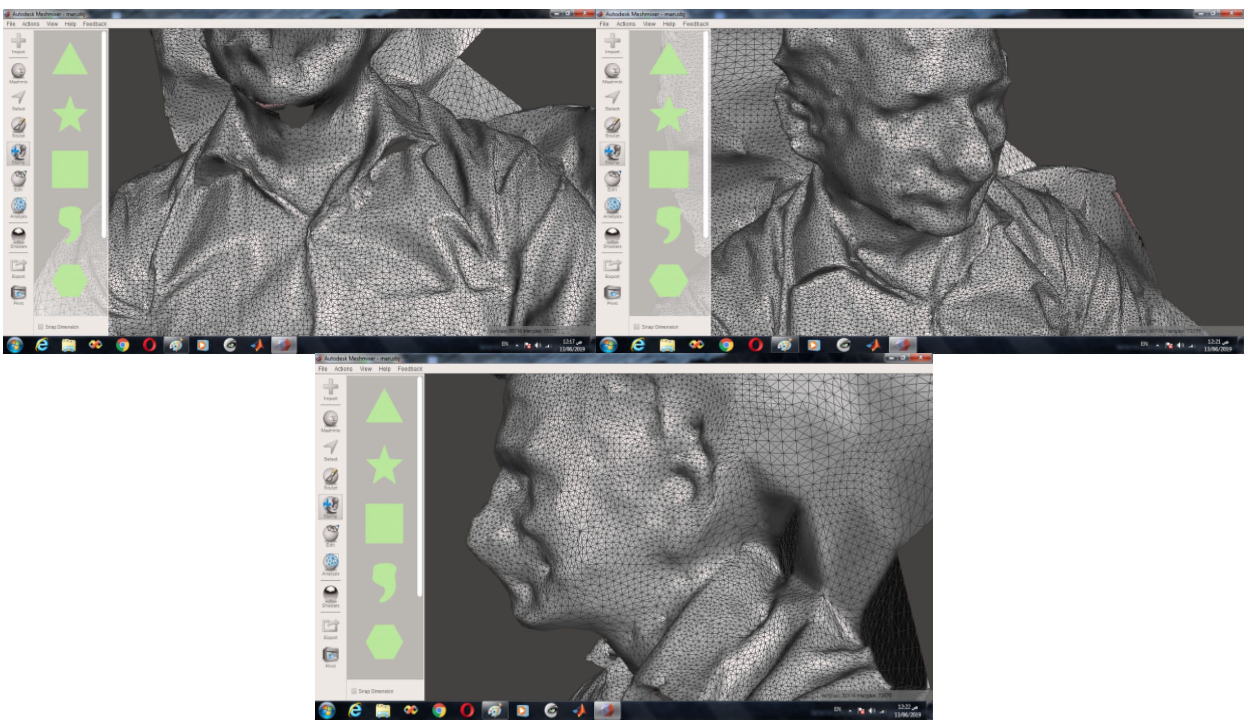
(b) The 3D Mesh for Soft Toy showed by 3D MeshMixer software

Figure 12: (a) and (b) show the 3D object and the mesh surface for the decompressed Soft Toy images by our proposed image compression at higher compression ratio up to 97.6%

Figure 13: Decompressed 20 frames of Face images, dimensions for each image = 2448 x 3264.



(a) Face frames converted to 3D object by using 3DF Zephyr Software



(b) The 3D Mesh for Face showed by 3D MeshMixer software

Figure 14: Show the 3D object and the mesh surface for the decompressed Face images by our proposed image compression algorithm at higher compression ratios up to 98.5%

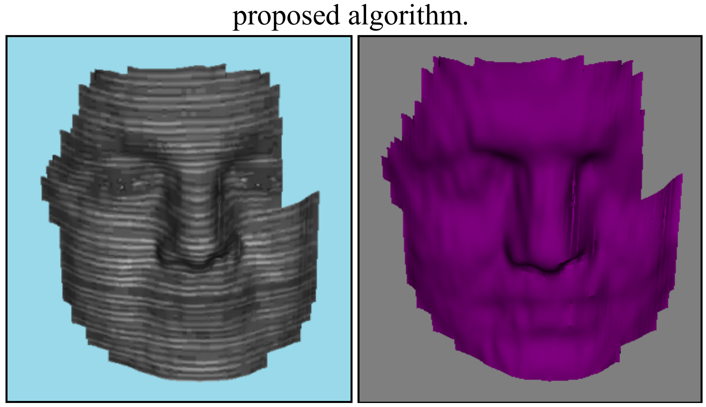
5. Comparison with the JPEG Technique for Structured Light Images

Figures 15, 16 and 17 show a 3D mesh surface reconstruction for each decompressed image: Face1, Face2, and Face3 using the traditional JPEG technique in direct comparison with our proposed method. Wavefront's Object (.OBJ) format for 3D structures is also quoted for comparison to stress how much storage space and transmission bandwidth can be saved by adopting our method. Table 3 summarizes the experimental results.

Table 3. Results of traditional JPEG technique and OBJ file sizes for grayscale structured light images compared to our proposed algorithm

Frame name	Our Proposed Compression Algorithm	JPEG technique Compression			3D Object (.OBJ) file size
		Compressed Size	RMSE	Average PSNR	
Face1	14.0 KB	22.0 KB	7.0	39.6	6.77 MB
Face2	17.9 KB	32.1 KB	7.96	39.1	6.6 MB
Face3	21.1 KB	37.8 KB	10.6	37.8	6.67 MB

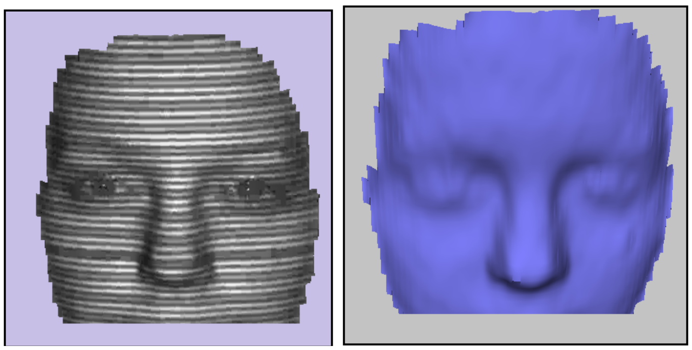
(a) Left - Decompressed image by traditional JPEG technique, (right) Decompressed image by our proposed algorithm.



(b) Reconstructed 3D mesh surface from decompressed 2D "Face1" image by traditional JPEG technique

Figure 15: (a) and (b) 3D mesh surface reconstruction for "Face1" by traditional JPEG technique

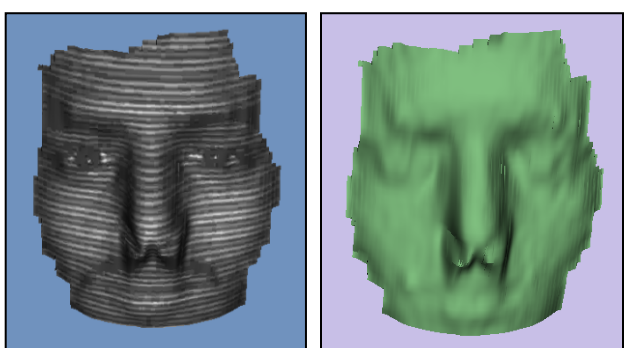
(a) Left - Decompressed image by traditional JPEG technique, (right) Decompressed image by our proposed algorithm.



(b) Reconstructed 3D mesh surface from decompressed 2D "Face2” image by traditional JPEG technique

Figure 16: (a) and (b) 3D mesh surface reconstruction for "Face2" image by traditional JPEG technique

(a) left - Decompressed image by traditional JPEG technique, (right) Decompressed image by our proposed algorithm.



(b) "Face3" reconstructed 3D mesh surface from decompressed 2D image by traditional JPEG technique
Figure 17: (a) and (b) 3D mesh surface reconstruction for "Face3" by traditional JPEG technique

It's clearly noticeable from the results that the reconstructed 3D surfaces using the traditional JPEG technique are substantially degraded with artifacts and missing parts in different areas of the surfaces. This is because the JPEG technique is unable to compress images at higher compression ratios up to 99% (in "face 1") and 98% (in "face 2" and "face 3"). Our proposed algorithm demonstrates better visual properties compared to the JPEG technique, higher compression ratios, while capable of keeping image details suitable for 3D reconstruction.

6. Comparison with the JPEG Technique for Multiple Viewpoint Images

The next three Figures 18, 19, and 20 illustrate 3D mesh surface reconstruction of the three sets of multiple viewpoint images by the traditional JPEG technique compared with our proposed method. Table 4 summarizes the experimental results where the 3D file size in object format are also quoted for comparison.

Table 4. Results of the traditional JPEG technique for multiple viewpoint images compared to our proposed algorithm

Sequence	Our Proposed Developed JPEG Compression Algorithm	JPEG technique Compression			3D Object (.OBJ) file size
		Compressed Size	Average RMSE	Average PSNR	
Apple	3.1 MB	4.77 MB	7.4	39.4	15.65 MB
Soft Toy	2.39 MB	5.59 MB	5.7	40.5	28.8 MB
Face	2.25 MB	3.37 MB	11.1	37.6	71.3 MB

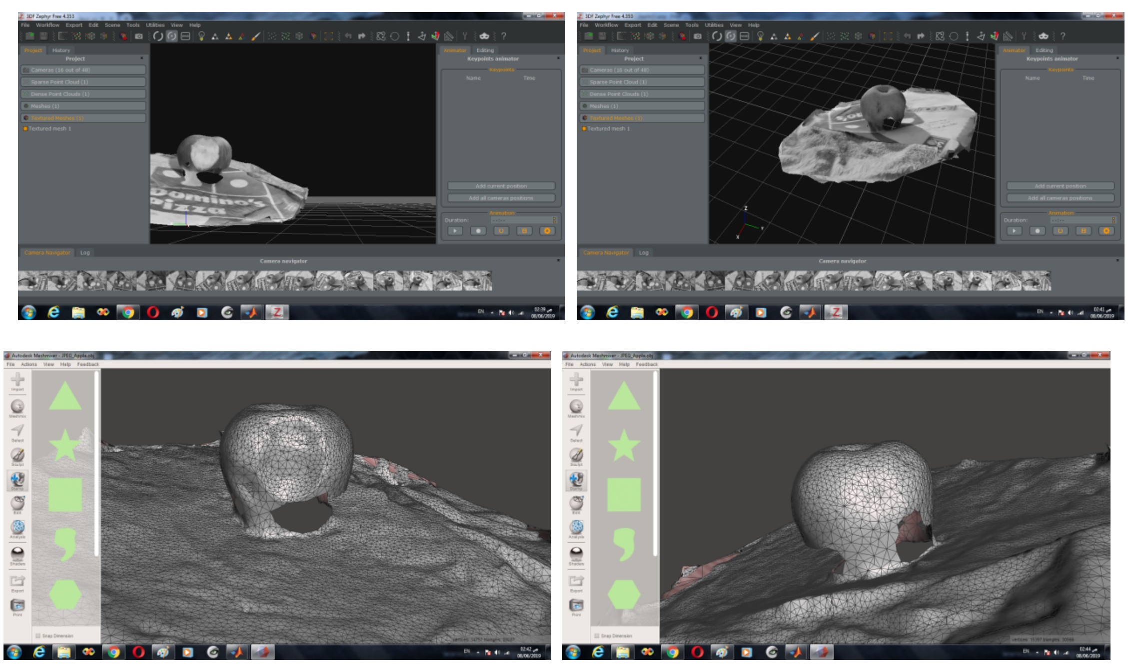


Figure 18: (Top) 3D object for Apple decompressed by the traditional JPEG algorithm. (Bottom) 3D mesh reconstruction at higher compression ratio of 94%

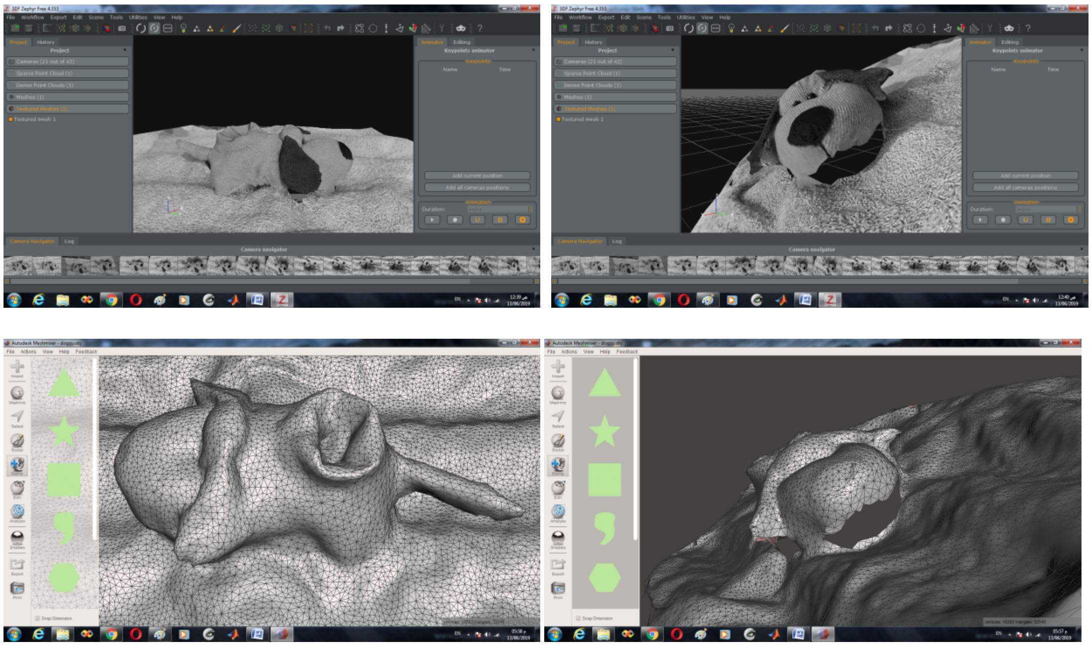


Figure 19: (Top) 3D object for Soft Toy decompressed by the traditional JPEG algorithm. (Bottom) 3D mesh reconstruction at higher compression ratio of 97%

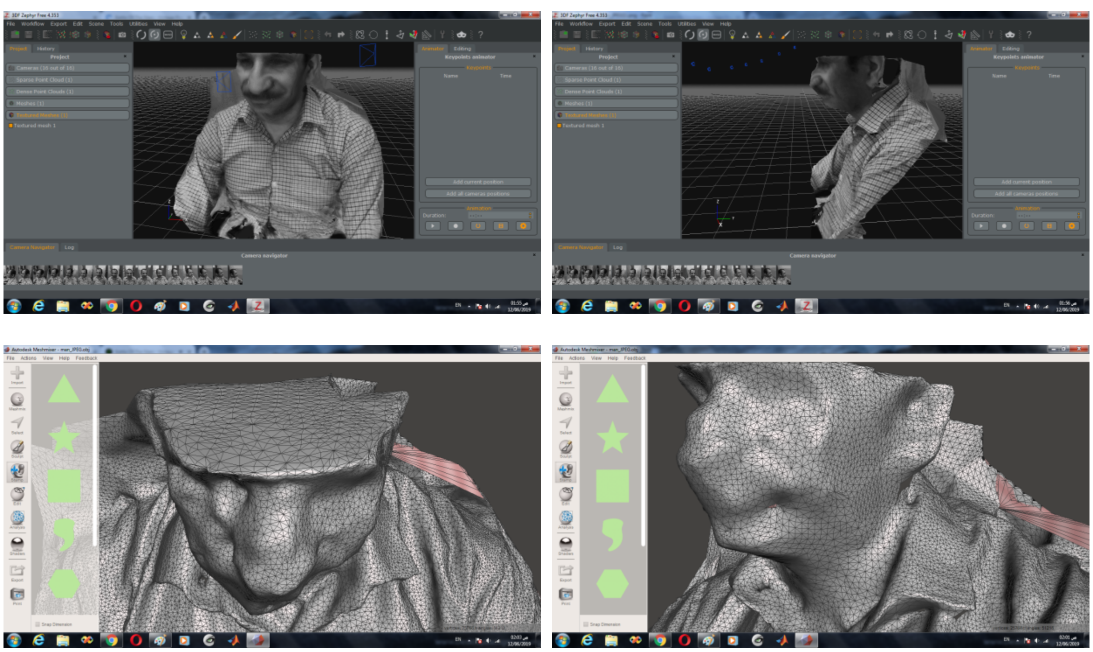
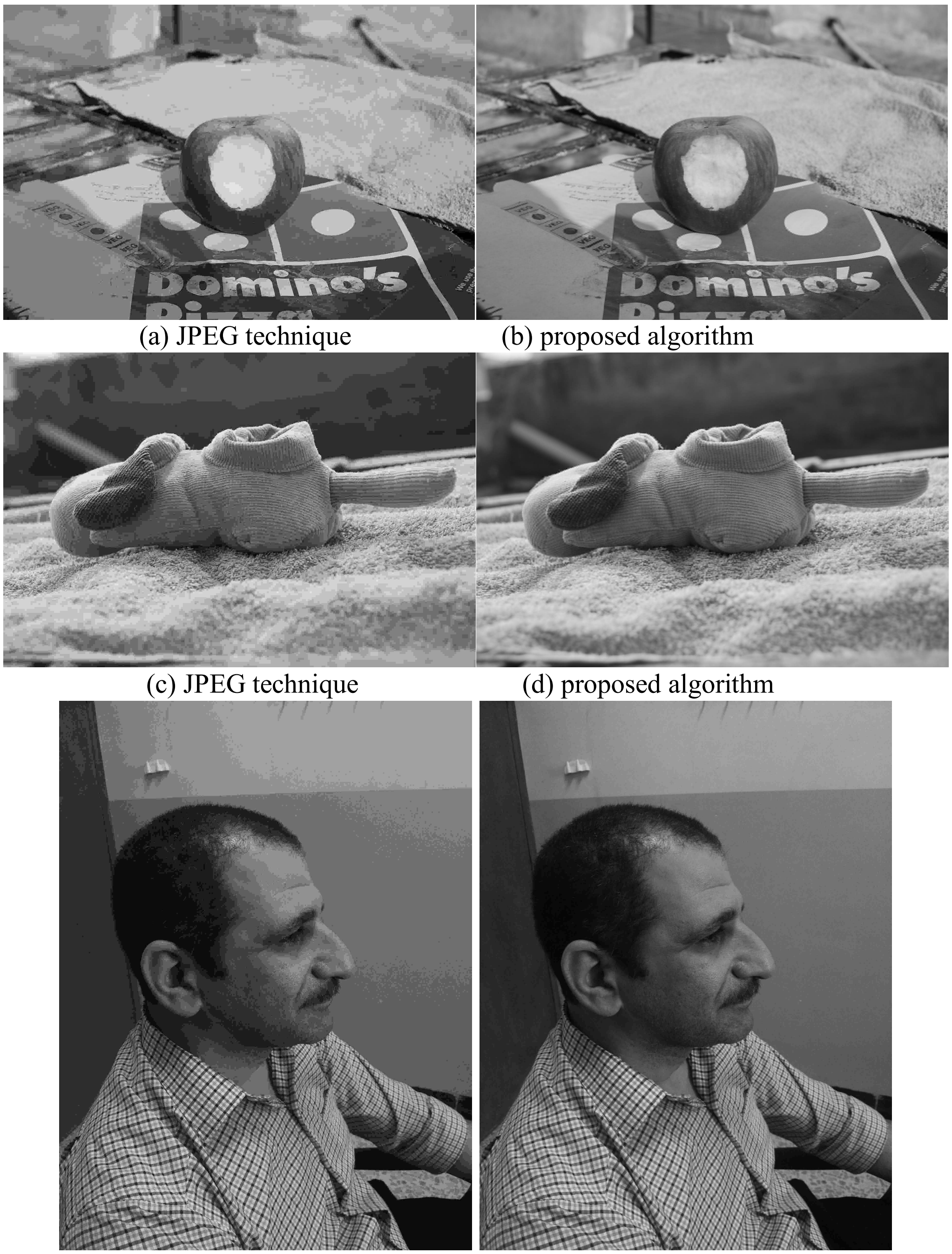


Figure 20: (Top) 3D object for Face image decompressed by the traditional JPEG algorithm. (Bottom) 3D mesh reconstruction at higher compression ratio of 94%

From the figures it is obvious that the traditional JPEG technique is inefficient to compress images at high compression ratios for sequence of images for 3D reconstruction from multiple viewpoints. All 3D reconstructions show artifacts and missing surface information details at high compression ratios. While our proposed method can successfully achieve compression ratios of up to 99%, JPEG at 97% yields such corrupted 3D meshes that we can state it fails. Finally, Figure 21 shows further details from a perceptual image quality assessment of our proposed algorithm compared with JPEG.



(a) JPEG technique (b) proposed algorithm

(c) JPEG technique (d) proposed algorithm

(e) JPEG technique (f) proposed algorithm

Figure 21: (a, c and e) Perceptual assessment: the decompressed image quality shows artifacts at higher compression ratios by the JPEG algorithm, while the other decompressed images (b, d and f) kept the quality at higher compression ratios by our proposed algorithm.

7. Conclusion

In this paper, we have introduced and demonstrated a new compression algorithm and demonstrated its superior quality through a direct comparison with the JPEG technique for 3D mesh reconstruction from structured light images and from sequences of multiple viewpoint images. The contributions of this paper are achieving high compression ratios without adversely affecting 3D reconstruction and with minimum loss using a single transformation with less complexity as compared to existing algorithms in the literature. The method is based on applying a single level DCT to obtain the DC and AC coefficients and then quantizing the frequency domain coefficients using an optimized quantization matrix. The coefficients matrix is then subjected to a series of operations to increase high frequencies and eliminating zeros to attain high compression ratios and a precise 3D reconstruction. The developed algorithm has been tested with images of various sizes and multiple viewpoint images within the context of 3D reconstruction. The results demonstrated that our approach yields a higher quality reconstruction of 3D surfaces than the traditional JPEG technique. The reconstructed 3D surfaces of the traditional JPEG technique showed degradation and missing information in some parts of surfaces in higher compression ratios, while our proposed algorithm illustrated better visual properties.

Acknowledgments

We gratefully acknowledge the Communication and Computing Research Centre (CCRC) and the Research and Innovation Services (RIS) at Sheffield Hallam University for their support.

References

[1] Yongjie Jessica Zhang (2016) - Geometric Modeling and Mesh Generation from Scanned Images - CRC Press, ISBN 13: 978-1-4822-2777-2.

[2] Mario Botsch, Leif Kobbelt, Mark Pauly, Pierre Alliez and Bruno Levy (2010) - Polygon Mesh Processing- A K Peters, ISBN 978-1-56881-426-1.

[3] A. MAGLO, G. LAVOU´E, F. DUPONT and C. HUDELOT (2013), 3D mesh compression: survey, comparisons and emerging trends ACM Computing Surveys, Vol. 9, No. 4, Article 39.

[4] RODRIGUES, Marcos, ROBINSON, Alan and OSMAN, Abdulsslam (2011). 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.

[5] Marcos Rodrigues, Abdusslam Osman and Alan Robinson (2013), Partial Differential Equations for 3D Data Compression and Reconstruction, Advances in Dynamical Systems and Applications, Volume 8, Number 2, pp. 303–315.

[6] M. M. Siddeq, M. A. Rodrigues (2016), 3D Point Cloud Data and Triangle Face Compression by a Novel Geometry, Proceedings of the international conference on computational methods, 3, 379-394.

[7] SIDDEQ, Mohammed and RODRIGUES, Marcos (2017), DCT and DST based Image Compression for 3D Reconstruction. 3D Research, 8 (5), 1-19.

[8] SIDDEQ, M. M. and RODRIGUES, Marcos (2014). A novel image compression algorithm for high resolution 3D reconstruction. *3D research*, 5 (7).

[9] M. Siddeq (2010), JPEG and Sequential Search Algorithm applied on Low- Frequency Sub-Band for Image Compression (JSS), Journal of Information and Computing Science, Vol. 5, No. 3, 2010, pp. 163-172.
[10] Siddeq and Rodrigues (2017), A novel high-frequency encoding algorithm for image compression, EURASIP Journal on Advances in Signal Processing 2017:26.
[11] M.M. Siddeq and M.A. Rodrigues, (2014) A New 2D Image Compression Technique for 3D Surface Reconstruction, 18th International Conference on Circuits, Systems, Communications and Computers, Santorin Island, Greece: 379-386.
[12] Sheffield Hallam University, M.M. Siddeq and M.A. Rodrigues, (2016), Image Data Compression and Decompression Using Minimize Size Matrix Algorithm. WO 2016/135510 A1.
[13] M.M. Siddeq and RODRIGUES, Marcos (2015). 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, 6 (3), p. 26.
[14] Gregory K. Wallace (1992), The JPEG Still Picture Compression Standard, IEEE Transactions on Consumer Electronics, Vol. 38, No. 1.
[15] K. Sayood (2003), Lossless Compression Handbook. Academic Press, ISBN: 9780126208610.
[16] M. Siddeq (2010), JPEG and Sequential Search Algorithm applied on Low-Frequency Sub-Band for Image Compression (JSS), Journal of Information and Computing Science,Vol. 5, No. 3, pp. 163-172
[17] K. R. Rao and P. Yip (1990), Discrete Cosine Transform. Academic Press, ISBN: 978-0-12-580203-1.
[18] Ankit Chouhan and M.J. Nigam (2016), Double Compression of JPEG Image using DCT with Estimated Quality Factor, 1st IEEE International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES).
[19] N. Ahmed, T. Natarajan and K. R. Rao (1974), Discrete Cosine Transforms. IEEE Transactions Computer, C-23 90-93.
[20] David Salomon (2007), Data Compression: The Complete Reference, Fourth Edition, Springer, ISBN: 1-84628-602-6.
[21] Neethu.K.J and Sherin Jabbar (2015), Improved Quality of JPEG Compressed Image Using Approximate K-SVD Algorithm, IEEE 2nd International Conference on Innovations in Information Embedded and Communication Systems.
[22] M Rodrigues, M Kormann, C Schuhler, P Tomek (2013), Robot trajectory planning using OLP and structured light 3D machine vision, in Lecture notes in Computer Science Part II. LCNS, 8034 (8034) (Springer, Heidelberg, pp. 244–253.
[23] M Rodrigues, M Kormann, C Schuhler, P Tomek (2013), Structured light techniques for 3D surface reconstruction in robotic tasks, in Advances in Intelligent Systems and Computing, ed. by J KACPRZYK (Springer, Heidelberg), pp. 805–814.
[24] M Rodrigues, A Osman, A Robinson (2013), Partial differential equations for 3D data compression and reconstruction. Journal Advances in Dynamical Systems and Applications 12(3), 371–378.
[25] 3DF-Zephyr, https://www.3dflow.net/3df-zephyr-pro-3d-models-from-photos/ , last access Janiuary 2020.
[26] 3D MeshMixer, http://www.meshmixer.com/ , last access Janiuary 2020.