Sheffield Hallam University

A novel Hexa data encoding method for 2D image crypto-compression

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

Published version

Copyright and re-use policy

A novel Hexa data encoding method for 2D image crypto-compression

Mohammed M. Siddeq [1] · **Marcos A. Rodrigues** [1]

Abstract

We proposed a novel method for 2D image compression-encryption whose quality is demonstrated through accurate 2D image reconstruction at higher compression ratios. The method is based on the DWT-Discrete Wavelet Transform where high frequency sub-bands are connected with a novel Hexadata crypto-compression algorithm at compression stage and a new fast matching search algorithm at decoding stage. The novel crypto-compression method consists of four main steps: 1) A five-level DWT is applied to an image to zoom out the low frequency sub-band and increase the number of high frequency sub-bands to facilitate the compression process; 2) The Hexa data compression algorithm is applied to each high frequency sub-band independently by using five different keys to reduce each sub-band to 1/6 of its original size; 3) Build a look up table of probability data to enable decoding of the original high frequency sub-bands, and 4) Apply arithmetic coding to the outputs of steps (2) and (3). At decompression stage a fast matching search algorithm is used to reconstruct all high frequency sub-bands. We have tested the technique on 2D images including streaming from videos (YouTube). Results show that the proposed crypto-compression method yields high compression ratios up to 99% with high perceptual quality images.

Keywords 2D image compression · DWT · Hexadata encoding · Fast matching search

1 Introduction

Image compression has acquired increased importance within the context of data streaming, satellite communications, and remote sensing techniques. Large transmission bandwidths

✉   Marcos A. Rodrigues
     M.Rodrigues@shu.ac.uk

     Mohammed M. Siddeq
     mamadmmx76@gmail.com

[1]   GMPR-Geometric Modelling and Pattern Recognition Research Group, Sheffield Hallam University, Sheffield, UK

mean high cost and massive amounts of storage space, leading to the requirements for fast and efficient image and data compression algorithms [4,5]. JPEG is a lossy coding method considered adequate as an image compression tool [1, 2] for many applications including photography, web browsing, image editing and printing. However, with the development of multimedia technologies and the need to secure data while reducing the amount of bandwidth and storage space, research on multimedia compression and encryption have become hot topics. Due to the characteristics of large data volumes and real time requirements, traditional ciphers need improvements to be efficiently applied to data.

The latest international image compression standards JPEG2000 [7, 9] is widely used and it is our purpose to develop a secure compression method based on JPEG2000 codec. Some algorithms have been reported [8, 10] in the literature keeping the file format unchanged and at low computational cost, but not secure. If one adds standard encryption/decryption to these algorithms, processing time becomes a major bottleneck for real time image and video processing and transmission or streaming. There are other aspects to consider such as the time required for coding/decoding and other processing requirements [11, 12]. It is acknowledged in the literature the it is still difficult to perform compression and encryption in a time-efficient way as noted by Chang and Liu [13]. In an attempt to reduce processing time, methods have been proposed combining compression and encryption into a single process [14, 15] resulting in high computationally intensive methods which are, above all, insecure techniques. In video processing, encryption and decryption times can be reduced by using partial or selective encryption. Partial encryption can be very effective to many applications since high speed encryption is obtained by selectively encrypting sensitive data components [16, 17].

Our aims are to develop image compression methods that yield high compression ratios with high perceptual quality of image, secure and of fast execution. In this paper, we build on the concept of compression with partial encryption by proposing a method of data compression that generates a set of data-dependent compression keys. The proposed method is based on converting sets of six data items into single floating-point values derived from the data-dependent keys. An overview of the method is depicted in Fig. 1. A 5-level DWT-Discrete Wavelet Transform is applied to the image. The fifth level LL of the DWT is subjected to arithmetic coding while the other levels representing high frequency coefficients are combined together and subjected to the Hexadata method proposed here, then zeros are removed from the data followed by arithmetic coding.

The Hexadata method generates a set of compression keys. Lest the keys not be compromised, the method is a secure crypto-compression on its own. It is thus, a per-file crypto compression method which, for security in depth, only the keys need be further encrypted by the AES-Advanced Encryption Standard method providing absolute security while not compromising the speed of decoding. However, it is not in the scope of this paper to discuss the extra AES step, neither to demonstrate mathematically the security of the generated keys. The remaining of this paper is organized as follows. Section 2 introduces the Discrete Wavelet Transform and Section 3 describes the Hexadata method for compression and decompression. Experimental results are presented in Section 4 and conclusions in Section 5.

2 The discrete wavelet transform (DWT)

Recent image compression methods, especially at low bit rate, play a significant role in applications such as storage on low memory devices, data transmission over narrow-band channels, wireless
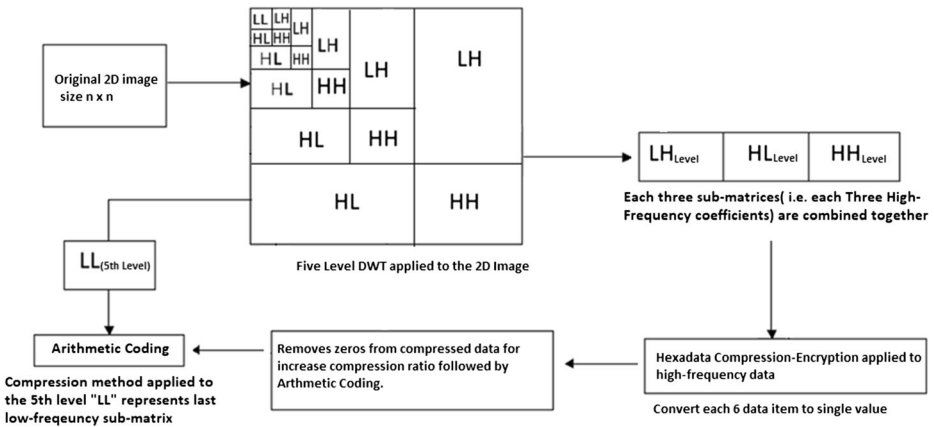
Fig. 1 The proposed 2D image crypto-compression

communication and data streaming over the Internet. It has been shown in the literature that wavelet-based coding can provide improvements in image quality at higher compression ratios.

DWT is a multi-resolution decomposition technique of the input signals. Signals are first decomposed into two subspaces namely low-frequency sub-band (achieved through low-pass filtering) and high-frequency (high-pass filtering) sub-band. For the classical DWT, a low-pass digital filter and a high-pass digital filter implement the forward decomposition of a signal. Both the digital filters are derived using the scaling function and the corresponding wavelets [3]. The system down-samples the signal by banks of half-band filters resulting in the decomposition process [6].

In this work we use a five–level DWT decomposition to reduce the low-coefficient values (i.e. zoom-in LL size) and to increase the number of high-frequency coefficients. Additionally, the high-frequencies at the first level (i.e. LH1, HL1 and HH1) are set to zero. This is because higher image resolution helps the DWT to remove insignificant coefficients from high-frequency sub-bands and the ability of DWT to recover approximately the original 2D image without the need for LH1, HL1 and HH1 sub-bands [18–20]. All the others high-frequency sub-bands in the lower levels are compressed by our approach as illustrated in Fig. 1.

After a five-level DWT is applied to the 2D image, a quantization process is applied to each sub-band independently. The quantization process is based on dividing a sub-band by a uniform value, as described by the following equation:

$$H_Q = round\left[\frac{H}{Q}\right] \tag{1}$$

Where $0 < Q <= Max$. The maximum value (Max) of Q is the maximum of a high-frequency sub-band which means that Q is strictly data-dependent, and H_Q represents a quantized High-frequency sub-band. Here Q is the quantization value, which means all the coefficients are divided by a single value Q then rounded to integer. Due to the properties of the DWT, each Low-frequency sub-band is quantized by Eq. (1) (i.e. $L_Q = round(L/Q)$) before being decomposed into further sub-bands.

3 The Hexadata crypto-compression method

The GMPR-Geometric Modelling and Pattern Recognition Research proposed Hexa-data compression is a method to perform data compression with high compression ratios. The main novel step is that from a data represented as an array or a matrix of integer values, divide the data into blocks of 6 items, and convert each such set to a single floating-point through hierarchical two-level key sets. The steps in the algorithm are:

1. Generate three keys and convert each three data items to a single value by multiplying each value by their respective key and sum over the result. The output from such triplet encoding is an ordered list or array of coded values [18, 20]. The following equation represents triplet encoding:

$$e(i) = K1 \times d(n) + K2 \times d(n+1) + K3 \times d(n+2) \qquad (2)$$

Where $e(i)$ is the encoded output from a stream of original data $d(n)$.

2. The output from step 1 is converted to another stream of encoded data by multiplying each value by their respective second level keys as shown in Fig. 2 and the following equation represents the second level encoding:

$$C(j) = K4 \times e(p) + K5 \times e(p+1) \qquad (3)$$

Where $C(j)$ is the final encoded output from previous encoded data $e(i)$. The key values K1, K2, K3, K4 and K5 are generated by a key generator algorithm according to the following steps [18, 20]:

During crypto-compression, the probabilities of compressed data are computed resulting in a data set that we call Limited-Data [18, 20], which are used later at decompression stage as shown in Fig. 3.

3.1 The crypto-compression step

In this section we illustrate the encoding steps. Assume we have the following input: Data = [65, 8, 90, 8, 100, 12, 1, 0, 8, 23, 0, 9, 1, 0, 8, 23, 0, 9] (data size is 18). The keys for the first level are generated from the key generator algorithm: L1 = [1, 169, 28,390]. The keys for the second level are generated from a random number between 0 and 1: L2 = [1, 0.905] Where: L1 = [K1, K2, K3] and L2 = [K4, K5].

In Fig. 4, we stress that sorting by the first data item is a very important function in our proposed method as it is used in the decompression stage for fast search. Meanwhile, the relevant 6 data items are saved in a Limited-Data array, because these data will later be used to recover the original data. The Limited-Data array is sorted by the insert sort algorithm as each data item is generated. It is one of popular sorting methods [21] with linear time complexity $O(n)$ given that data are sorted in ascending order as they are included in the array.

The main reason to remove the output and replace by "Nil", is to reduce the size of the Limited-Data array. Also, the outputs required to decrypt the data and can be separately encrypted for higher security such as by the AES method. The last step in the algorithm is

```
M=(max(HQ))/2;      //max is a function outputting the maximum value
K1=1;               //set first 0 < key <=1
K2=K1+M+F;          //where F >=1 is an integer scaling factor
K3=F*M*(K1+K2);     //where * is the dot multiplication operator
K4=rand;            //second level random key
K5=rand;            //second level random key
```

to apply Arithmetic Coding to the Limited-Data and the output to produce a stream of compressed data.

Some observations are made as follows. In the example above, L2 seems to have many data items and may be wrongly interpreted as a handicap to compression. But the fact is that most of the data are repeated in 2D images which help to achieve a high compression ratio. In a five level DWT transform it would normally mean that more than 50% of data are zero or insignificant. The approach works well and yields compression ratios up to 99% with good perceptual quality of the image as shown in Section 5.

3.2 The decoding step

The decoding algorithm represents the inverse of Hexa-data crypto-compression algorithm. It starts by computing the summation of each 6 relevant data from Limited-Data through equation Eq. (3) by using the same keys (K1 – K5). Figure 5 illustrates the decoding steps.

Decompression begins by matching each generated set of possible data with the outputs in the Limited-Data and, if the match is successful, the result contains the values we are after namely the relevant 6 data items representing the decoded data. It is a simple search within a look up table with time complexity of $O(log\ n)$, because the array is sorted and a binary search algorithm is used for fast recovery.

Note that a binary search requires the Limited-Data be sorted in ascending order according to the output. The algorithm compares each item of compressed data with the middle value Limited-Data. If the values match, then its relevant 6 data items are returned (i.e. the 6 relevant
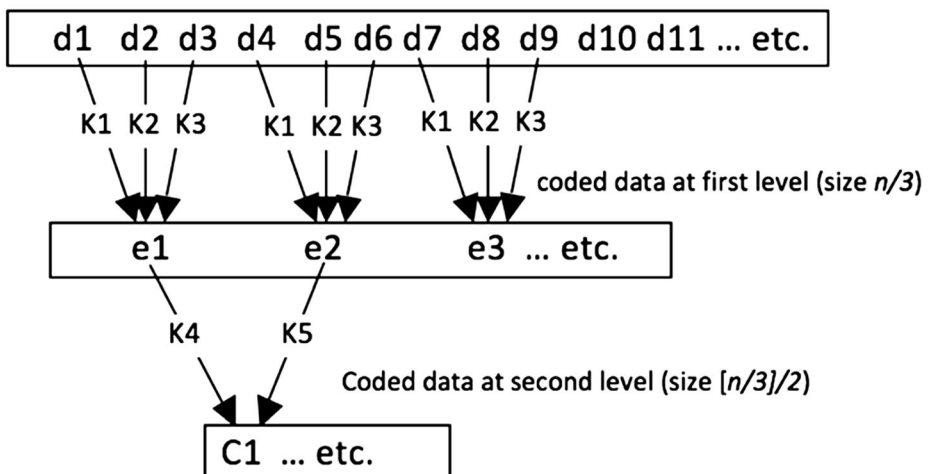

Fig. 2 The structure of the Hexa-data lossless crypto-compression algorithm

Input data: d1, d2, d3, d4, d5, d6, d7, d8 ...

Compression-Encryption algorithm

Example of Compressed Data

C1, C2, C1, C3, C1, C2, C1, C3

Limited Data (i.e. unique data from the compressed set)
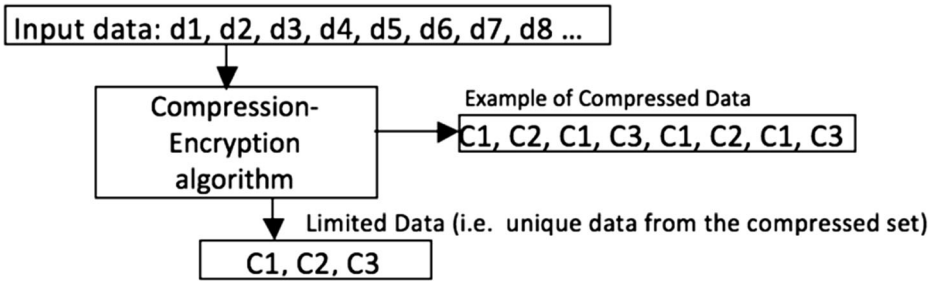
C1, C2, C3

Fig. 3 Limited-Data represents the unique data set from the compressed data. This set represents the space search, which is used at decompression stage

Original data (size 18)

65, 8, 90, 8, 100, 12, 1, 0, 8, 23, 0, 9, 1, 0, 8,

K1 K2　K1 K2　K1 K2　K1 K2　K1 K2　K1 K2

Output using L1 keys　　　　　　　　　Output (size 6)

K4　　K5　　K4　　K5　　K4　　K5

2880134.14,　458378.365,

The compressed-encrypted data from

Limited Data Computed.

(output)					(relevant 6 data)	
2880134.14	65	8	90	8	100	12
458378.365	1	0	8	23	0	9

After sorting.

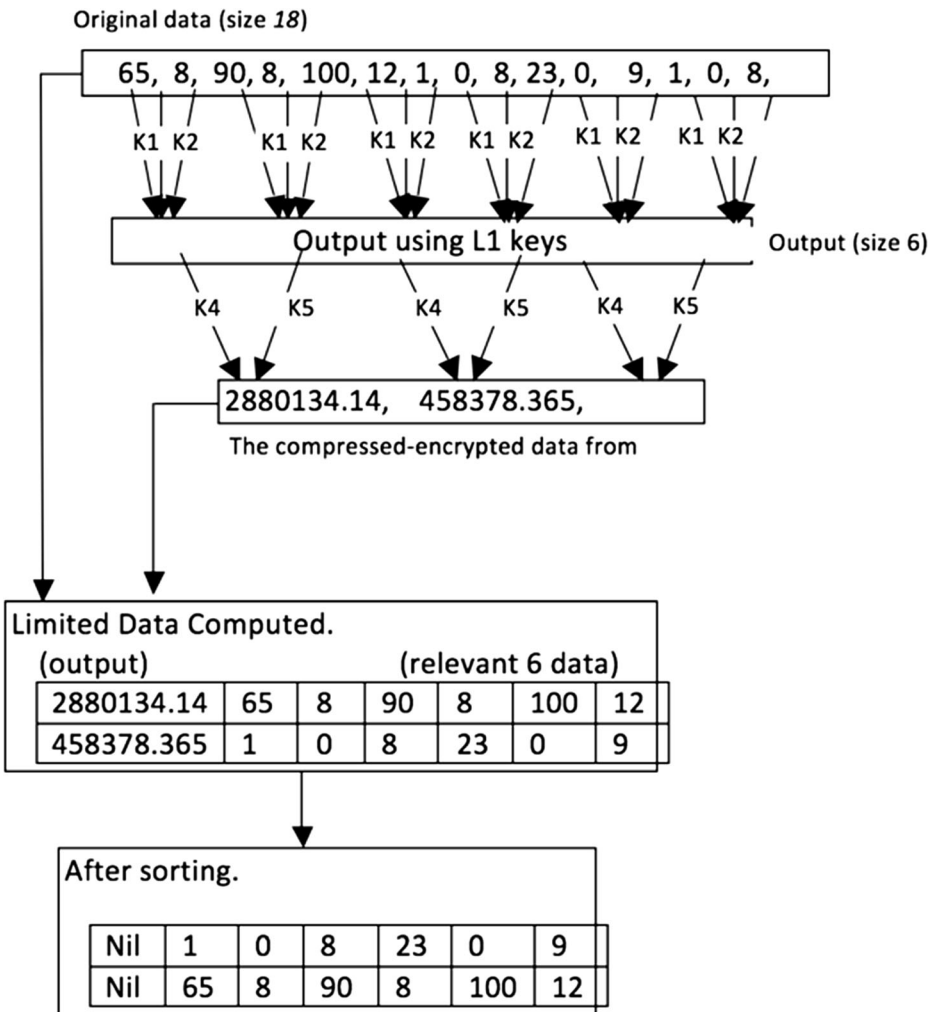| Nil	1	0	8	23	0	9
Nil	65	8	90	8	100	12

Fig. 4 Example of the Hexadata coding algorithm at work

items are the decoded data). Otherwise, if the data value is less than the middle element of the Limited-Data, then the algorithm is repeated on the sub-array to the left of the middle element or, if the value is greater, on the sub-array to the right [21]. The possibility of a "Not Matched" does not exist, because the Limited-Data has been previously built at compression time containing one item of each original data. This decoding method runs much faster than our previous algorithm proposed in the Patent WO 2016/135510A1 [22].

3.3 Removing Zeros from high-frequency sub-band

After the five level DWT is applied to the 2D image, at 5th level the LL5 size is very small and can be reduced to a few bytes by Arithmetic Coding. Meanwhile, all other high-frequency sub-bands are encoded by the proposed Hexa-data crypto-compression method. In order to achieve higher compression ratios, the output from the Hexadata algorithm can still be further compressed by removing zeros from the encoded data.

To separate zeros from nonzero data, a zero-array is computed by detecting the number of zeros between two nonzero values. For example, assume that a data set H = {29,023.567, 0, 0, 0, 9457.334, 0, 0, 0, 0, 0, −7123.123} represents an encoded high-frequency sub-band. The zero-array will be Z = {0,3,0,5,0} where the zeros are placeholders for nonzero data and the other numbers are the zero counts between two consecutive non-zero data. Techniques to further increase the compression ratio can be applied here, such as replacing the number "5" with "3" and "2" to increase the probability (i.e. the re-occurrence) of redundant data [18]. This would yield a new equivalent zero-array as Z = {0,3,0,3,2,0}. Finally, each encoded high-
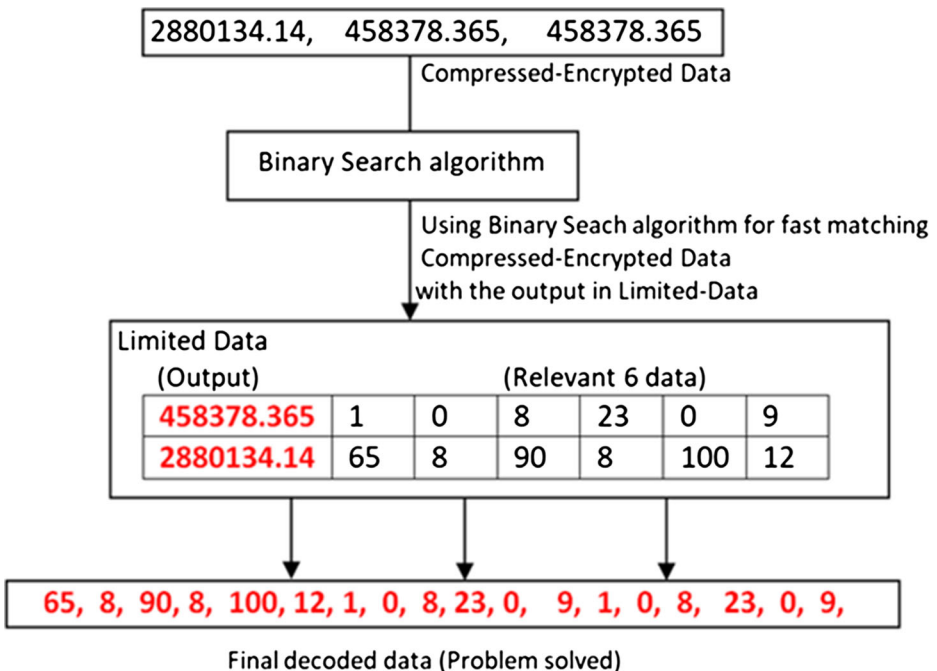

Fig. 5 Hexadata decoding steps

frequency sub-band has its own zero-array and nonzero-array and each array is compressed by arithmetic coding for higher compression ratios.

4 Experimental results

The proposed image decoding starts by recovering the high-frequency sub-bands. At the 5th level the high-frequency sub-bands (LH5, HL5 and HH5) are recomposed with LL5 (i.e. apply inverse DWT) to generate new LL4 which is recomposed with its relevant high-frequencies. This process continues until the last LL at the 1st level is recovered. In the process, all other levels of LL, HL, LH and HH are recovered completing thus, the 2D image decoding. The experimental results described here were implemented in MATLAB R2014a running on an Intel Core i7-3740QM microprocessor (4 cores).

Results are described in two ways: first, the method is applied to images of different sizes and their visual quality are assessed with Root Mean Square Error (RMSE) and Peak Signal to Noise Ratio (PSNR). Second, the proposed compression technique is applied to a stream of video images. Table 1 shows the first part of results by applying the crypto-compression method to three selected images shown in Fig. 6.

In Table 1, "L" and "H" refer to quantization value "Q" (see Eq. 1), which is applied to both low and high frequency sub-bands in each level respectively. Meanwhile, "N" refers to "No quantization" for the high-frequency sub-band at the first level DWT – otherwise image quality would be seriously compromised.

Figure 6 illustrates the perceptual quality of decoded images. The images on the left are decoded from crypto-compression with low compression ratios with high image quality, while decompressed images on the right refer to high compression ratios with good image quality. The results in Fig. 6 demonstrate that our proposed crypto-compression method is capable of compressing images at low bit rate while preserving image details.

Additionally, Table 2 shows our proposed Hexa-data compression applied to 2D images of various sizes for 3D reconstruction from a single image. The 3D mesh reconstruction is based on Convolutional Neural Networks (CNN), which depend on training on an appropriate set of 2D images and 3D facial models. The CNN technique works with a single 2D face image and camera lens alignment is not required (i.e. it does not need calibration for an image), it accepts arbitrary facial poses and expressions, and its output is the reconstructed 3D facial geometry [23]. After the Hexa-data method is applied, the decoded images are uploaded to the website [23] and after a few seconds a 3D model is returned as shown in Fig. 7. The main reason to test our method on 3D reconstruction is to provide another level of perceptual assessment of the quality of the compression and decoding methods. Anyone without any training can perform a visual assessment of the quality of a 3D reconstructed face. Furthermore, the widespread of 3D applications into VR-Virtual Reality and AR-Augmented Reality enabled by a range of 3D applications (i.e. Autodesk, 3Dmax, MeshLab, Geomagic, and a range of Animation studios) depend on 2D images.

In a second test, we applied our proposed crypto-compression method to a 720dpi video sequence. We used Windows movie maker application to grab images from the video and the compression method was applied to each image independently (i.e. we did not resort to other video compression techniques). Results are summarized in Table 3 and Fig. 8 shows the performance of the coding and decoding method respectively.

Table 1 Hexa-data crypto-compression results for selected 2D images

2D Image	Original Size	Compressed Size	Bpp-Bits per pixel	Quantization factor		RMSE	PSNR
				Y	CbCr		
Girl.BMP	6.59 MB	181 KB	0.026	L = [2 2 1 1 1] H = [N, 20,10,5,5]	L = [2 2 1 1 1] H = [N, 20,10,5,5]	6.4	40.0
Girl.BMP	6.59 MB	79.1 KB	0.011	L = [2 2 2 2 2] H = [N, 20,20,10,10]	L = [2 2 2 2 2] H = [N, 20,20,10,10]	8.3	38.9
Fish.BMP	23.5 MB	247 KB	0.01	L = [2 2 1 1 1] H = [N, 20,10,5,5]	L = [2 2 1 1 1] H = [N, 20,10,5,5]	2.6	43.9
Fish.BMP	23.5 MB	80.1 KB	0.0033	L = [2 2 1 1 1] H = [N, 20,10,5,5]	L = [2 2 1 1 1] H = [N, 20,10,5,5]	4.3	41.7
Relax.BMP	19.3 MB	348 KB	0.017	L = [2 2 1 1 2] H = [N, 10,10,10,5]	L = [2 2 1 2 2] H = [N, 20,20,10,5]	4.6	41.5
Relax.BMP	19.3 MB	190 KB	0.0096	L = [2 2 2 2 2] H = [N, 20,10,10,10]	L = [2 2 2 2 2] H = [N, 20,10,10,10]	6.1	40.2

Concerning crypto-compression as presented here, we do not perform a comparative analysis with other techniques because the final crypto-compressed file consists of the compressed data, arithmetic coding probabilities, compression keys, generated limited-data and Hexa-data probabilities. All such information is included for each image as a file header and the header itself can be as large as the actual compressed data. Normally the size of the header file can represent around 25% of the final crypto-compressed file. Since state-of-the-art methods do not have such headers as they are not designed with security in mind, it would not lead to a fair comparison: if header information is removed and only the actual compressed data sizes are compared it would be wrong, as our data cannot be decompressed without header information. If header information is added to the compressed data still would not be right as the size of the header is data-dependent and a direct comparison would yield inconclusive results as some files would be larger and some would be equivalent to standard methods.

The purpose of the research reported here is to demonstrate that the method can be applied to image and video compression where security (or sensitivity) of data is a concern. Furthermore, it is demonstrated that the method yields recovery of the original images with high quality. Although results reported in Table 3 are impressive in terms of compression ratios, we are working on improving these for image and video sequences with a follow on in-depth comparative analysis with state-of-the-art methods. Research is under way and results will be reported in the near future.

5 Conclusion

This paper has proposed and and demonstrated a novel method for image crypto-compression for 2D images and illustrated the quality of compression through objective measures such as

(a) Girl image original size 6.59 MB (1920 x 1200 pixels). Left: decompressed image with RMSE=6.4. Right: decompressed Image with RMSE=8.3.

(b) Fish image, original size 23.5 MB (3504 x 2336 pixels). Left: decompressed image with RMSE=2.6. Right: decompressed image RMSE=4.3.

(c) Relax image, original size 19.3 MB (3184 x 2124 pixels). Left: decompressed image with RMSE=4.6. Right: decompressed image with RMSE=6.1.

Fig. 6 Crypto-compressed and decoded by our proposed method. (**a**) Girl image original size 6.59 MB (1920 × 1200 pixels). Left: decompressed image with RMSE = 6.4. Right: decompressed Image with RMSE = 8.3. (**b**) Fish image, original size 23.5 MB (3504 × 2336 pixels). Left: decompressed image with RMSE = 2.6. Right: decompressed image RMSE = 4.3. (**c**) Relax image, original size 19.3 MB (3184 × 2124 pixels). Left: decompressed image with RMSE = 4.6. Right: decompressed image with RMSE = 6.1.

Table 2 Hexa-data Crypto-Compression results for 2D images used in 3D applications

2D Image	Original Size	Compressed Size	Bpp-Bits per pixel	Quantization factor		RMSE	PSNR
				Y	CbCr		
Face1.BMP	1.0 MB	14.1 KB	0.013	L = [1 2 2 2 2] H = [N 30 20 20 10]	L= [2, 3] H = [N 30 30 20 10]	4.8	41.3
Face2.BMP	7.17 MB	24.6 KB	0.0033	L = [2 2 2 2 1] H = [N 30 20 10 10]	L = [3 2 2 2 2] H = [N 30 30 20 10]	4.9	41.2
Face3.BMP	3.0 MB	60.1 KB	0.019	L = [1 1 2 2 1] H = [N 30 20 10 5]	L = [2 2 2 2 1] H = [N 30 30 20 5]	4.2	41.8

(a)   Uncompressed 2D images

(b)   Decoded 2D images from Hexadata crypto-compression

(c)   The decompressed 2D images converted to 3D faces

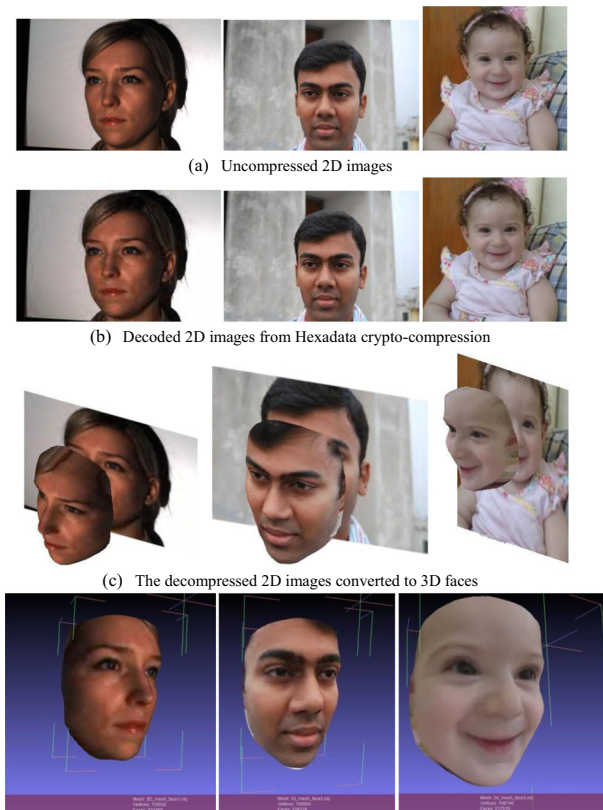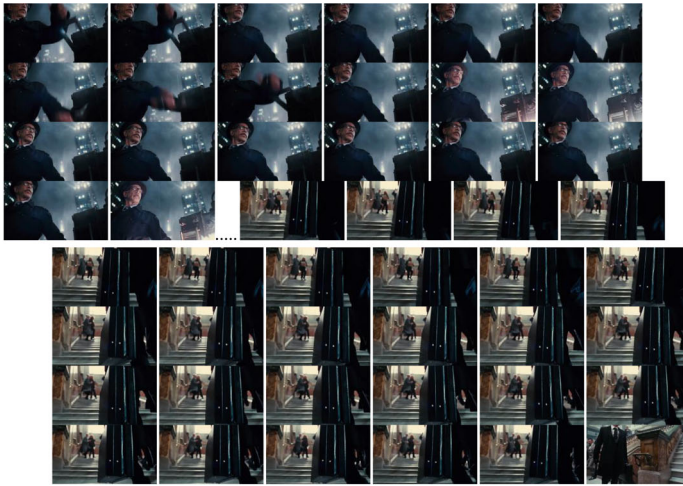**Fig. 7** (**a**) Uncompressed images: (Left) Face1 (744 × 480), (Middle) Face2 (1936 × 1296), and (Right) Face3 (1024 × 1024), (**b**) Compressed and decoded 2D images by our approach, (**c**) The decompressed 2D images used to reconstruct 3D faces by using the web 3D application, (**d**) 3D faces saved as 3D objects used in 3D applications
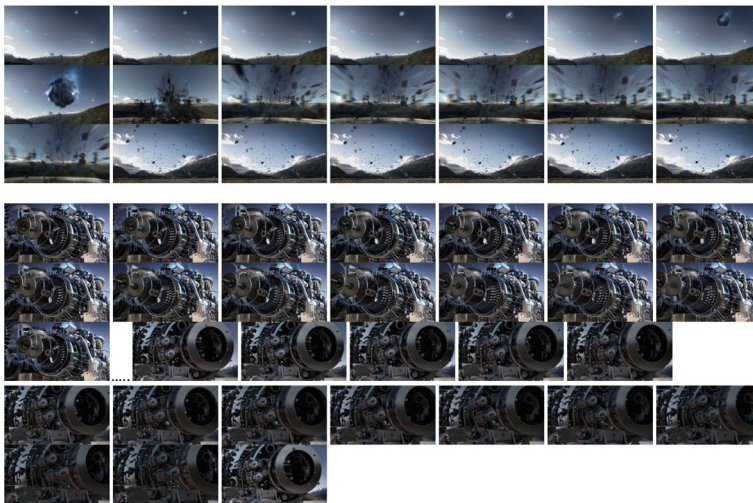
RMSE and PSNR and the perceived quality of the visualization. As per JPEG2000, our proposed method is based on DWT, but it is different in the way it is applied. It incorporates

Table 3 Hexa-data Crypto-Compression results for stream of 2D images

Video name	Total original size	Each frame size (1280 × 720)	No. of Frames	Movie duration (sec.)	Total crypto-compression size	Quantization factor		Average RMSE and PNSR for all the frames	
						Y	CbCr		
Video 1	1.15 GB	2.63 MB	452	22.6	19.7 MB	L = [1 2 2 2 1]	L = [1 2 2 2 2]	RMSE	PSNR
						H = [N 10 20 10 5]	H = [N 10 20 10 5]	3.2	43
Video 2	1.41GB	2.63 MB	550	27.5	37 MB	L = [1 1 1 2 2]	L = [1 1 2 2 2]	3.7	42.4
						H = [N 30 20 20 10]	H = [N 30 20 20 10]		

(a) Sample of decoded Video1 (JUSTICE LEAGUE) by our crypto-compression method. Video duration:
22.6s consisting of 452 frames [24]. Compression ratio=0.016

(b) Sample of decoded Video2 (Samsung Sound HD) by our crypto-compression method. Video duration:
27.5s consisting of 452 frames [25]. Compression ratio=0.019

Fig. 8 Decompressed-Decrypted Videos by our proposed method. (**a**) Sample of decoded Video1 (JUSTICE LEAGUE) by our crypto-compression method. Video duration: 22.6 s consisting of 452 frames [24]. Compression ratio = 0.016 (**b**) Sample of decoded Video2 (Samsung Sound HD) by our crypto-compression method. Video duration: 27.5 s consisting of 452 frames [25]. Compression ratio = 0.019

a number of additional steps at compression stage notably the minimization of high frequency encoding by the Hexa-data algorithm and fast matching search algorithm at decompression stage. The results showed that our approach yields good image quality with high compression ratios. Furthermore, it is demonstrated that it is able to reconstruct 3D surfaces based on 2D images also at high compression ratios. At the same time, the compressed image data are encoded by five different keys; this added a novel feature to our compression algorithm that

can be exploited in security applications. A summary of the method main advantages and disadvantages is given below.

Advantages:

- The method uses a five-level DWT to increase the number of high frequencies and keep a single low-frequency sub-band (i.e. the low-frequencies can compress to a few bytes depending on image size).
- At each level the high-frequencies sub-bands are combined together followed by the Hexadata method applied to the combined sub-bands to reduce the matrix size, leading to increased compression ratios, and coding the matrix by using five different keys. This is the main feature of our proposed algorithm lending it to secure data compression using partial encryption.
- At decompression stage, a fast matching search algorithm is used to decode the high-frequency matrix, using the five symmetric keys from the compression steps. The generation and use of such keys render the proposed method suitable for security applications, as data can only be decompressed with the keys. Another feature of the algorithm is that it runs much faster than our previous work proposed in the patent [22].
- The experiments show that the technique is suitable for real-time applications such as compression of 3D data and video data streaming. Moreover, compression ratios up to 99% can be achieved without significant image degradation. Equally, the method can be applied to video sequences with good image quality.

Disadvantages:

- The information needed to decode the data (keys, limited-data, image and block size in the form of a look up table) are kept in the header file and such information increases the compressed data size.
- The complexity of the crypto-compression steps is greater than that of existing codecs due to the coding of each six items of data which also increases the execution time for large images.

Further work includes the mathematical analysis of the crypto-compression method – that is, what would be the required effort to decode the image if the compression keys and other information in the look up table were not available. In addition, we are investigating making the AES an integral component of the crypto-compression method such that it can be widely used for secure data compression and encryption.

References

1. Christopoulos C, Skodras A, Ebrahimi T (2000) The JPEG sill image coding system: an overview. IEEE Trans Cons Elect 46:1103–1127
2. Akan R, Sıddıka Nilay Keskin in SN Applied Sciences (2019) The effect of data size of ANFIS and MLR models on prediction of unconfined compression strength of clayey soils. SN Appl Sci 1(6). Springer
3. Pan H, Siu W-C, Law N-F (2007) Lossless image compression using binary wavelet transform. IET Image Process 1(4):353–362
4. Taubman D, Marcellin MW, JPEG (2000) Image compression, fundamentals, standards and practice. Kluwer Academic Publishers, 2002
5. Duan X, Liu J, Zhang EJ (2018) Efficient image encryption and compression based on a VAE generative model. Real-Time Image Proc. https://doi.org/10.1007/S11554-018-0826-4
6. Enas K, Hassan E, George LE, Mohammed FG (2018) Color image compression based on DCT, differential pulse coding modulation, and adaptive shift coding. J Theor Appl Inf Technol 96(11)
7. Yang X, Sun Q, Wang T (2018) Image quality assessment improvement via local gray-scale fluctuation measurement. Multimed Tools Appl 77:14–32. Springer
8. Vizireanu DN, Preda RO (2005) A new digital watermarking scheme for image copyright protection using wavelet packets. In: 7th international conference of IEEE on telecommunications in modern satellite, vol 2. Cable and Broadcasting Services, pp 518–520
9. ISO/IEC 15 444-1: Information Technology JPEG 2000 Image Coding System –Part 1: Core Coding System (2000)
10. Wu Y, Ma D, Deng RH (2004) Progressiveprotectionof JPEG2000 Codestreams. In: Proceedings of IEEE conference on image processing, Singapore, pp 3447–3450
11. Watanabe O, Nakazaki A, Kiya H (2004) A fast Image-Scramble method using public-Key encryption allowing backward compatibility withJPEG2000. In: Proceedings of the 2004 international conference on image processing (ICIP 2004). IEEE, October 24–27, Singapore, pp 3435–3438
12. Grobois R, Gerbelot P, Ebrahimi T (2001) Authentification and access control in the JPEG2000 compressed domain. In: Proceedings of the SPIE 46th annual meeting, applications of digital image processing XXIV, July 29th–August 3rd
13. Naveen Kumar R, Jagadale BN, Bhat JS (2019) A lossless image compression algorithm using wavelets and fractional Fourier transform. SN Appl Sci 1(3). Springer
14. Lin Y, Amit A, Marcellin M, Bilgin A (2018) Task based JPEG 2000 image compression: an information-theoretic approach. In: Data compression conference, Snowbird, p 423423. https://doi.org/10.1109/DCC.2018.00076
15. Yang CN, Chou YS, Liu Y et al (2018) Exploiting error control in matrix coding-based data hiding over lossy channel. J Real-Time Image Proc. https://doi.org/10.1007/s11554-0180829-1
16. Neetu S, Abhinav G, Roop C (2018) Statistical characterization of block variance and AC DCT coefficients for power law enhanced images. IET Image Process 12(11):2128–2137
17. Darwish SM (2019) A modified image selective encryption-compression technique based on 3D chaotic maps and arithmetic coding. Multimed Tools Appl 78(14):19229–19252. Springer
18. Siddeq M, Rodrigues M (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 Res 6(3):26
19. Siddeq MM, Rodrigues M (2015) Applied sequential-search algorithm for compression-encryption of high-resolution structured light 3D data. In: Blashki K, Xiao Y (eds) MCCSIS: Multi-conference on computer science and information systems 2015. IADIS Press, pp 195–202
20. Siddeq M, Rodrigues M (2017) A novel high frequency encoding algorithm for image compression. EURASIP J Adv Signal Process 26:1–17. https://doi.org/10.1186/s13634-017-0461-4
21. Knuth D (1997) Sorting and searching: section 6.2.1: searching an ordered table, The art of computer programming 3, 3rd edn. Addison-Wesley, pp 409–426. ISBN 0-201-89685-0
22. Sheffield Hallam University, Siddeq MM, Rodrigues MA (2016) Image data compression and decompression using minimize size matrix algorithm. WO 2016/135510 A1
23. Jackson AS, Bulat A, Argyriou V, Tzimiropoulos G (2017) Large pose 3D face reconstruction from a single image via direct volumetric CNN regression. Notingham Univeristy. http://cvl-demos.cs.nott.ac.uk/vrn/.

Last access 2018

24. YouTube (2017) Justice League. https://www.youtube.com/watch?v=FrVcEndce9I. Last access Feb 2018
25. YouTube (2009) Samsung Sound HD. https://www.youtube.com/watch?v=UJ1MOWg15Ec. Last accessed Feb 2018

Publisher's note   Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.