A New Representation of Image Through Numbering Pixel Combinations

J. Said[1], R. Souissi[2], H. Hamam[1]
[1]Faculty of Engineering
Moncton, NB
Canada
[2]ISET-Sfax
Tunisia
Habib.Hamam@umoncton.ca

ABSTRACT: *A new way to represent Image consists in numbering all possible pixel combinations. Instead of defining it by a matrix of pixels, the number of the pixel combinations that corresponds to the given image is a unique identifier of this image. To store the image in a memory, we need only to write this number in a file. This new method has several intrinsic advantages such as compression and encryption.*

Keywords: Encryption, Index Image Combinations

Received: 24 November 2012, Revised 30 December 2012, Accepted 13 January 2013

© 2013 DLINE. All rights reserved

1. Introduction

In the last few decades the sharing of information throw internet has reached a large amount, the social network as facebook, twitter, google +…, are the main source of multiple kind of information especially multimedia information as picture and video. Managing a big quantity of information has led to increase bandwidth, storage requirements and even security requirements, these requirements needs a new design for the existing techniques which allow processing and exchanging data quickly and safely. In the last few years some new approaches were developed employing new techniques such as prediction, transform coding, vector quantization, etc., these techniques have to compress data and at the same time represent it efficiently. In this paper we focus especially in the image data representation the proposed approach here is based on the following principles. First how to memorize efficiently [8] the combination of pixels of image and consequently precise a unique descriptor for each image this may have a good consequence for security. Two how to store this descriptor to achieve a better performance for the storage requirements.

Image with size $W \times H$ is defined by a matrix of $W \times H$ pixels. To store an image in a file, the most straightforward way is to write the number of columns and lines of the matrix in the file, then to write all $W \times H$ pixel values in the same file. If the image contains only gray values varying between 0 and 255, each pixel is represented by 8 bits = 1 byte. Let us suppose that the values W and H do not exceed $2^{16} = 65536$. We need 2×2 additional bytes to write the values of W and H. Thus the size of the image will be $(4 + W \times H) \times 8$ bits.

In this paper, we present a new way of defining an image. Instead of explicating the pixels and writing them in the file, we assign a unique number of each combination of pixels. This number is referred to as "*index*" in what follows. Thus an image is nothing but a combination of pixels and this combination is numbered. This number is stored in the file with 4 additional bytes to write the size of the matrix (W and H). For brevity of notation and because it they present a small number, these 4 bytes will be ignored in what follows. It is worth noting that the size of the image is part of the header of any image file format (JPEG, TIFF, …) [1] and is not specific to our method.

2. Proposed Approach

Let us consider the set of Images with 2×2 pixels ($W = 2$ and $H = 2$): Pixel 1 to Pixel 4 as pointed out by Table 1. Each pixel may have a value between 0 and $255 = L - 1$. The number of combinations is $L^{W \times H} = 256^{2 \times 2}$ since we have 4 pixels and each pixel may take 1 of L ($L = 256$) values. Since the total number of combinations is $256^4 = 2^{32}$, then index of any image is between 0 and $2^{32} - 1$. To store this index in a file, we need between 0 and 32 bits, which is the size of the image file according to this new way of representing images.

On the other hand if we have to represent the image with matrix representation each pixel needs 8 bits and the total number of bits needed for the image is $4 \times 8 = 32$ bits. Thus, there is a reduction of the file size compared to the straight forward format (no compression). In this format 32 bits are required for any image with size 2×2 pixels, whereas for our new method 32 bits is the biggest memory size (most unfavorable case) required. Depending on the combination, 1, 2, 3… or 32 bits are required to store the image in a file. This is a first gain.

If the index has a low value, such as 11, it is represented by a small number of bits (4 bits in the case of 11). We can choose the way of numbering the combinations so that to increase the compression rate. The compression rate is defined as the ratio of bits before compression to the bits after compression. Also the way of numbering may be used in protecting data (cryptography, watermarking and steganography) [2-6].

Thus, the compression ratio varies from (last combination):

$$\min = \frac{W \times H \times roundup\,(log_2\,(L))}{roundup\,(W \times H \times log_2\,(L))} : 1$$

Index	Pixel 1	Pixel2	Pixel3	Pixel4
0	0	0	0	0
1	1	0	0	0
2	2	0	0	0
		. . .		
4421768	68	0	68	0
		. . .		
256^4-1	255	255	255	255

Table 1. All possible combinations for Image 2×2 pixels [7]

Index	Pixel 1	Pixel2	Pixel3	Pixel4
0	68	68	0	0
1	68	68	1	0
2	68	68	2	0
		. .		
65025	68	68	255	255
65026	68	0	68	0
		. .		
130051	68	255	68	255
130052	68	0	0	68
		. .		
195077	68	255	255	68
195078	0	68	68	0
		. .		
260103	255	68	68	255
260104	0	68	0	68
		. .		
325129	255	68	255	68
325130	0	0	68	68
		. .		
390150	255	255	68	68

Table 2. All possible combinations for Image 2×2 pixels Having 2 pixels with value 68

to (combinations 0 and 1):

$$\max = W \times H \times roundup\,(log_2(L)) : 1$$

Some proprieties of the image can be used to further compress the image or to introduce new functionalities in data protection. For example, the histogram of a given image shows some properties of this image. The histogram points out the most frequent gray levels in the image. We can choose a way of numbering so that the combinations, including the most frequent gray levels with their respective frequencies, have lower indexes. Let us suppose the gray level g (for example $g = 68$) is repeated G times ($G = 2$) in the images. Table 2 illustrates this case.

The number of possible combinations including $G = 2$ pixels with gray level $g = 68$ among $W \times H= 4$pixels is $C_{W\times H}^{G}(L-1)^{W\times H-G}$ $= C_4^2\, 255^2 = 390150 \sim 2^{19}$. Thus we need 19 bits instead of 32. However, we should add to the header the fact that 68 is repeated 2 times. For this purpose we need $8 + \log_2(W \times H) = 8 + 2$ bits additionally. The gray level ($g = 68$) is expressed by 8 bits and its frequency ($G = 2$) is expressed by a maximum of 2 bits (4 pixels). The result is then 29 bits.

In general the number of bits is:

$$\log_2(C_{W\times H}^{G}(L-1)^{W\times H-G}) + 8 + \log_2(W\times H)$$

Thus the compression ratio is:

$$CR = \frac{W\times H\times 8}{\log_2(C_{W\times H}^{G}(L-1)^{W\times H-G}) + 8 + \log_2(W\times H)}$$

3. Generalizing result with larger Image

In the previous example we applied our method with only one pixel and we got reduction of size about 9.1%, () which corresponds to a compression ratio of 32:26.With larger images with several frequent gray levels, the reduction of size may exceed 50%.

The table below shows the experimental results obtained by the application our Method on the COIL-100* images library, we choose the most frequent 15 values of pixel instead of the most frequent one value of pixel

$$Index = \prod_{i=1}^{15} C^{G_i}_{W\times H-\sum_{k=1}^{i} G_k}(L-i)^{W\times H-\sum_{k=1}^{i} G_k}$$

Image1		Image2		Image3		Image5		Image6	
G i	F i	G i	F i	G i	F i	G i	F i	G i	F i
G1	10307	G 1	1316	G 1	10921	G1	7769	G 1	6950
G2	276	G 2	671	G 2	179	G2	108	G 2	488
G3	78	G 3	614	G 3	169	G3	104	G 3	460
G4	73	G 4	557	G 4	161	G4	104	G 4	406
G5	65	G 5	420	G 5	157	G5	101	G 5	245
G6	63	G 6	286	G 6	135	G6	99	G 6	179
G7	61	G 7	265	G 7	124	G7	99	G 7	177
G8	58	G 8	234	G 8	118	G8	97	G 8	172
G9	51	G 9	218	G 9	102	G9	96	G 9	161
G10	51	G 10	200	G 10	100	G10	96	G 10	133
G11	49	G 11	183	G 11	98	G11	96	G 11	124
G12	46	G 12	164	G 12	85	G12	96	G 12	105
G13	45	G 13	117	G 13	74	G13	95	G 13	84
G14	44	G 14	114	G 14	68	G14	94	G 14	80
G15	43	G 15	106	G 15	66	G15	94	G 15	79

Table 3. The frequency of each Gray Level In all the combinations

BN 1	69918	**BN2**	110159	**BN3**	85889	**BN 4**	93845	**BN 5**	98149
H1	311	**H2**	330	**H3**	322	**H4**	328	**H5**	326

Table 4. The Bits Numbers and HeadersIn all the combinations

Image i	**Reduction Rate i**
Image1	**46,55**
Image2	**15,92**
Image3	**34,39**
Image4	**28,33**
Image5	**25,06**

Table 5. Compression rate forall the combinations

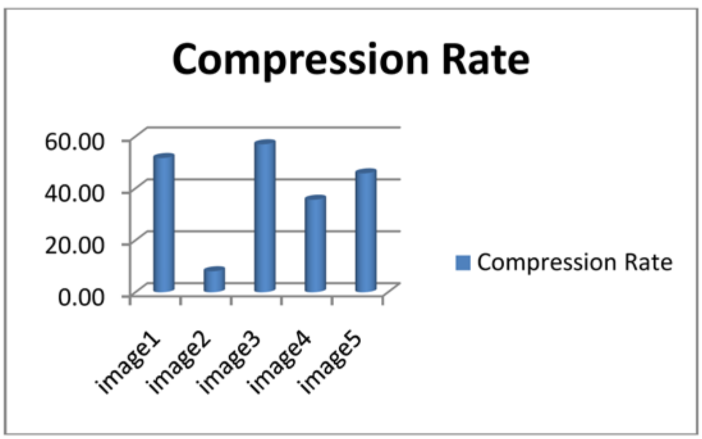

Figure 1. Histogram of the Compression rateof all the combinations

Here some experimental results

* Number of frequent gray levels = 15

* $W \times H = 16384$

* 1 Pixel is represented by 8 bits

* g_i : 1.. 15 : Gray 1 .. 15

* f_i : Frequency 1 .. 15

* H_i : 1..5 : Header 1.. 5

* CR_i: 1.. 5 : Compression ratio 1.. 5

* BN_i : 1.. 5 : Bits Number 1.. 5

and

$$CR_i = \frac{W \times H \times 8}{\log_2\left(\prod_{i=1}^{15} C^{G_i}_{W \times H - \sum_{k=1}^{i} G_k} (L-i)^{W \times H - \sum_{k=1}^{i} G_k}\right) + \sum_{i=1}^{15} 8 + \log_2(max\,(G_i))}$$

The figure below shows the variation of the compression ratio based on dimension of the image (L: 128-> 1024 and H: 128-> 1024).

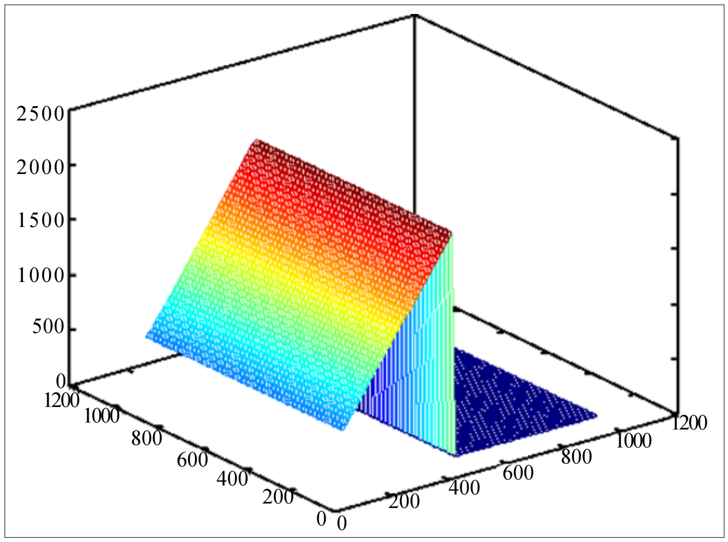

Figure 2. The variation of Compression Ratio for Large Image

4. Conclusion

We presented a new way to represent images. It allows us to compress information and apply data protection namely cryptography, watermarking and steganography. We can use the histogram of the image to further compress the image.

References

[1] Miano, J. (1999). Compressed Image File Formats: JPEG, PNG, GIF, XBM, BMP. Adison-Weley | 1999 | ISBN: 0201604434

[2] Johnson, N. F., Katzenbeisser, S. (2000). Information Hiding, ch. A survey of steganographic techniques. Artech House, Norwood, MA.

[3] Lee, Y., Chen, L. (2000). High capacity image steganographic model, *IEEE Proceedings on Vision, Image and Signal Processing*, 147, 288 -294.

[4] Morimoto, N. (1999). Digital watermarking technology with practical applications, *In*: Informing Science Special Issue on Multimedia Informing Technologies, 2, 107-111.

[5] Petitcolas, F. (2000). Watermarking schemes evaluation, *IEEE Signal Processing Magazine*, 17, 58-64.

[6] Ahmed, M., Kiah, M., Zaidan, B., Zaidan, A. (2010). A novel embedding method to increase capacity and robustness of low-bit encoding audio steganography technique using noise gate software logic algorithm, *Journal of Applied Sciences*, 10, 59-64.

[7] Hamam, H. (2013). A new representation of data through combinations, ICCAT 2013

[8] Berg, A. P., Mikhael, W. B. (1997). An efficient structure and algorithm for image representati on using non-orthogonal basis images, *IEEE Trans. Circuits and Systems Part* II*: Analog and Digital Signal Processing*, 44 (10) 818–828, Oct.