An Analysis of Variation in Lossless Image Compression using FMM & Threshold Value 10

Dr. Sandeep Mathur[1]
Department of Mathematics
Jodhpur Institute of Engineering & Technology
Jodhpur, India

Dr. Anjali Mathur[2]
Department of Mathematics
Jodhpur Institute of Engineering & Technology
Jodhpur, India

Nitesh Agarwal[3]
Department of Computer Science
Jodhpur Institute of Engineering & Technology
Jodhpur, India

***Abstract*— A digital image store its color information in digits format in digital devices. This information store in pixel matrix. Image compression process reduce required storage size of image w. r. t to digital device or communication system. Image compression process use two technique to compress image lossless image compression & lossy image compression. Images that provide numerical, secure & financial information compressed using lossless image compression because we required original data back after decompression process. Lossless image compression uses some entropy encoding techniques like RLE, Huffman encoding, LZW encoding etc. Present papers deals with lossless image compression using RLE as entropy encoding, & compare this lossless image compression with some modification by FMM (Five Module Method) & by Threshold value 10. RLE give best compression ratio when image pixel matrix has repeated sequence of pixels. To make repeated sequence in pixel matrix in this paper two method used FMM & TH=10. These method modify pixel original matrix & make repeated sequence in this matrix before RLE to get a good compression ratio**.

Key Words*: *FMM, RLE, MSE, PSNR, TH=10.

1. INTRODUCTION

There are numerous applications of image processing, such as satellite imaging, medical imaging and video where the image size or image stream size is too large and require a large amount of storage space or high bandwidth for communication in its original form. Every storage device & communication bandwidth cannot satisfy this requirement hence image compression techniques are used in such type of applications where image size is too large to store in digital device & too large for communication purpose. Image compression plays a very important role in application like tele-videoconferencing, remote sensing, document & medical imaging and facsimile transmission, which depends on the efficient manipulation, storage & transmission of binary, gray scale or color images.

Image compression techniques can be classified into two categories lossless image compression & lossy image compression. Images that provide numerical, Secure & financial information compressed using lossless image compression because we required original data back after decompression process. But other images like multimedia images can be compressed using lossy image compression because the human eye is very tolerant of approximation error in an image. Hence we may decide to exploit this tolerance to produce increased compression, at the expense of image quality by reducing some pixel data or information. Lossless image compression use some entropy encoding techniques like Run Length Encoding (RLE), Huffman Encoding, LZW (Lempel Ziv Welch) Encoding, and Area Encoding. This paper deals with RLE as a entropy encoding in lossless image compression. RLE entropy encoding give good compression ratio when image have repeated pixel value sequentially but all the image not have such type of repeated pattern hence present paper use FMM (Five Module Method) & Threshold value before RLE to make repeated sequence.

1.1 Lossless Image Compression

In lossless compression, every single bit of data that was originally in the file remains after the file is uncompressed. All of the information is completely restored. This is generally the technique of choice for text or spreadsheet files, where losing words or financial data could pose a problem. Process of lossless image compression is shown in fig 1 [11].

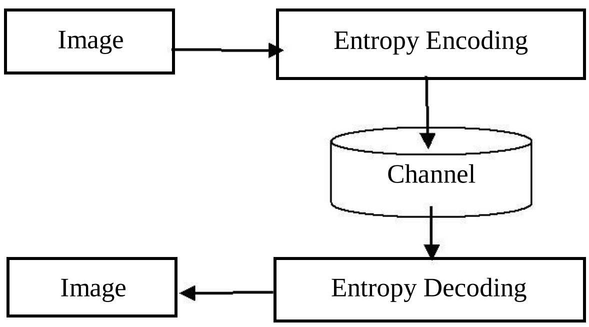


Fig 1: Lossless Image Compression Process

1.2 Five Module Method (FMM)

In most of images, there is a common feature which is the neighboring pixels are correlated. Therefore, finding a less correlated representation of image is one of the most important tasks. One of the basic concepts in compression is the reduction of redundancy and Irrelevancy. This can be done by removing duplication from the image. Sometime, Human Visual System (HVS) cannot notice some parts of the signal, i.e. omitting these parts will not be noticed by the receiver. This is called as Irrelevancy. FMM read each pixel value row by row & divide each pixel value by 5 & add or subtract the reminder from original pixel to get repeated pixel values. The basic idea in FMM is to check the whole pixels metrics and transform each pixel into a number divisible by 5 according to the following conditions.

```
if A(i,j) Mod 5 = 4
      A(i,j)=A(i,j)+1
else if A(i,j) Mod 5 = 3
             A(i,j)=A(i,j)+2
       else if A(i,j) Mod 5 = 2
                    A(i,j)=A(i,j)-2
              else if A(i,j) Mod 5 = 1
                           A(i,j)=A(i,j)-1
```

For ex.

121	122	122	123	124	125	105	110
130	132	132	131	134	135	133	220
221	222	222	223	224	225	205	300
425	426	427	500	501	502	501	905
521	522	522	523	524	525	555	660
630	632	632	631	634	635	633	633
851	852	852	963	964	965	205	300
425	426	427	500	501	502	501	905

Table 1: Input Pixel matrix

Table 1 shows a 2D pixel matrix but this matrix cannot be compressed using RLE because pixel value not repeated sequentially. FMM method convert this matrix so that it can be compressed using RLE

120	120	120	125	125	125	105	110
130	130	130	130	135	135	135	220
220	220	220	225	225	225	205	300
425	425	425	500	500	500	500	905
520	520	520	525	525	525	555	660
630	630	630	630	635	635	635	635
850	850	850	965	965	965	205	300
425	425	425	500	500	500	500	905

Table 2: Transformed Pixel matrix

After the FMM pixel matrix contain a good no of repeated pixel as shown in table 2. This repeated pixel helps the RLE to compress pixel matrix.

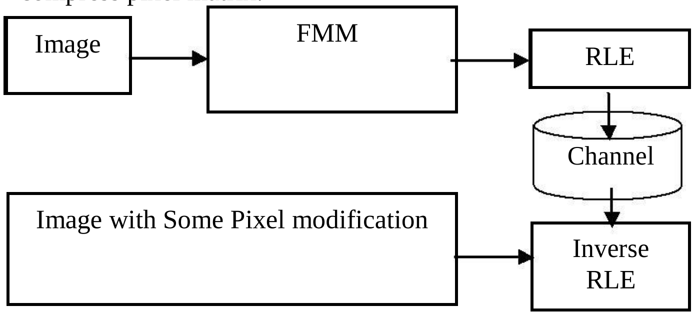


Fig 2: Image Compression Using

FMM FMM algorithm

```
Input: Pixel matrix of input image.
Output: Transformed Pixel Matrix.
{ w = width of pixel matrix; h
  = height of pixel matrix;
  pixel[h][w] = pixel matrix of original
  image; for(i=0; i<h; i++)
 { for(j=0; j<w; j++)
  {  if pixel(i,j) Mod 5 = 4
         pixel(i,j)=pixel(i,j)+1
     else if pixel(i,j) Mod 5 = 3
            pixel(i,j)=pixel(i,j)+2
          else if pixel(i,j) Mod 5 = 2
                 pixel(i,j)=pixel(i,j)-2
               else if pixel(i,j) Mod 5 = 1
                      pixel(i,j)=pixel(i,j)-1
  }
 }

}[5].
```

1.3 TH (Threshold) value Method

TH value method take pixel value from 2D pixel matrix row by row. TH value method uses two node 1st node store the 1st pixel & 2nd node move forward row by row in pixel matrix. TH value method take pixel difference between pixel value store in these two node & repeat the pixel value store in node 1 until difference between these two nodes are not greater than threshold value 10. Once a difference greater than 10 occurs node 1 store that pixel & node 2 traverse pixel one by one from neighboring pixel of pixel store in node 1 & same process is follow until complete pixel matrix not traversed. For example –

201	200	205	210	301	300	305	310
406	404	407	406	506	504	507	506
602	601	600	602	702	701	700	702
828	829	830	832	928	929	930	932
301	305	302	303	105	104	100	102
655	654	653	650	651	652	657	658
702	705	704	702	701	801	805	809
105	105	105	105	105	106	112	111

Table 3: Pixel matrix of input image

Table 3 metrics cannot be compressed by RLE

201	201	201	201	301	301	301	301
406	406	406	406	506	506	506	506
602	602	602	602	702	702	702	702
828	828	828	828	928	928	928	928
301	301	301	301	105	105	105	105
655	655	655	655	655	655	655	655
702	702	702	702	702	801	801	801
105	105	105	105	105	105	112	112

Table 4: Pixel matrix of After TH value = 10

After the TH value method pixel matrix contain a good no of repeated pixel as shown in table 2. This repeated pixel helps the RLE to compress pixel matrix.

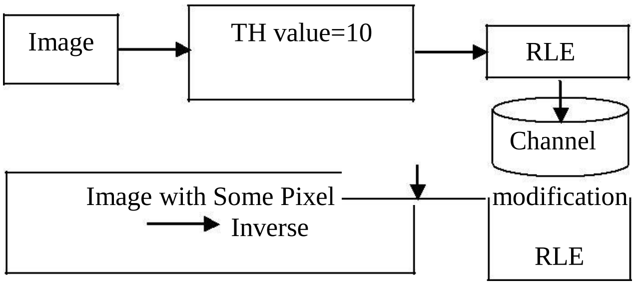


Fig 3: Image Compression Using TH value=10

TH value = 10 algorithm

Input: Pixel matrix of input image.
Output: Modified Pixel Matrix.

```
{ w = width of pixel matrix;
  h = height of pixel matrix;
  pixel[h][w] = pixel matrix of original
  image; for(i=0; i<h; i++)
 { j=0;
   tmp=pixel[i][j];
   for(j=0; j<w; j++)
  { if( (difference between tmp & pixel[i][j]) > 10 )
     pixel[i][j] = tmp;
     else
    tmp=pixel[i][j];
 }}} [1].
```

THV method used in such type of image where modifying some pixel data does not cause any big problem.

1.1.2 RLE (Run Length Encoding)

This is a very simple compression technique method used for compressing sequential data. Many digital image consist pixel values that are repeats sequentially for such type of image RLE is useful. In TH value method, & in FMM RLE receive sequential data from pixel matrix modified by TH value method & FMM, & store pixel value that repeats & no of time that pixel value repeat sequentially. For example table 2 by RLE compressed as

Pixel Value	Repetition	Pixel Value	Repetition	Pixel Value	Repetition
120	3	205	1	630	4
125	3	300	1	635	4
105	1	425	3	850	3
110	1	500	4	965	3
130	4	905	1	205	1
135	3	520	3	300	1
220	1	525	3	425	3
220	3	555	1	500	4
225	3	660	1	905	1

Table 5: Compressed Data after RLE for Table 2

Table 5 required less storage space as compare to table 1. Table 1 require total 64 values to store but table 3 require only 54 values to store [5].

Compression Ratio (CR) = 64/54 = 1.19

Table 4 by RLE compressed as

Pixel Value	Repetition	Pixel Value	Repetition
201	4	301	4
301	4	105	4
406	4	655	8
506	4	702	5
602	4	801	3
702	4	105	6
828	4	112	2
928	4		

Table 6: Compressed Data after RLE for Table 4

Table 6 required less storage space as compare to table 3. Table 3 require total 64 values to store but table 6 require only 30 values to store [5].

Compression Ratio (CR) = 64/30 = 2.13

2. Main Results & Outputs

2.1 Implementation of lossless Image Compression using RLE

Steps involved in this implementation

1. Create pixel matrix of the image.
2. Use RLE as entropy encoding on pixel matrix
3. Store matrix obtain by RLE method in to secondary storage.
4. To get required image read encoded matrix from secondary storage & apply entropy decoding (Run Length Decoding) on that encoded matrix.
5. Using this decoded matrix make pixel matrix & then using this pixel matrix to obtain required image.
6. Now find Compression Ratio by following formula

$$CR = \frac{Original\ \ Image\ \ size}{Output\ \ Image\ \ size} \quad (1)$$

2.2 Implementation of Image Compression using FMM & RLE

Steps involved in this implementation

1. Create pixel matrix of the image.
2. Apply FMM method on pixel matrix & apply FMM algorithm.
3. Use RLE as entropy encoding on pixel matrix obtain from FMM algorithm.
4. Store matrix obtain by RLE method in to secondary storage.
5. To get required image read encoded matrix from secondary storage & apply entropy decoding (Run Length Decoding) on that encoded matrix.
6. Using this decoded matrix make pixel matrix & then using this pixel matrix make required image.
7. Now we Find MSE (Mean Squared Error), PSNR (Peak Signal To Noise Ratio) & CR (Compression Ration) to determine quality of image obtain by proposed method [5] -

$$MSE = \frac{1}{H * W} \sum_{x=0}^{H-1} \sum_{y=0}^{W-1} [O(x, y) - M(x, y)]^2 \quad (2)$$

$$PSNR = 20 * \log_{10}(MAX) - 10 * \log_{10}(MSE) \quad (3)$$

CR can be calculated using eq. (1).

Where H=Height of Image, W= Width of Image, variable MAX shows max value of a pixel for example here image is 8 bit hence MAX=255,

2.3 Implementation of Image Compression using TH value = 10 & RLE

Steps involved in this implementation

1. Create pixel matrix of the image.
2. Apply TH value =10 method on pixel matrix & apply TH value = 10 algorithm.
3. Use RLE as entropy encoding on pixel matrix obtain from TH value = 10 algorithm.
4. Store matrix obtain by RLE method in to secondary storage.
5. To get required image read encoded matrix from secondary storage & apply entropy decoding (Run Length Decoding) on that encoded matrix.
6. Using this decoded matrix make pixel matrix & then using this pixel matrix make required image.
7. Now we Find MSE (Mean Squared Error), PSNR (Peak Signal To Noise Ratio) & CR (Compression Ration) to determine quality of image obtain by proposed method by eq. (2), (3) & (1) respectively.

2.4 Outputs

2.4.1 Lossless image compression with RLE only

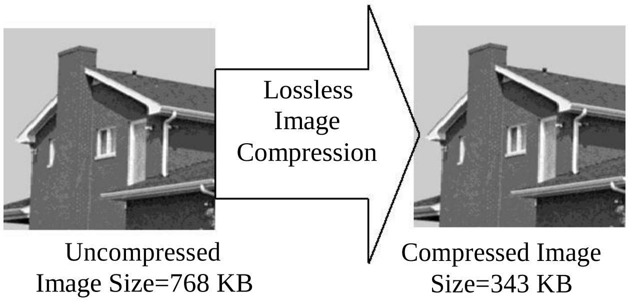


Fig 4: Lossless Image compression using RLE

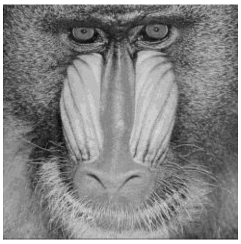
Input image	Compressed size	CR
 Size=768 KB Lena.bmp	552	7.46
 Size=768 KB Baboon.bmp	624	3.44
 Size=768 KB Zelda.bmp	504	7.46
 Size=768 KB House.bmp	344	10.68
 Size=768 KB Pappers_grey.bmp	424	7.46

Table 7: Compression Ratio

2.4.2 Image Compression Using FMM & RLE

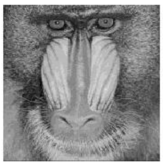
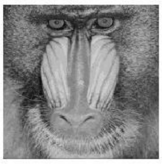
Input image	Compressed image	MSE	PSNR	CR
 Size=768 KB Lena.bmp	 Size=296 KB	5.99	40.36	2.59
 Size=768 KB Baboon.bmp	 Size=432 KB	5.99	40.36	1.78
 Size=768 KB Zelda.bmp	 Size=280 KB	5.99	40.36	2.74
 Size=768 KB House.bmp	 Size=176 KB	4.49	41.60	4.36
 Size=768 KB Pappers_grey. bmp	 Size=296 KB	5.83	40.47	2.59

Table 8: MSE, PSNR & CR value of image after FMM & RLE

2.4.3 Image Compression Using TH value=10 & RLE

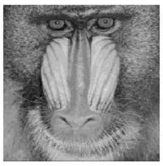
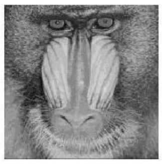
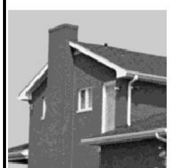
Input image	Compressed image	MSE	PSNR	CR
 Size=768 KB Lena.bmp	 Size=87.9 KB	20.95	34.92	8.73
 Size=768 KB Baboon.bmp	 Size=319 KB	15.31	36.28	2.41
 Size=768 KB Zelda.bmp	 Size=71.9 KB	22.61	34.56	10.68
 Size=768 KB House.bmp	 Size=63.9 KB	16.56	35.94	12.01
 Size=768 KB Pappers_grey. bmp	 Size=135 KB	21.05	34.90	5.69

Table 9: MSE, PSNR & CR value of image after TH value = 10 & RLE

3. CONCLUSION

The result presented in this document shows that

1. The results shows that RLE gives compressed image without any pixel loss but compression ratio of RLE is not good w.r.t FMM & TH value = 10 .

2. FMM give good compression ratio than RLE but its compression ratio not god w.r.t. TH value = 10.

3. By comparing Table 7, Table 8 & Table 9 it is clear TH value = 10 gives best compression ratio.

4. By comparing Table 7, Table 8 & Table 9 it is clear FMM gives best quality of compressed image because it gives least MSE & high PSNR but CR value less than TH value=10 method. Hence with respect to quality order of best method is RLE > FMM > TH value = 10 & with respect to CR the order of best method is TH value = 10 > FMM > RLE.

5. Both method can be used in lossy image compression before entropy encoding technique.

REFERENCES

[1] A. H. Husseen , S. Sh. Mahmud & R. J. Mohammed " Image Compression Using Proposed Enhanced Run Length Encoding Algorithm ", Ibn Al-Haitham Journal For Pure And Applied Science, VOL 24(1), pp. 315-328, 2011.

[2] A. M. Eskicioglu, and P. S. Fisher, "Image quality measures and their performance," IEEE Trans. Commun., vol. 43, no. 12, pp. 2959-2965, Dec. 1995.

[3] Asha Lata & Permender singh "Review of Image Compression Techniques", International Journal of Emerging Technology & Advanced Engineering (IJETAE), vol 3, issue 7, pp. 461-464, July 2013.

[4] Debashish Chakroborty, Amiya Halder "An Efficient Lossless Image Compression Using Special Character Replacement", IEEE ICCET-2010, 13-14 November pp E-62 – E-67, Jodhpur, Rajasthan, India.

[5] Firas A. Jassim and Hind E. Qassim, " FIVE MODULUS METHOD FOR IMAGE COMPRESSION", signal & image processing: An International Journal (SIPIJ), vol 3, no.5, pp. 19-28, Octobar 2012.

[6] Gabriela Dudek , Przemysław Borys , Zbigniew J. Grzywna, Lossy dictionary-based image compression method, Image and Vision Computing, v.25 n.6, p.883-889, June, 2007.

[7] Harley R. Myler and Arthur R. Weeks "The Pocket Handbook of Image Processing Algorithms in C", ISBN 0-13-642240-3 Prentice Hall P T R Englewood Cliffs, New Jercy 07632.

[8] Iain E.G. Richardson "H.264 and MPEG-4 Video Compression: Video Coding for Next-generation Multimedia", ISBN 0470848375, 9780470848371, Wiley, 2003.

[9] Jesse D. Kornblum "Using JPEG quantization tables to identify imagery processed by software", ELSEVIER, DIGITAL INVESTIGATION 5, pp. S21-S25,2008.

[10] Subramanya A, "Image Compression Technique," Potentials IEEE, Vol. 20, Issue 1 pp 19-23, Feb-March 2001.

[11] V.Singh "Recent Patents on Image Compression – A Survey", Recent Patents on signal Processing (Bentham Open), vol 2, pp. 47-62, 2010.

.