ISSN: 2278-0181
NCETRASECT-2015 Conference Proceedings

Image Compression using THV method in Lossless Image Compression & in Lossy Image Compression

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

Nitesh Agarwal[2]
Department of Computer Science
Jodhpur Institute of Engineering & Technology
Jodhpur, India

Dr. Sandeep Mathur[3]
Department of Mathematics
Jodhpur Institute of Engineering & Technology
Jodhpur, India

***Abstract*—Image compression process reduce required storage size of image. 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 use some entropy encoding technique but its compression ratio is low w.r.t lossy image compression. 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. Using this concept this paper proposed a THV (Threshold Variable) method before entropy encoding technique to get more compressed size using lossless image compression as well as using lossy image compression & give a comparative study in which technique, proposed method is more useful. This paper deals with comparative study of a compressed image on the basis of different value of threshold variable used in THV method.**

Key Words: THV, DCT, RLE, MSE, PSNR.

1. INTRODUCTION

A digital image is a 2D pixel matrix where each position of pixel gives a color information for image in bits format. On the basis of this bits format image is classified as 2 bit, 6 bit, 8 bit, 16 bit, 24 bit & 32 bit. When an image is design using one of these format each pixel store information in particular bit format in which they are build. To store image in only bits format some time required a high amount of storage device for example if an image is build using 8 bit format then each pixel must have 8 bit storage even it represent information that required less than 8 bit to store. As the digital devices has limited storage & transmission capability we need to compress the image by some suitable method to satisfy this limitation. For example when we play a YouTube video if our communication system have enough bandwidth to play YouTube high resolution video then we can play video without any buffering problem but if bandwidth is low then YouTube use some compression method & transfer the bits of video frame in compressed format according to our network bandwidth size. By lossless image compression we get original image in decompression process without any loss of pixel value. 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.

1.1 Lossless Image Compression

Lossless image compression process compress the image for storage & communication purpose in such a way that original image can be retrieve during decompression process without any loss or modification of information (pixel bits). Lossless image compression use some entropy encoding techniques to compress digital image such as RLE (Run Length Encoding), Huffman Encoding, LZW (Lempel-Ziv-Welch) Encoding, Area Encoding etc. Process of lossless image compression is shown in fig 1 [14].

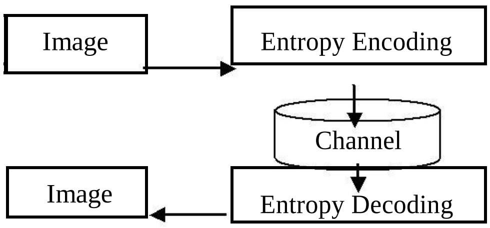


Fig 1 Lossless Image Compression

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 introduce a THV module before encoding technique to make a repeated sequence [5].

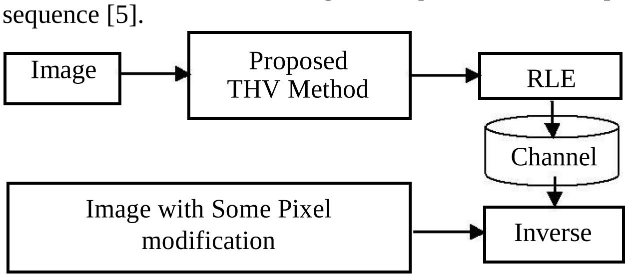


Fig 2 Proposed Lossless Image Compression

In proposed method as shown in fig 2 image in pixel format transfer to the THV module & then send to the entropy where using RLE proposed method compress input image. THV module modify pixel values of image to get repeated sequence hence output image after the decoding process is equivalent to original image but not equal to the original image data.

1.1.1 THV (Threshold Variable) Method

THV method take pixel value from 2D pixel matrix row by row. THV method uses two node 1st node store the 1st pixel & 2nd node move forward row by row in pixel matrix. THV 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 variable t used in THV method. One a difference greater than t 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 –

101	100	105	110	201	200	205	210
306	304	307	306	406	404	407	406
502	501	500	502	602	601	600	602
728	729	730	732	828	829	830	832
201	205	202	203	905	904	900	902
555	554	553	550	551	552	557	558
602	605	604	602	601	701	705	709
905	905	905	905	905	906	912	911

Table 1: Pixel matrix of input image

Table 1 shows a 2D pixel matrix but this matrix cannot be compressed using RLE because pixel value not repeated sequentially. THV method convert this matrix so that it can be compressed using RLE. Let the value of variable t in THV method is 10 then converted pixel matrix is

101	101	101	101	201	201	201	201
306	306	306	306	406	406	406	406
502	502	502	502	602	602	602	602
728	728	728	728	828	828	828	828
201	201	201	201	905	905	905	905
555	555	555	555	555	555	555	555
602	602	602	602	602	701	701	701
905	905	905	905	905	905	912	912

Table 2: Pixel matrix of After THV method with variable t =10

After the THV 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.

THV 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; t = THV variable;
 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]) > t
      ) 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 proposed THV method RLE receive sequential data from pixel matrix modified by THV method & 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
101	4	201	4
201	4	905	4
306	4	555	8
406	4	602	5
502	4	701	3
602	4	905	6
728	4	912	2
828	4		

Table 3: Compressed Data after RLE

Table 3 required less storage space as compare to table 1. Table 1 require total 64 values to store but table 3 require only 30 values to store [5].

Compression Ratio (CR) = 64/30 =

2.13 1.2 Lossy Image compression

Lossy image compression gives a high compression ration then lossless image compression. It is say as lossy because it modify or destroy some pixel information by using some transformation like DCT, DST, KLT, DFT etc. & using quantization table before entropy encoding.

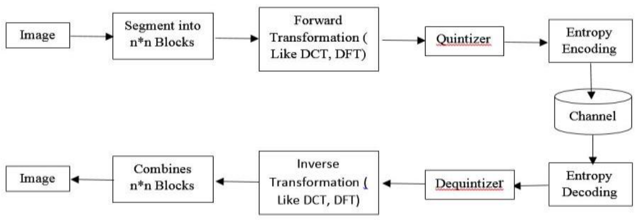


Fig 3: Lossy Image Compression

In this paper we use DCT (Discrete Cosine Transformation) to transform image in frequency domain [4].

1.2.1 DCT (Discrete Cosine Transform) [11]

DCT convert an image into its equivalent frequency domain by partitioning image pixel matrix into blocks of size N*N. An image is a 2D pixel matrix hence 2D DCT is used to transform an image.

2-D DCT can be defined as

$$C(u,v)=\alpha(u)\alpha(v)\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)\cos\left[\frac{\pi(2x+1)u}{2N}\right]\cos\left[\frac{\pi(2y+1)v}{2N}\right] \quad (1)$$

for $u, v = 0,1,2,\ldots,N-1$.

& inverse transformation is defined as

$$f(x,y)=\sum_{u=0}^{N-1}\sum_{v=0}^{N-1} \alpha(u)\alpha(v)c(u,v)\cos\left[\frac{\pi(2x+1)u}{2N}\right]\cos\left[\frac{\pi(2y+1)v}{2N}\right] \quad (2)$$

Where $C(u,v)$ represents frequency value for u, v & $f(x,y)$ represents pixel color value at position (x, y).

$$\alpha(u)=\begin{cases}\sqrt{\frac{1}{N}} & for\ u=0\\ \sqrt{\frac{2}{N}} & for\ u\neq 0\end{cases} \quad (3)$$

$$\alpha(v)=\begin{cases}\sqrt{\frac{1}{N}} & for\ v=0\\ \sqrt{\frac{2}{N}} & for\ v\neq 0\end{cases} \quad (4)$$

1.2.2 Quantization

A Quantizer simply reduces the number of bits needed to store the transformed coefficients by reducing the precision of those values. Since this is a many-to-one mapping, it is a lossy process and is the main source of compression in an encoder.

The quantization matrix is designed to provide more resolution to more perceivable frequency components over less perceivable components (usually lower frequencies over high frequencies) in addition to transforming as many components to 0, which can be encoded with greatest efficiency. A DCT block is quantize using following formula

$$QDCT(i,j)=ROUND\left(\frac{DCT(i,j)}{QT(i,j)}\right) \quad (5)$$

& this QDCT block dequantize by following formula

$$DCT(i,j)=ROUND\left(QDCT(i,j)*QT(i,j)\right) \quad (6)$$

For i, j= 0, 1, 2, 3…….,N-1

Where (i,t) define position of input & output value, QDCT is DCT block after quantization, QT is standard quantization matrices & defined as

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	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

Table 4: Standard Quantization Table [9]

1.2.3 Proposed THV method in Lossy Image Compression

Proposed THV method can be used in Lossy Image compression before entropy encoding as

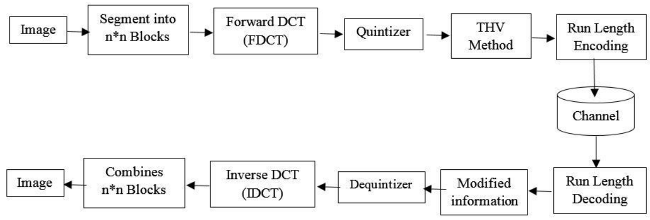


Fig 4: Lossy Image Compression with THV Method

The THV algorithm is work in same way as in lossless image compression the only difference is that in lossless image compression it take input from image pixel matrix but in lossy image compression it take input from quantize block & loop in THV depend upon dimension of quantize block not on dimension of image.

2. MAIN RESULTS & OUTPUTS

2.1 Implementation of Proposed THV method in lossless Image Compression

Steps involved in this implementation

1. Create pixel matrix of the image.
2. Apply THV method on pixel matrix & apply THV algorithm.
3. Use RLE as entropy encoding on pixel matrix obtain from THV 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

ISSN: 2278-0181
NCETRASECT-2015 Conference Proceedings

proposed method for each t variable used in THV method. MSE_X, $PSNR_X$ & CR_X calculated by following formulas [12] -

$$MSE_t = \frac{1}{H*W}\sum_{x=0}^{H-1}\sum_{y=0}^{W-1}[O(x,y) - M_t(x,y)]^2 \qquad (7)$$

$$PSNR_t = 20*\log_{10}(MAX) - 10*\log_{10}(MSE_t) \qquad (8)$$

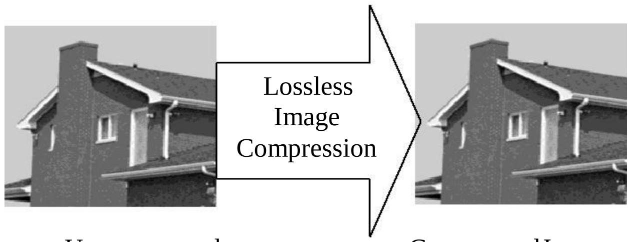
$$CR_t = \frac{Original\ \ Image\ \ size}{Output\ \ Image\ \ size} \qquad (9)$$

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, MSE_t, $PSNR_t$ & CR_t is MSE, PSNR & CR at variable t used in THV method.

Quality of image obtain by proposed method is depend on MSE_t & $PSNR_t$ value. If as the MSE value increases PSNR value decreases then we get a bad quality of image by proposed method & if as the MSE value decreases PSNR value increases we get a batter quality image hence on basis of this MSE_t & $PSNR_t$ value proposed method gives a best value of X on which we get a high compressed image with best quality.

2.1.1 Outputs

2.1.1.1 Lossless image compression without THV method



Uncompressed Image Size=768 KB

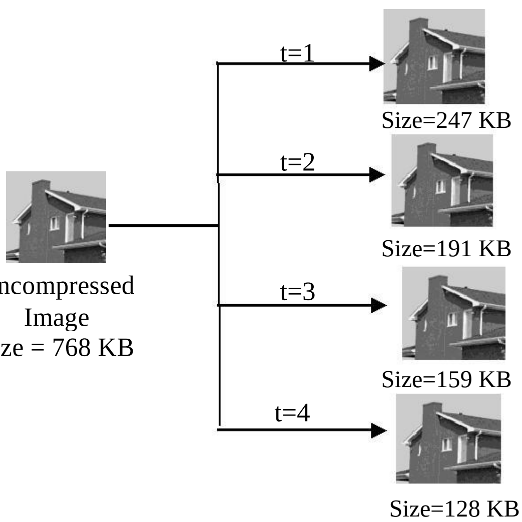
Compressed Image Size=343KB

Fig 5: Lossless Image Compression without THV Method

	MSE	PSNR	CR
Lossless Image Compression without THV Method	0	Infinity	2.24

Table 5: MSE, PSNR, CR without THV method

2.1.1.2 Lossless Image Compression with proposed THV method



Compressed Images

Fig 6: Proposed Lossless Image Compression with variation in THV Variable t

t	MSE_t	$PSNR_t$	Compressed Image Size	CR_t
1	0.18	55.50	247	3.11
2	1.73	45.75	191	4.02
3	2.58	44.02	159	4.83
4	3.71	42.43	128	6
5	5.12	40.9	111	6.92
6	6.92	39.72	95.9	8
7	8.96	38.61	87.9	8.74
8	11.25	37.62	79.9	9.61
9	13.73	36.75	71.9	10.68
10	16.56	36.94	63.9	12.02
11	19.33	35.27	63.9	12.02
12	22.92	34.53	55.9	13.74
13	26.16	33.95	48	16
14	29.96	33.66	48	16
15	33.76	32.85	48	16
16	38.58	32.27	39.9	19.24

Table 6: MSEt, PSNRt, CRt on different value of t

2.1.1.2.1 Graphs

1) THV variable t vs. CR_t

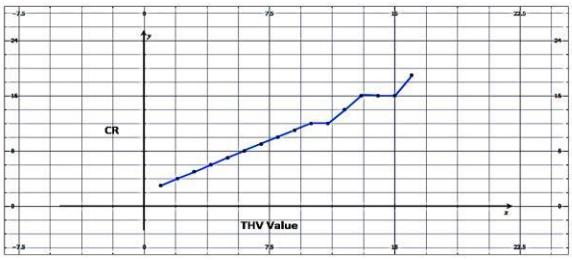


Fig 7: Variation in CRt with different value of THV variable t

2) THV variable t vs. MSE_t

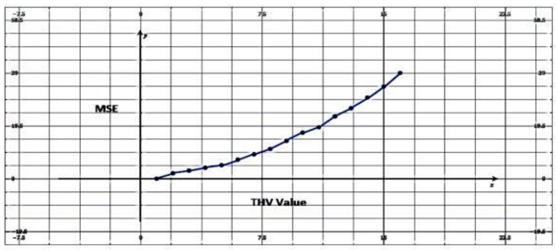


Fig 8: Variation in MSEt with different value of RVM variable t

3) THV variable t vs. $PSNR_t$

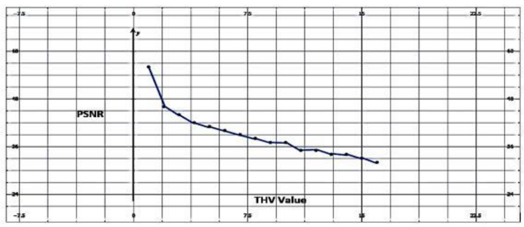


Fig 9: Variation in PSNRt with different value of THV variable t

4) THV variable t vs. $PSNR_t$ & MSE_t

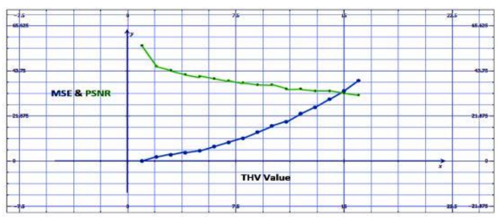


Fig 10: Variation in PSNRt & MSEt with different value of THV variable t

2.2 Implementation of Proposed THV method in lossy Image Compression

Steps involved in this implementation

1. Create pixel matrix of the image & divided it into blocks of size 8*8
2. Apply FDCT (Forward Discrete Sine Transform) on each 8*8 block of pixel matrix to get equivalent 8*8 DCT blocks using eq (1).
3. Apply eq (5) on each block of DCT to get QDCT block.
4. Apply THV algorithm on each block of QDCT to get THV block.
5. Combine each THV block & apply RLE on combine block & store this encoded block on secondary storage.
6. To get required image read encoded matrix from secondary storage & apply entropy decoding (Run Length Decoding) on that encoded matrix.
7. Divide this decoded matrix in to blocks of size 8*8.
8. Apply eq (6) on each block to get DCT blocks.
9. Apply eq (2) on each DCT block to get IDCT blocks.
10. Combine all IDCT blocks to get pixel matrix.
11. Using pixel matrix we get required image.

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 for each t variable used in THV algorithm.

2.2.1 Outputs

2.2.1.1 Lossy Image compression without THV method

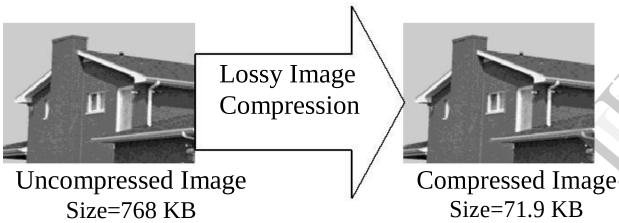


Fig 11: Lossy Image Compression without THV Method

	MSE	PSNR	CR
Lossy Image Compression without THV Method	7.37	39.45	10.68

Table 7: MSE, PSNR, CR without THV

2.2.1.2 Lossy Image Compression with proposed THV method

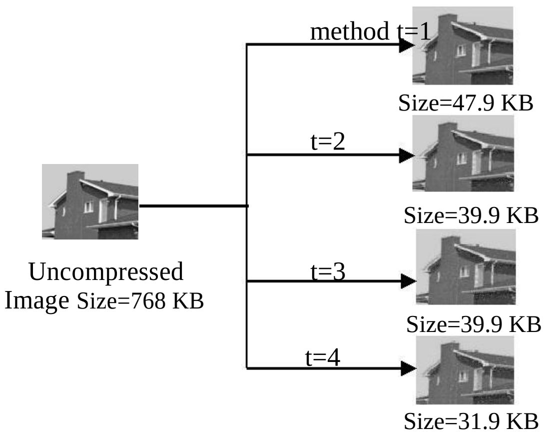


Fig 12: Lossy Image Compression with THV Method

t	MSE_t	$PSNR_t$	Compressed Image Size	CR_t
1	502.13	21.12	47.9	16
2	762.40	19.30	39.9	19.24
3	982.66	18.20	39.9	19.24
4	1141.13	17.56	31.9	24.08
5	1262.50	17.12	31.9	24.08
6	1338.30	16.87	31.9	24.08
7	1416.67	16.61	32	24
8	1470.40	16.46	32	24
9	1533.23	16.27	31.9	24
10	1576.20	16.15	31.9	24
11	1615.81	16.04	32	24
12	1659.64	15.93	32	24
13	1680.84	15.88	32	24
14	1703.88	15.82	32	24
15	1741.97	15.72	31.9	24
16	1832.85	15.50	31.9	24

Table 8: MSEt, PSNRt, CRt on different value of t

2.2.1.2.1 Graphs

1) THV variable t vs. CR_t

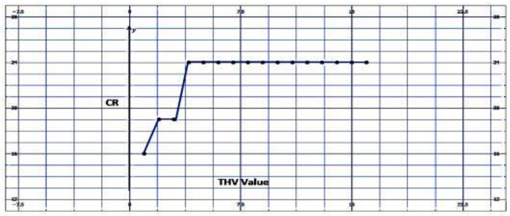


Fig 13: Variation in CRt with different value of THV variable t

2) THV variable t vs. MSE_t

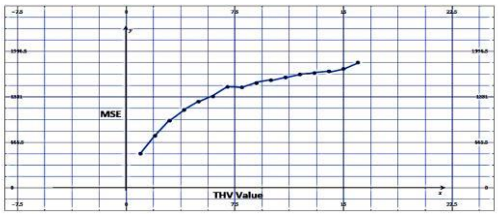


Fig 14: Variation in MSEt with different value of RVM variable t

3) THV variable t vs. $PSNR_t$

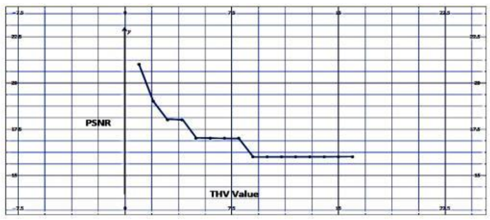


Fig 15: Variation in PSNRt with different value of THV variable t

4) THV variable t vs. $PSNR_t$ & MSE_t

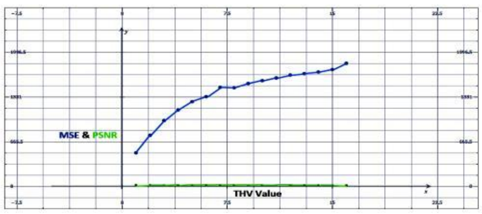


Fig 16: Variation in PSNRt & MSEt with different value of THV variable t

3. CONCLUSION

The result presented in this document shows that

1. The results shows that as the value of variable t increases storage size of image decreases as shown in Fig 6, Fig 12, Table 6 & Table 8.
2. As the value of t increases CR_t also increases as in lossless image compression as shown in Fig 7 but in lossy image compression as the value of t increases CR_t goes to a constant value.
3. As the value of t increases proposed process add more noises in the image i.e. value of MSE_t increases as shown in Fig 8 & Fig 14.
4. The results shows that proposed method give good results in lossless image compression but in lossy image compression present method not works well because it add high amount of noises in image that cannot be tolerable this can be seen by comparing Table 6 & Table 8.
5. As the value of t increases $PSNR_t$ value decreases as shown in Fig 9 & Fig 15.
6. As the MSE_t value decreases & $PSNR_t$ increases quality of image improves but CR_t decreases.
7. Fig 13 show after value 3 of variable t CR_t almost constant in lossy image compression.
8. Image used in this paper for lossless image compression have almost same MSE_t & $PSNR_t$ value for t=15 as shown in Fig 10 .
9. Lossy image compression with THV method give more compression ratio than lossless image compression with THV method but lossless image compression with THV method give good results because its MSE_t values are less than MSE_t values in lossy image compression with THV for same value of t.

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. Raid, W.M. khedr, M.A. El-dosuky and wesan Ahmed " JPEG image compression using Discrete Cosine Transform – A survey", International Journal of Computer Science & Engineering Survey (IJCSES), vol 5, no 2, pp. 29-47, April 2014.

[3] Andrew B. Watson, "Image Compression Using Discrete Cosine Transform", NASA Ames Research Centre, 4(1), pp. 81-88,1994

[4] Anjali Kapoor and Dr. Renu Dhir, "Image Compression Using Fast 2-D DCT Technique", International Journal on Computer Science and Engineering (IJCSE), vol. 3 pp. 2415-2419, 6 June 2011

[5] 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.

[6] 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

[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] Maneesha Gupta and Dr.Amit Kumar Garg, "Analysis Of Image Compression Algorithm Using DCT" International Journal of Engineering Research and Applications (IJERA), vol.2, pp. 515-521, Jan-Feb 2012

[11] N. Ahmed, T. Natarajan, and K.R. Rao, "Discrete Cosine Transform", IEEE Transactions on Computers, vol. C-32, pp. 90-93, Jan. 1974.

[12] Nitesh Agarwal and Dr. A.M. Khan "Application of DCT in image processing", International Journal of Engineering Research & Technology (IJERT), ISSN: 2278-0181, pp. 185-189, 2014.

[13] Swati Dhamija and Priyanka Jain "Comparative Analysis for Discrete Sine Transform as a suitable method for noise estimation" IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 5, No 3, September 2011 pp. 162-164.

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