ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 8, August 2013

Huffman Based LZW Lossless Image Compression Using Retinex Algorithm

Dalvir Kaur[1], Kamaljit Kaur [2]

Master of Technology in Computer Science & Engineering, Sri Guru Granth Sahib World University, Fatehgarh Sahib, Punjab, India[1]

Assistant Professor, Department Of Computer Science & Engineering, Sri Guru Granth Sahib World University, Fatehgarh Sahib, Punjab, India[2]

Abstract-Image compression is an application of data compression that encodes the original image with few bits. The objective of image compression is to reduce irrelevance and redundancy of the image data in order to be able to store or transmit data in an efficient form. So image compression can reduce the transmit time over the network and increase the speed of transmission. In Lossless image compression no data loss when the compression Technique is done. In this research, a new lossless compression scheme is presented and named as Huffman Based LZW Lossless Image Compression using Retinex Algorithm which consists of three stages: In the first stage, a Huffman coding is used to compress the image. In the second stage all Huffman code words are concatenated together and then compressed with LZW coding and decoding. In the third stage the Retinex algorithm are used on compressed image for enhance the contrast of image and improve the quality of image. This Proposed Technique is used to increase the compression ratio (CR), Peak signal of Noise Ratio (PSNR), and Mean Square Error (MSE) in the MATLAB Software.

Keywords-Compression, Encode, Decode, Huffman, LZW.

I. INTRODUCTION

Image compression is an application of data compression that encodes the original image with few bits. The objective of image compression is to reduce irrelevance and redundancy of the image data in order to be able to store or transmit data in an efficient form. Image compression means the reduction of the size of image data, while retraining necessary information. Mathematically this means transforming a 2D pixel array into a statically uncorrelated data set. The transformation is applied prior to storage or transmission of the image. At late time the compressed image is decompressed to reconstruct the image and an approximation of it. So image compression is used to minimize the amount of memory needed to represent an image. Images often require a large number of bits to represent them, and if the image needs to be transmitted or stored, it is impractical to do so without somehow reducing the number of bits.

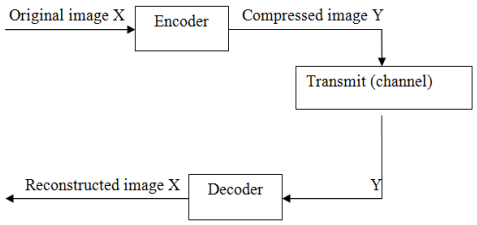

Fig1.Block diagram of image compression system [1].

A common characteristic of most images is that the neighbouring pixels are correlated and therefore contain redundant information. The foremost task then is to find less correlated representation of the image. Two fundamental components of compression are redundancy and irrelevancy reduction.

Redundancies reduction aims at removing duplication from the signal source (image/video).

Irrelevancy reduction omits parts of the signal that will not be noticed by the signal receiver, namely the Human Visual System.

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 8, August 2013

In an image, which consists of a sequence of images there are three types of redundancies in order to Compress file size. They are:

a. Coding redundancy: Fewer bits to represent frequently occurring symbols.
b. Interpixel redundancy: Neighbouring pixels have almost same value.
c. Psycho visual redundancy: Human visual system cannot simultaneously distinguish all colors.

A. Need Of Image Compression

Image compression is important for web designers who want to create faster loading web pages which make the websites more accessible to others.
Image compression also saves lot of unnecessary bandwidth by providing high quality image fraction of file size.
Image compression is also important for those people who attach photos to email which will send the email more quickly, save bandwidth costs.
For digital camera users and people who saves lots of photos their hard-drive, Image compression is more important by compression image, store more images on our hard disk thus saving the memory space.
Images transmitted over the internet are an excellent example of why data compression is important. Suppose we need to download a digitized color photograph over a computer's 33.6 kbps modem. If the image is not compressed (a TIFF file, for example), it will contain about 600 kilo bytes of data [2].

B. Image Compression Coding

Image compression coding is to store the image into bit-stream as compact as possible and to display the decoded image in the monitor as exact as possible. Now consider an encoder and a decoder as shown in Fign2. When the encoder receives the original image file, the image file will be converted into a series of binary data, which is called the bit stream. The decoder then receives the encoded bit stream and decodes it to form the decoded image. If the total data quantity of than the total data quantity of the original image, then this is called image compression. Image compression algorithms are also used the coding decoding process, same methods are used for coding and decoding means compression and decompression. The full compression flow is as shown in Fig 2.

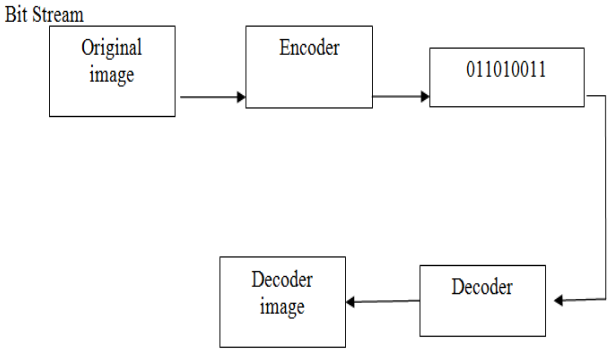

Fig.2 - The basic flow of image compression coding [1].

C. Types Of Compression

Compression can be divided into two categories, as Lossless and Lossy compression. In lossless compression, the reconstructed image after compression is numerically identical to the original image . In lossy compression scheme, the reconstructed image contains degradation relative to the original. Lossy technique causes image quality degradation in each compression or decompression step. The following are the some of the lossless and lossy data compression techniques:

- **Lossless coding techniques**
  1. Run length encoding
  2. Huffman encoding
  3. LZW coding
  4. Area coding
- **Lossy coding techniques**
  1. Transformation coding
  2. Vector quantization
  3. Fractal coding
  4. Block Truncation Coding
  5. Sub band coding

II. PROPOSED WORK

This paper proposes a compression technique using the two lossless methodologies Huffman coding and Lempel Ziv Welch coding to compress image. In the first stage, the image is compressed with Huffman coding resulting the Huffman tree and Huffman Code words. In the second stage, all Huffman code words are concatenated together and then compressed by using Lempel Ziv Welch coding. In the third stage the Retinex algorithm are used on compressed image for enhance the contrast of image and improve the quality of image.

A. ALGORITHM AND FLOWCHART

The algorithm for proposed work is given as

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 8, August 2013

Step1- Read the image on to the workspace of the mat lab.
Step2- Call a function which will find the symbols (i.e. pixel value which is non-repeated).
Step3- Call a function which will calculate the probability of each symbol.
Step4- Probability of symbols are arranged in decreasing order and lower probabilities are merged and this step is continued until only two probabilities are left and codes are assigned according to rule that the highest probable symbol will have a shorter length code.
Step5- Further Huffman encoding is performed i.e. mapping of the code words to the corresponding symbols will result in a Huffman codeword's
Step6- Concatanate all the Huffman code words and apply Lzw encoding will results in Lzw Dictionary and final encoded Values (compressed data).
Step7- Apply Lzw decoding process on Final Encoded values and output the Huffman code words
Step8- Apply Huffman Encode value on the LZW Encoding process.
Step9-In final apply the Multiscale Retinex Algorithm on cpmpressed image to enhance the quality and color og the image.
Step10-In last step the Recovered image is generated

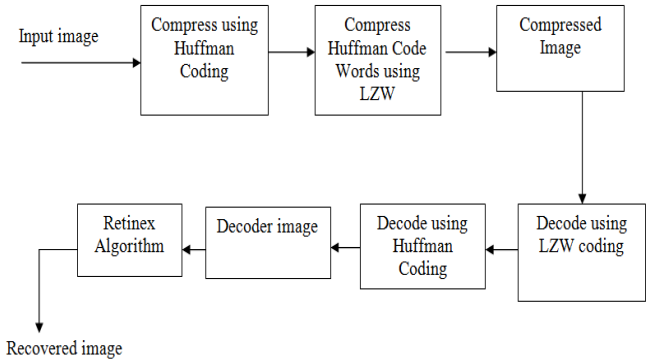

Fig 3.Flow Chart of Huffman Based LZW Lossless Image Compression using Retinex Algorithm

B. Huffman Coding and Decoding Process

Huffman Encoding Algorithms use the probability distribution of the alphabet of the source to develop the code words for symbols. The frequency distribution of all the characters of the source is calculated in order to calculate the probability distribution. According to the probabilities, the code words are assigned. Shorter code words for higher probabilities and longer code words for smaller probabilities are assigned. For this task a binary tree is created using the symbols as leaves according to their probabilities and paths of those are taken as the code words. Two families of Huffman Encoding have been proposed: Static Huffman Algorithms and Adaptive Huffman Algorithms. Static Huffman Algorithms calculate the frequencies first and then generate a common tree for both the compression and decompression processes. The compression program maintains a count of how many times each symbol has occurred so far in the source text. To encode the next symbol, the symbol counts are used as estimates of the relative probabilities of the symbols and a table of Huffman codes based on these frequencies is constructed. The Huffman code in the table is used to encode the next symbol. The decoding algorithm can re-create the same set of symbol frequencies from its de-compressed image pixels and use the table to re-construct the same table of Huffman codes. Thus it can uniquely decode one symbol, update the frequency count of that symbol, update its table of Huffman codes and then decode the next symbol, and so on.

$$\text{Avg } L = L1 * P(1) + L2 * P(2) + \ldots.. + Li * P(i)$$

$$\text{Avg } L = \Sigma\, Li * P(i)$$

In this equation Avg is count the average of the probability of all symbols. P used for probability the input symbols L1 is the leaf which count the lower probability of the symbols.L2 is the second lower probability symbols. L1+L2 create the parent of the node This process is repeated while the each symbols are processed. After processed the each symbols the tree is generated. In tree Huffman tree all leaf node assign the lower probability and addition of last two leaf node and create the parent of the leaf this process can repeated until the whole tree is processed. Huffman algorithm can used this process for encoding. The reverse of this process is used for decoding the input data.

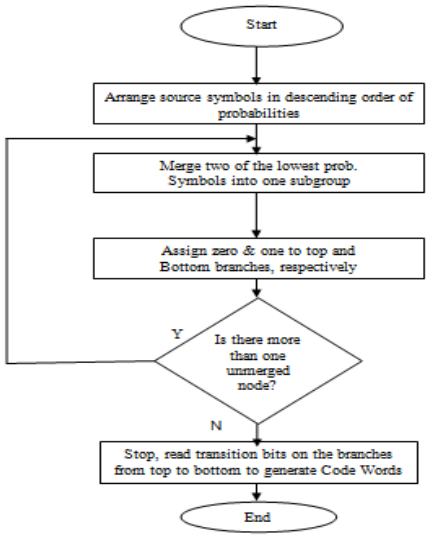

Fig4.Flowchart: Huffman algorithm.

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 8, August 2013

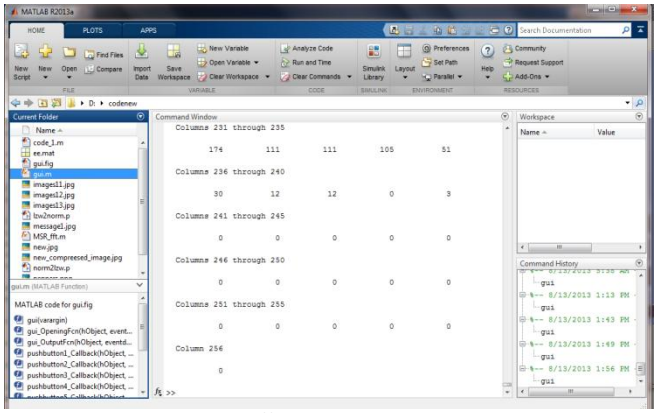

Fig 5. Huffman Coding Screenshot

C. Lzw Coding And Decoding

Lempel, Ziv and Welch (LZW) compression name is originate from the scientists Abraham Lempel, Jakob Ziv and Terry Welch. LZW compression algorithm is Simple, lossless and dictionary based compression algorithm. Dictionary based algorithms scan a file and search the sequences of data or string that occur more than once in a file. LZW compression works by replacing strings of characters with single codes without doing any analysis of the incoming text data. It adds every new found characters of string in the dictionary and data compression occurs on the single code. These strings are then stored in a dictionary and the compressed file with references are put wherever repetitive data occurred. The replaced code can be of any arbitrary length, but it must have more bits in it than a single character. The first 256 codes when using eight bit characters are initially assigned to the standard character set and the remaining codes are assigned to strings as the algorithm proceeds. The sample program runs as shown with 12 bit codes. LZW is an adaptive technique. As the compression algorithm runs, a changing dictionary of the strings that have appeared in the text so far is maintained. Because the dictionary is pre- loaded with the 256 different codes that may appear in a byte, it is guaranteed that the entire input source may be converted into a series of dictionary indexes. If "A" and "B" are two strings that are held in the dictionary, the character sequence "AB" is converted into the index of "A" followed by the index of "B". "A" greedy string matching algorithm is used for scanning the input, so if the first character of "B" is "x", then "Ax" cannot be an element of the dictionary. This means codes 0-255 refer to individual bytes, while codes 256-4095 refers to substrings. Thus, there are Advantages and disadvantages of LZW compression; the size of files usually increases to a great extent when it includes lots of repetitive data or monochrome images. LZW compression is the best technique for reducing the size of files containing more repetitive data.LZW compression is fast and simple to apply. Since this is a lossless compression technique, none of the contents in the file are lost during or after compression. The decompression algorithm always follows the compression algorithm. LZW algorithm is efficient because it don't need to pass the string table to the decompression code. The table can be recreated as it was during compression, using the input stream as data. This avoids insertion of large string translation table with compression data [16].

```
w := NIL;
while (there is input){
K := next symbol from input;
if (wK exists in the dictionary) {
w := wK;  } else {
output (index(w));
add wK to the dictionary;
w := K;}}
```

Fig.3.3 Dictionary Creation in LZW Algorithm [2].

In Fig.3.3 w is the dictionary of string starting at the current position the inner loop finds this match. The iteration then outputs the index for w and adds the string wK to the dictionary, where K is the next character after the match.

D. Steps Followed By LZW Encoding And Decoding

LZW Encoding: LZW encoding is working based on the occurrence multiplicity of bit sequences in the pixel to be encoded. Its principle consists in substituting patterns with an input image, by progressively building a dictionary.

1. Initial table with initial character strings
2. P=first input character
3. WHILE not end of input stream
4. C=next input character
5. IF P+C is in the string table
6. P=P+C
7. ELSE
8. output the code for P
9. add P+C to the string table
10. P=C
11. END WHILE
12. output code for P

LZW Decoding: In decoding process, the algorithm rebuilds the dictionary in the opposite direction; it thus does not need to be stored.

1. Initialize table with single character strings
2. OLD = first input code
3. output translation of OLD
4. WHILE not end of input stream
5. NEW = next input code
6. IF NEW is not in the string table
7. S = translation of OLD

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 8, August 2013

8. S = S+C
9. ELSE
10. S = translation of NEW
11. output S
12. C = first character of S
13. OLD + C to the string table
14. OLD = NEW
15. END WHILE

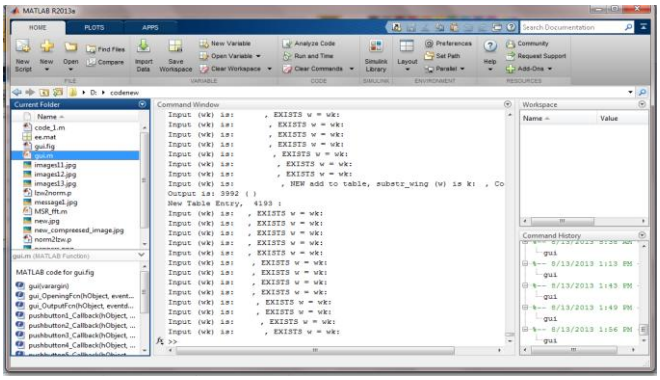

Fig 6. LZW Coding Screenshot

E. Image Enhancement Using Retinex Algorithm

Retinex algorithm is used to enhance the image contrast for dynamic compression. Retinex theory is introduced by Land to explain human's visual model, and establish illumination invariance model of which the color has nothing to do with. The basic objective of Retinex model is carry out image reconstruction, making the image after reconstruction the same as the observer saw the images at the scene. Retinex model is based on reflection imaging illumination model ,it is very similar to homomorphism filtering: irradiation light is more smooth than the changes of reflected light, you can use low-pass filter to estimate the fuzzy computing on the input image; reflected light is divided from the input image and smooth images.

Retinex ("Retina" and "Cortex" abbreviate) is proposed by Edwin Land is different from the traditional image enhancement algorithm, such as linear, nonlinear transform, image sharpening and so on, that can only enhance particular type of characteristics of the image, such as compress the dynamic range of images, or image edge enhancement, etc., Retinex balance three aspects in compress the dynamic range of gray-scale, edge enhancement and color constancy, which can be use with different types of images and self-adaptive enhance. Retinex basic principles are to be divided into an brightness image and reflection image, then enhance images to achieve the purpose by reducing the impact of image brightness on reflection [15].

Retinex provides:

- Great dynamic compression
- Increased Sharpness and Color
- Accurate scene rendition
- Fast Algorithm that is used with different color bands

a. Single-scale Retinex (SSR)

The Single-scale retinex is given by

$$Ri(x, y) = \log Ii(x, y) = \log[F(x, y) * Ii(x, y)]$$

Where Ii(x, y) is image distribution in the ith color band, F(x,y) is the normalized surround function.

b. Multiscale Retinex (MSR)

Because of the tradeoff between dynamic range compression and color rendition, we have to choose a good scale c in the formula of F(x, y) in SSR. If we do not want to sacrifice either dynamic range compression or color rendition, multiscale retinex, which is a combination of weighted different scale of SSR, is a good solution,

$$RMSRi = \sum wnRni$$

Where N is the number of the scales, Rni is the ith component of the nth scale, *wn is* weight values

c. Multiscale Retinex for color Restoration (MSRCR)

In this method enhance the MSR method to increase the color contrast of the compressed image.

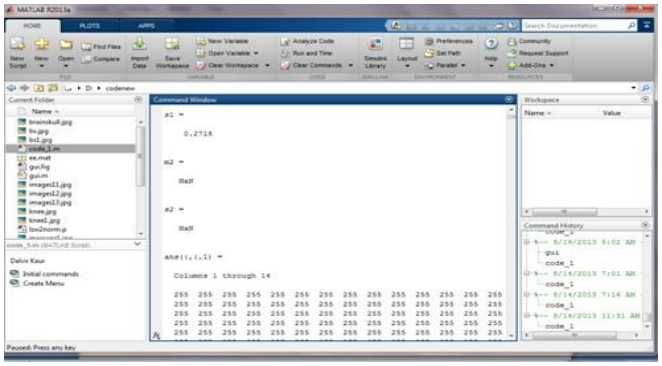

Fig 8. Image Enhancement Using Retinex Algorithm

In this proposed work we can use the Multiscale Retinex Algorithm on compressed image to improve the image quality and color of the image. Using this algorithm increase the visibility of the compressed image and we can see the clear image after the compression.

In Fig 7.we can see GUI of Huffman Based LZW Lossless Image Compression Using Retinex Algorithm. In this interface the first one is the original image and second one is the compressed image and third one is the enhance image using Retinex Algorithm.

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 8, August 2013

Fig 7.Graphical User Interface of Huffman Based LZW Lossless Image Compression using Retinex Algorithm

III. RESULTS

The experiment results are shown below.

Fig 9. Perform Huffman Based LZW Compression using Retinex algorithm on LENA image

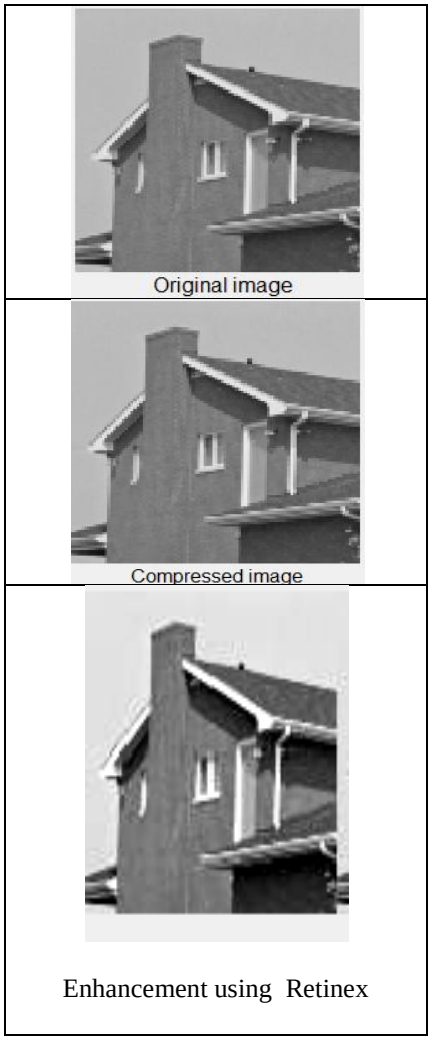

Fig 10. Perform Huffman Based LZW Compression using Retinex algorithm on house image

TABLE I

Quality Metrics Calculations on Lena and House images using Huffman Based LZW Lossless Image Compression using Retinex Algorithm

IMAGES	***Compression Ratio***	***PSNR***	***MSE***
Lena	6.23	48.06	1.87
House	5.45	47.30	1.65

Graphical Representation of Images Based on Different Quality Metrics using Huffman Based LZW Lossless Image Compression using Retinex Algorithm

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 8, August 2013

GRAPH 1

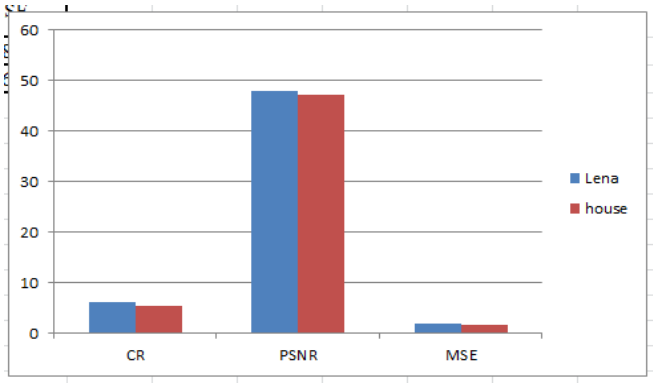

Fig 11. Quality Metrics Visualisation on Lena and House images using Huffman Based LZW Lossless Image Compression using Retinex Algorithm

IV. CONCLUSION

Compression is a topic of much importance and many applications. This thesis presents the lossless image compression on different images. Different algorithms have been evaluated in terms of the amount of compression they provide, algorithm efficiency, and susceptibility to error. While algorithm efficiency and susceptibility to error are relatively independent of the characteristics of the source ensemble, the amount of compression achieved depends upon the characteristics of the source to a great extent. It is concluded that the higher data redundancy helps to achieve more compression. Reproduced image and the original image are equal in quality by using Retinex Algorithm, as it enhances the image contrast using MSR.

V. FUTURE SCOPE

As a future work more focus can be on improvement of compression ratio using the new techniques. The proposed technique can be experimented on different kinds of data sets like audio, video, text as till now it is restricted to images. New methods can be combined and proposed that decreases the time complexity incurred in creating dictionary in LZW Algorithm. The experimental dataset in this research is somehow limited; so applying the developed methods on a larger dataset could be a subject for future research which may lead to new observations and conclusions. Further the work can be extended to video compression. Video data is basically a three dimensional array of color pixels, that contains spatial and temporal redundancy. Similarities can thus be encoded by registering differences within a frame where data frame is a set of all pixels that correspond to a single time moment.

REFRENCES

[1] Vartika Singh **"**A Brief Introduction on Image Compression Techniques and Standards" *International Journal of Technology and Research Advances Volume of 2013 issue II.*
[2] Mamta Sharmap "Compression Using Huffman Coding*" IJCSNS International Journal of Computer Science and Network Security, VOL.10 No.5, May 2010.*
[3] Sindhu M, Rajkamal R"Images and Its Compression Techniques A Review"*International Journal of Recent Trends in Engineering, Vol 2, No. 4, November 2009.*
[4] C.Saravanan,R. Ponalagusamy "Lossless Grey-scale Image Compression using Source Symbols Reduction and Huffman Coding".
[5] Mayur Nandihalli,Vishwanath Baligar "lossless gray scale images using dynamic array for predction and applied to higher bitplan" *International Journal of Scientific and Research Publications, Volume 3, Issue 1, January 2013.*
[6] R.S.Aarthi, D. Muralidharan, P. Swaminathan "DOUBLE COMPRESSION OF TEST DATA USING HUFFMAN CODE *" Journal of Theoretical and Applied Information Technology15 May 2012.*
[7] A. Alarabeyyat, S. Al-Hashemi1, T. Khdour1, M. Hjouj Btoush1, S. Bani-Ahmad1, R. Al-Hashemi "Lossless Image Compression Technique Using Combination Methods "*Journal of Software Engineering and Applications, 2012.*
[8] Md. Rubaiyat Hasan "Data Compression using Huffman based LZW Encoding Technique" *International Journal of Scientific & Engineering Research, Volume 2, Issue 11, November-2011.*
[9] Monika and Pragati Kapoor "A Huffman based LZW Approach to Perform Image Compression" 2007.
[10] A.Mallaiah, S. K. Shabbir, T. Subhashini " An Spiht Algorithm With Huffman Encoder For Image Compression And Quality Improvement Using Retinex Algorithm " *International Journal Of Scientific & Technology Research Volume 1, Issue 5, June 2012.*
[11] C. Saravanan, M. Surender "Enhancing Efficiency of Huffman Coding using Lempel Ziv Coding for Image Compression"*International Journal of Soft Computing and Engineering (IJSCE) January 2013.*
[12]Mr.D.V.Patil, Mr.S.G.Sutar Mrs.A.N.Mulla.. "Automatic Image Enhancement for Better Visualization using Retinex Technique" *International Journal of Scientific and Research Publications, Volume 3, Issue 6, June 2013.*
[13]Jiang Xing-fanga,b) , Tao Chun-kanb)." Advanced Multi Scale Retinex Algorithm for Color Image Enhancement"*International Symposium on Photoelectronic Detection and Imaging 2007.*
[14] Zia-ur Rahman a,1, Daniel J. Jobson b,, Glenn A. Woodell Investigating the relationship between image enhancement and image compression in the context of the multi-scale Retinex b J. Vis. Commun. Image R. 22 (2011) 237–250 science direct.
[15] Asadollah Shahbahrami, Ramin Bahrampour, Mobin Sabbaghi Rostami, Mostafa Ayoubi Mobarhan,"Evaluation of Huffman and Arithmetic Algorithms for Multimedia Compression Standards"
[16] http://www.TheLZWcompressionalgorithm.html
[17] http://en.wikipedia.org/wiki/Lossless_JPEG