Image Compression Using MH Encoding

M.Sathya Deepa[#1], Dr.N.Sujatha[#2]

[1] *Research Scholar,* [2] *Assistant Professor*

[1,2]*PG & Research Department of Computer Science, Raja Dhoraisingham Govt. Arts College, Sivagangai,TamilNadu, India.*

***Abstract* -- Image Compression is the application of Data Compression on digital image. The techniques used for the image compression are broadly divided into lossy and lossless compression techniques. One of the lossless compression techniques is Huffman encoding. In this paper, the new method MH Encoding (Median Filter + Huffman Encoding) is used to compress the images. Here the images are preprocessed using median filters and then the Huffman encoding is used for compressing the image to improve the compression ratio.**

***Keywords* -- compression, codewords, median filters, Huffman, encode, frequency.**

I. INTRODUCTION

Image processing [1] is a general term for the wide range of techniques that are used to manipulate, modify or classify images in various ways. In general, a digital image acquired through satellite or a digital camera is used for analysis through computers, and hence the term digital image processing. It refers to processing of a two dimensional picture in digital form by a digital computer. A digital image can be defined as an array of real or complex numbers represented by a finite number of bits. Image data compression is concerned with reduction of the number of bits required to store or transmit images without any appreciable loss of information. Compression [2] is achieved by removing one or more of the three basic data redundancies: 1) Coding redundancy, which is presented when less than optimal code words are used; 2) Interpixel redundancy, which results from correlations between the pixels of an image; 3) Psychovisual redundancy, which is due to data that are ignored by the human visual system. Image compression techniques can be broadly divided into two – Lossy and Lossless Compression.

A. Lossy Compression

A compression technique does not decompress data back to 100% of the original. Lossy methods provide high degrees of compression and result in smaller compressed files, but there is a certain amount of visual loss when restored. Audio, video and some imaging applications can tolerate loss, and in many cases, it may not be noticeable to the human ear or eye. In other cases, it may be noticeable, but not that critical to the application. The more tolerance for loss, the smaller the file can be compressed, and the faster the file can be transmitted over a network. Examples of lossy file formats are MP3, MPEG and JPEG. Lossy compression [3] is never used for business data and text, which demand a perfect "lossless" restoration.

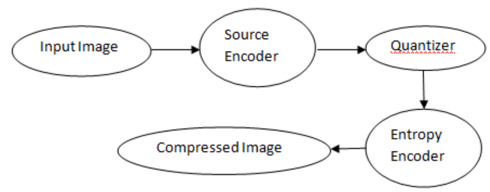


Fig 1 Block Diagram for Lossy Compression

B. Lossless Compression

In Lossless compression [4], after decompression the images are almost the same as input images. It can allow the difference between the original image and the reconstructed image up to the certain predefined value. Lossless compression can be a valuable solution where we have strict constrains on the reconstruction. In addition, this method is useful where little information on each pixel is very important. We call wavelet a loss less technique. In wavelet algorithm, we do successive re-construction. This makes it possible to receive all the information without a data loss. RLE, LZW, Entropy coding are some examples of lossless data compression.

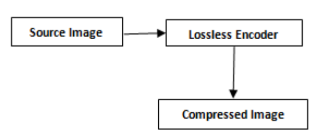


Fig 2 Block Diagram for Lossless Compression

One of the Lossless Encoding Technique is the Huffman Encoding Technique.

II. HUFFMAN ENCODING

In 1952, D.A. Huffman invented a coding technique to produce the shortest possible average code length given the source symbol set and the associated probability of occurrence of the symbols. Codes generated using these coding techniques are popularly known as Huffman codes [5]. It is used to encode messages depending upon their probabilities. The following algorithm is the Huffman coding algorithm [6] for image compression.

Huffman Coding Algorithm

- Order the symbols according to the probabilities
  Alphabet set: $S_1, S_2, \ldots, S_N$
  Probabilities: $P_1, P_2, \ldots, P_N$
  The symbols are arranged so that $P_1 \geqq P_2 \geqq \ldots \geqq P_N$

- Apply a contraction process to the two symbols with the smallest probabilities. Replace the last two symbols S_N and S_{N-1} to form a new symbol H_{N-1} that has the probabilities $P_1 + P_2$.
  The new set of symbols has N-1 members: $S_1, S_2, \ldots, S_{N-2}, H_{N-1}$
- Repeat the step 2 until the final set has only one member.
- The codeword for each symbol S_i is obtained by traversing the binary tree from its root to the leaf node corresponding to S_i.

III. MEDIAN FILTERS

The median filter [7] is a nonlinear digital filtering technique, often used to remove noise. Such noise reduction is a typical pre-processing step to improve the results of later processing. Median filtering is very widely used in digital image processing because, under certain conditions, it preserves edges while removing noise. The main idea of the median filter is to run through the signal entry by entry, replacing each entry with the median of neighboring entries. The pattern of neighbors is called the "window", which slides, entry by entry, over the entire signal. For 1D signal, the most obvious window is just the first few preceding and following entries, whereas for 2D (or higher-dimensional) signals such as images, more complex window patterns are possible (such as "box" or "cross" patterns). Note that if the window has an odd number of entries, then the median is simple to define: it is just the middle value after all the entries in the window are sorted numerically.

Given a set of random variables $X=(X_1,X_2,\ldots X_N)$, the order statistics $X_{(1)} \leq X_{(2)} \leq \ldots . X_{(N)}$ are random variables, defined by sorting the values of X_i in an increasing order. The median value [8] is then given as

$$\mathrm{median}(\mathcal{X}) = \begin{cases} X_{(K+1)} = X_{(m)}, & \text{for } N = 2K+1 \\ 1/2\left(X_{(K)} + X_{(K+1)}\right), & \text{for } N = 2K, \end{cases}$$

Where m=2K+1 is the median rank. The median is considered to be a robust estimator of the location parameter of a distribution and has found numerous applications in smoothing and denoising, especially for signals contaminated by impulsive noise. For a grayscale input image with intensity values $x_{i,j}$, the two-dimensional median filter is defined as

$$y_{i,j} = \underset{(r,s)\in\mathcal{W}}{\mathrm{median}}(x_{i+r,j+s}),$$

Where W is a window over which the filter is applied.

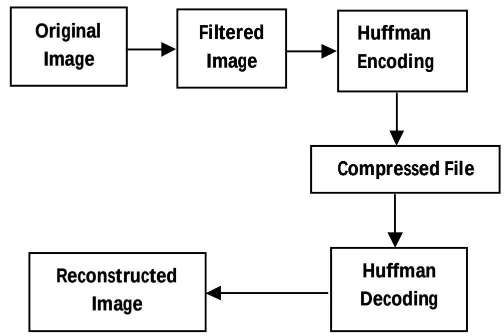
Median Filter Algorithm

- The Median Filter [9] is performed by taking the magnitude of all of the vectors within a mask and sorted according to the magnitudes.
- The pixel with the median magnitude is then used to replace the pixel studied.
- The Simple Median Filter has an advantage over the Mean filter since median of the data is taken instead of the mean of an image.
- The pixel with the median magnitude is then used to replace the pixel studied.
- The median of a set is more robust with respect to the presence of noise. The median filter is given by
- Medianfilter $(x_1 \ldots . x_N) = \mathrm{Median}(\|x_1\|^2 \ldots . \|x_N\|^2)$.
- When filtering using the Simple Median Filter, an original and the resulting filtered pixel of the sample have the same pixel.
- A pixel that does not change due to filtering is known as the root of the mask.

IV. PROPOSED METHOD

This paper proposes a new compression technique called MH Encoding for compressing the grayscale images. Usually the compression done by Huffman encoding gives low compression ratio. To increase the compression ratio we use median filters before the image is encoded. It produces a better compression when compared with normal method.

A. Methodology



B. Algorithm

- Read the image.
- If it is color image, convert it into grayscale image.
- Remove the noise using median filter.
- Resize the image matrix (MxN) into column vector (MNx1).
- Find the probability for each pixel value (symbols).
- The probability values are sorted.
- Code words are generated.
- The mapping of the code words to the corresponding symbols will result in a Huffman codeword's.
- This is the compressed file of the input image.
- To reconstruct the image, the Huffman code words are decoded.
- Then the original and the reconstructed images are displayed.

The input images are read and if it is a color image, it is converted into gray scale image. In the gray scale images, the median filter is used and the input image matrix is formatted for the performance of Huffman encoding. In Huffman coding, the pixel values are termed as symbols. The probability of these symbols are calculated and sorted to find the code words. These code words are then mapped to symbols to find the final Huffman code words. This is the final compression file of the input image. To reconstruct the image, Huffman decoding is performed. The results are displayed.

C. Results

The results are generated using MH Encoding method. Here we use bitmap images of size 256x256. If the size of the images varies, the compression ratio will be the same.

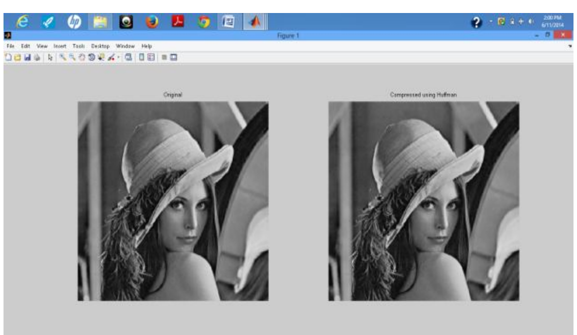


Fig 3 Original & Reconstructed Lena image compressed by Huffman encoding only

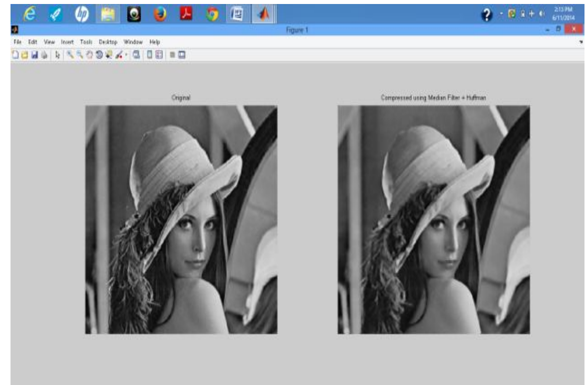


Fig 4 Original & Reconstructed Lena image compressed by Median Filter + Huffman encoding

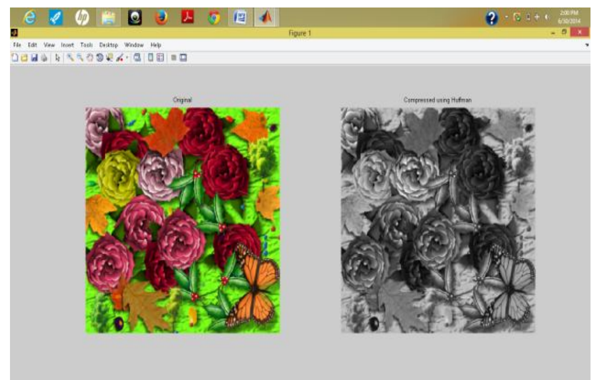


Fig 5 Original & Reconstructed color image compressed by Huffman encoding only

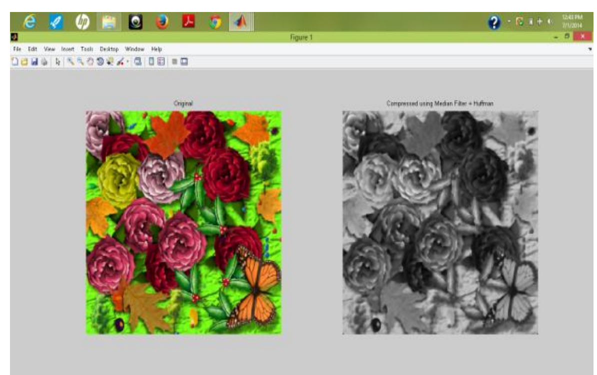


Fig 6 Original & Reconstructed color image compressed by Median Filter + Huffman encoding

Table 1 gives the compressed pixels and its file size and Table 2 gives the compression ratio using Huffman encoding, Median Filter + Huffman encoding.

TABLE 1
COMPRESSED PIXELS AND ITS FILE SIZE

S.No.	Images	Huffman		Median Filters+ Huffman	
		In Pixels	In Kb	In Pixels	In Kb
1.	Lena	62213	62	61855	61
2.	Cameraman	58469	58	57453	57
3.	Barbara	61475	61	60548	60
4.	Baboon	55154	55	54162	54
5.	Elaine	61434	61	61296	61
6.	Texture	60421	60	60348	60
7.	Moon	62651	62	62534	62
8.	Bear	61614	61	61200	61
9.	Flower	62666	62	61990	62
10.	Lena512	244719	240	243850	239

In the above table, the original pixel values are 65,536 for all 256x256 images. They are compressed to the above pixel values.

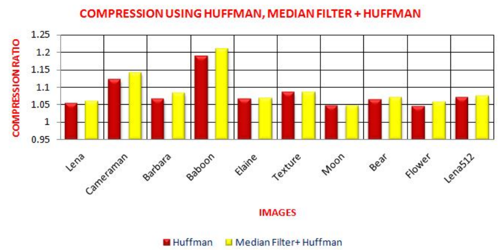
TABLE 2
COMPRESSION RATIO

S.No.	Images	Huffman		Median Filters+ Huffman	
		In Pixels	In Kb	In Pixels	In Kb
1.	Lena	1.0534	1.0525	1.0595	1.0585
2.	Cameraman	1.1209	1.1186	1.1407	1.1380
3.	Barbara	1.0661	1.0649	1.0824	1.0809
4.	Baboon	1.1882	1.1846	1.2100	1.2059
5.	Elaine	1.0668	1.0656	1.0692	1.0680
6.	Texture	1.0847	1.0831	1.0860	1.0845
7.	Moon	1.0460	1.0453	1.0480	1.0472
8.	Bear	1.0637	1.0625	1.0708	1.0696
9.	Flower	1.0458	1.0450	1.0572	1.0562
10.	Lena512	1.0712	1.0709	1.0750	1.0747

The Compression Ratio is given by

$$\text{Compression Ratio} = \frac{\text{Original Size}}{\text{Compressed Size}}$$

Table 2 shows that the compression ratio can be increased by using the median filters before using Huffman technique. Since the median filters remove the noises in the input image, it increases the compression ratio. The graphical representation for the compression ratio by only the Huffman encoding, Median Filters + Huffman Encoding is as follows.



V. CONCLUSION

We concluded that when we use only the Huffman encoding, the compression ratio is very less. If we use the median filter along with the Huffman encoding, the compression ratio is better. The reconstructed image in the new method is also has very good quality. In future, we can improve the compression ratio more by combining the Huffman encoding with other techniques as well as this method can be improved for the colored images also.

REFERENCES

[1]. S.K. Ghosh, "Digital Image Processing".

[2]. A. Alarabeyyat1, S. Al-Hashemi1, T. Khdour1, M. Hjouj Btoush1, S. Bani-Ahmad1, R. Al-Hashemi2, " Lossless Image Compression Technique Using Combination Methods", Journal of Software Engineering and Applications, 2012, 5, 752-763

[3]. http://www.pcmag.com/encyclopedia/term/46335/lossy-compression

[4]. M.B. Bhammar & K.A. Mehta, "Survey of various image compression techniques", International Journal of Darshan Institute on Engineering Research & Emerging Technologies, Vol 1, No.1, 2012.

[5]. P.S.Arun Kumar, " Implementation of Image Compression Algorithm using Verilog with Area, Power and Timing Constraints".

[6]. Jian-Jung Ding, "An Introduction to Image Compression".

[7]. http://en.wikipedia.org/wiki/Median_filter

[8]. Matthias Kirchner and Jessica Fridrich, "On Detection of Median Filtering in Digital Images".

[9]. Dr.G.Padmavathi, Dr.P.Subashini, Mr.M.Muthu Kumar and Suresh Kumar Thakur, "Performance analysis of Non Linear Filtering Algorithms for underwater images", International Journal of Computer Science and Information Security, Vol.6, No. 2, 2009.