ISSN:2320-0790

An International Journal of Advanced Computer Technology

# Huffman Coding Technique for Image Compression

**Prof. A. A. Shaikh[1], Prof. P. P. Gadekar[2]**

[1,2]P. Dr. V. Vikhe Patil Institute of Technology and Engineering (polytechnic), Pravaranagar

**Abstract:** Image compression is one of the most important steps in image transmission and storage. "A picture is worth more than thousand words "is a common saying. Images play an indispensable role in representing vitalin formation and needs to be saved for further use or can be transmitted over a medium. In order to have efficient utilization of disk space and transmission rate, images need to be compressed. Image compression is the technique of reducing the file size of a image without compromising with the image quality at acceptable level. Image compression is been used from a long time and many algorithms have been devised. In this paper we have converted an image into an array using Delphi image control tool. An algorithm is created in Delphi to implement Huffman coding method that removes redundant codes from the image and compresses a BMP image file (especially gray scale image) and it is successfully reconstructed. This reconstructed image is an exact representation of the original because it is loss less compression technique.

**Keywords:** Huffman, JPEG, GIF, BMP, Compression, loss

## 1. INTRODUCTION

### A. Need Of Image Compression

The change from the cine film to digital methods of image exchange and archival is primarily motivated by the ease and flexibility of handling digital image information instead of the film media. While preparing this step and developing standards for digital image communication, one has to make absolutely sure that also the image quality of coronary angiograms and ventriculograms is maintained or improved. Similar requirements exist also in echocardiography.

Regarding image quality, the most critical step in going from the analog world (cine film or high definition live video in the catheterization laboratory) to the digital world is the digitization of the signals. For this step, the basic requirement of maintaining image quality is easily translated into two basic quantitative parameters:

- The rate of digital image data transfer or **data rate** (Megabit per second or Mb/s)
- The total amount of digital storage required or **data capacity** (Megabyte or MByte).

Computer technology, however, provides flexible principles for processing large amounts of information. Among the algorithms available is image data reduction or 'image compression'. The principal approach in data compression is the reduction of the amount of image data (bits) while preserving information (image details).

If lossy compression has been used (a JPEG file), it will be about 50 Kbytes. The download time for these three equivalent files are 142 seconds, 71 seconds, and 12 seconds, respectively which is a huge difference .JPEG is the best choice for digitized photographs, while GIF is used with drawn images, such as company logos that have large areas of a single color.

### B. How an Image Is Stored

Considering a black and white image, it can be assumed to be made up of many pixels of different shades of gray which have a number value corresponding to the brightness or darkness of the shade. Black is 0, white is 255, and all the numbers in between are shades of gray. So, each pixel is coded as some integer from 0 to 255. To encode these integers are encoded using binary bit string. The maximum number of bits needed to code any number between 0 and 255 is 8. Bits will appear in the following form using binary code:

0 0 0 0 0 0 0 0

This 8-bit representation will code integer 0. Similarly 11111111 will code integer 255. As we move from right to left in a string the significance of that bit becomes twice. Thus the LSB holds a value

20(=1), similarly the nest LSB holds a value 21(=2) and so on. Thus the MSB will hold a value 27(=128). So, the 8-bit representation of 153 wouldlook likes this…

142 = 1 0 0 0 1 1 1 0

128 + 0 + 0 + 0 + 8 + 4 + 2 + 0 = 142

So, now we are able to encode these images using 8-bit representations of each pixel to code for a specific shade of gray. Now, once we have coded this image using this method, image compression is utilized in order to reduce the size of the image down so that fewer bits per pixel are used in the coding of the image. The idea behind image compression is to use the fewest possible bits per pixel to code the image while still having a compressed image comparable to the original image.

Image compression may be lossy or lossless. Lossless compression is preferred for archival purposes and often for medical imaging, technical drawings, clip art, or comics. Lossy compression methods, especially when used at low bit rates, introduce compression artifacts. Lossy methods are especially suitable for natural images such as photographs in applications where minor (sometimes imperceptible) loss of fidelity is acceptable to achieve a substantial reduction in bit rate. The lossy compression that produces imperceptible differences may be called visually lossless.

**C. Loss And Lossless Compression**

Methods for lossless image compression are:

- Run-length encoding – used as default method in PCX and as one of possible in BMP, TGA, TIFF
- Area image compression
- DPCM and Predictive Coding
- Entropy encoding
- Adaptive dictionary algorithms such as LZW – used in GIF and TIFF
- Deflation – used in PNG, MNG, and TIFF
- Chain codes

Methods for loss compression:

- Reducing the color space to the most common colors in the image. The selected colors are specified in the color palette in the header of the compressed image. Each pixel just references the index of a color in the color palette, this method can be combined with dithering to avoid pasteurization.
- Chroma subsampling. This takes advantage of the fact that the human eye perceives spatial changes of brightness more sharply than those of color, by averaging or dropping some of the chrominance information in the image.
- Transform coding. This is the most commonly used method. In particular, a Fourier-related transform such as the Discrete Cosine Transform (DCT) is widely used: N. Ahmed, T. Natarajan and K.R.Rao, "Discrete Cosine Transform," *IEEE Trans. Computers*, 90-93, Jan. 1974. The DCT is sometimes referred to as "DCT-II" in the context of a family of discrete cosine transforms; e.g., see discrete cosine transform. The more recently developed wavelet transform is also used extensively, followed by quantization and entropy coding.

- Fractal compression.

## 2. HUFFMAN CODING

Proposed by Dr. David A. Huffman in 1952. A method for the construction of minimum redundancy code. Huffman code is a technique for compressing data. Huffman's greedy algorithm looks at the occurrence of each character and it as a binary string in an optimal way. Huffman coding is a form of statistical coding which attempts to reduce the amount of bits required to represent a string of symbols. The algorithm accomplishes its goals by allowing symbols to vary in length. Shorter codes are assigned to the most frequently used symbols, and longer codes to the symbols which appear less frequently in the string (that's where the statistical part comes in). Code word lengths are no longer fixed like ASCII .Code word lengths vary and will be shorter for the more frequently used characters.

## 3. ALORITHM AND WORKING OF HUFFMAN CODING

Following steps describes working and algorithm for Huffman coding.

1. Read a BMP image using image box control in Delphi language. The T Image control can be used to display a graphical image-Icon (ICO), Bitmap (BMP), Metafile (WMF), GIF, JPEG, etc. This control will read an image and convert the mina text file.

2. Call a function that will Sort or prioritize characters based on frequency count of each characters in file.

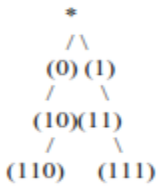
3. Call a function that will create an initial heap.Then re heap that tree according to occurrence of each node in the tree, lower the occurrence earlier it is attached in heap. Create a new node where the left child is the lowest in the sorted list and the right is the second lowest in the sorted list.

4. Build Huffman code tree based on prioritized list. Chop-off those two elements in the sorted list as they are now part of one node and add the probabilities. The result is the probability for the new node.

5. Perform insertion sort on the list with the new node.

6.Repeat STEPS 3, 4, 5 UNTIL you only have1 node left.

7. Perform a traver saloftree to generate code table. This will determine code for each element of tree in the followingway.
The code for each symbol may be obtained by tracinga path to the symbol from the root of the tree. A 1 is assigned for a branch in one direction and a 0 is assigned for a branch in the other direction. For example a symbol which is reached by branch in gright twice, then left once may be represented by the pattern '110'. The figure below depicts codes fornodes of a sample tree.




8. Once a Huffman tree is built, Canonical Huffman codes, which require less information to rebuild, may be generated by the following steps:

Step 1.Rememberthelengthsofthecodesresultingfrom a Huffman tree generated perabove.
Step 2. Sort the symbols to be encoded by the lengths of their codes (use symbol value to break ties).
Step 3. Initialize the current code to all zeros and assign code values to symbols from longest to shortest code as follows:

A. If the current code length is greater than the length of the code for the current symbol , right shift off the extra bits. Assign the code to the current symbol.
B. Increment the code value.
C. Get the symbol with the next longest code.
D. Repeat from Auntil all symbols are assigned codes.

9. Encoding Data- Once a Huffman code has been generated, data may be encoded simply by replacing each symbol with its code.

10. The original image is reconstructed i.e. decompression is d ne by using Huffman Decoding.

11. Generate a tree equivalent to the encoding tree. If you know the Huffman code for some encoded data, decoding may be accomplished by reading the encoded data one bit at a time. Once the bits read match a code for symbol, write out the symbol and start collecting bits again.

12. Read input character wise and left to the tree until last element is reached in the tree.

13. Output the character encodes in the leaf and returns to the root, and continues the step 12 until all the codes of corresponding symbols is known.

## 4. CONCLUSION AND FUTURE WORK

Huffman coding suffers from the fact that the un compressed have some knowledge of the probabilities of the symbolsin the compressed files this can need more bit to encode the fileif this information is unavailable compressing the file requirestwo passesfirst pass: find the frequency of each symbol and construct the huffman tree second pass: compress the file. This image compression method is well suited for gray scale (black and white) bit map images .This method can beimproved using adaptive Huffman coding technique that is anextension to Huffman coding.

## REFERENCES

[1] J. Ziv and A. Lempel, ``Compression of Individual Sequences Via Variable-Rate Coding," IEEE Transactions on Information Theory, Vol. 24, pp. 530--536, 1978.
[2] Chiu-Yi Chen; Yu-Ting Pai; Shanq-Jang Ruan, Low Power Huffman Coding for HighPerformance Data Transmission, International Conference on Hybrid Information Technology, 2006, 1(9-11), 2006 pp.71 – 77.
[3] Lakhani, G, Modified JPEG Huffman coding, IEEETransactions Image Processing, 12(2),2003 pp. 159 – 169.
[4] Rudy Rucker, "Mind Tools", Houghton MifflinCompany,1987
[5] J. Ziv and A. Lempel, ``A Universal Algorithm for Sequential Data Compression," IEEE Transactions On Information Theory, Vol. 23, pp. 337--342, 1977.
[6] T. A. Welch, ``A Technique for High-Performance Data Compression," Computer, pp. 8--18, 1984