Compression Techniques on Image Content

[1]Shreyanshi Basu, [2]Pranav Gaur
[1]Student, [2]Student
VIT University

***Abstract* - Conventional digital video playback systems provide only limited user interactivity, mostly in the form of VCR-like controls. In this review paper, we focus on the model where the temporal ordering and the spatial viewpoints of the video streams being viewed are completely determined at authoring time. We then wish to focus on a form of interactive video called active video, which supports hyper-linking among related video sequences and interpolation of video sequences with neighboring viewpoints, to offer end users the additional flexibility of choosing the sequencing and the viewing angle (even virtual ones) at playback time. However, active video has a substantially higher storage and transmission cost due to multiple time-synchronized video streams capturing the dynamic scene and the pixel level correspondence maps encoding the spatial association among the frame-pairs of all adjacent streams. The maps are interpolated at run-time to generate virtual views. We describe and evaluate the following three compression techniques that alleviate the storage and network transmission costs of active video: spatial video compression; lossy map compression; lossless map compression; fractal compression. This paper aims at comparing several papers based on the topic and come up with several compression techniques. It then plans to focus on few such techniques as the major methodology, perform a comparison and understand which one the best technique is.**

***keywords* - Image Compression, Huffman, Fractal Image Compression, DCT**

I. INTRODUCTION

Faster transmission of images is dependent on its size. Image Compression is the process of reducing the size of the image at a level that can still be viewed and accepted. This paper aims to study various compression techniques and determine the best technique based on a factor called the compression ration using Matlab

II. LITERATURE REVIEW

For the transmission and storage of image compression is one of the most crucial steps. There is a common saying which says that "A picture is worth more than thousand words ". Images have an important role in representing crucial information and data that needs to be saved for use in the future or can be passed over a medium. The most important reason for image compression is to maximize disk utilization space and transmission rate. Therefore, image compression is defined as the process of minimizing the file size of an image without the image quality getting compromised at moderately acceptable stage. In this paper the authors have used an image and then converted it into an array using Delphi image control tool. In this paper an algorithm has been devised in Delphi which uses the Huffman coding method. The Huffman coding removes unnecessary codes from the image and compresses a BMP image file especially the gray scale images and is successfully reconstructed. This reconstructed image constructed after the implementation of the algorithm is an exact representation of the original image because it uses the lossless compression technique.

Burrows-Wheeler Transform is a mechanism used for image compression. This master thesis has primarily studied this particular mechanism. The Burrows-Wheeler Transform is a process that is used to shrink images into the smallest available space to make sure image compression is obtained at its maximum possible level. This mechanism is primarily a data compression algorithm presented by Burrows and Wheeler. It uses a similar idea as mentioned in the above few papers that is to implement faster data transmission rate and increase the storage space vis different networks and thus achieve a better compression data ratio. The Burrows Wheeler Transforms is the best and favorite algorithm among many to compress text. It is also used to improve the performance of compression for images. This thesis has very nicely researched on the process for image shrinking and also the several factors for the Burrows-Wheeler Transforms process. This algorithm is joined with various other methods of image compression techniques such as run length encoding and move-to front and also several scanning road methods and is also tested with different block sizes to get the accurate result for image compression. The answers are received by examining empirically the processes done before such as discrete cosine transforms (DCT), discrete wavelet transforms (DWT) and predictive coding. This thesis has successfully set the JPEG image compression standard as a primary target for comparison with the several methods mentioned.

The process of converting a signal into elementary frequency parts or components is called the Discrete Cosine Transform. It is one of the most widely used procedures in image compression. In this process the authors have developed few unique and simple functions to implement the DCT and thereby compress the images. These unique functions show how Mathematics can be used in image processing algorithms and have successfully come up with a brilliant algorithm.

There is another paper that implements data and image compression. It is very well divided into an introduction, several mathematical measurement formulae and the experimental results. Images that are digital in nature have a lot of information and data that needs new and well researched techniques for the storage and the transmission of the ever-increasing volumes of information via several networks. In order to sort the problem as mentioned above we need to use the concept of image compression. Image compression as mentioned before removes the repetitive or the redundant data and at the same time

maintaining the important information in the image. It basically reduces the amount of data required to show the image that is digital in nature. This paper has implemented this concept by using a more effective and easy method of the run length coding algorithm. The algorithm proposed in this paper works on quantizing the coefficients of the discrete cosine transform (DCT) where there are a lot of coincident tokens. The several experimental results show that the proposed idea has been obtained.
In another paper a process is described which uses local operators of various scales but equally exact shape as the foundation function used for image encoding. It is different from the already known and famous techniques in that it has its code keys or elements in spatial frequency and also in space. The procedure first removes the pixel to-pixel correlation by removing a low pass filtered copy of the image from the main image. This result in a net compression of data since the difference, or error, image has low variance and entropy, and the low-pass filtered image may be shown at minimized sample density. We can compress the data further by quantizing the image that is different. These procedures are then followed repeatedly to reduce and shrink the low-pass image. Repetition of the steps at correctly expanded scales creates a data structure in the form of a pyramid. This process of encoding is as similar as to sample the image with operators of Laplacian with various scales. Because of which, the program tends to increase the minute image details. Another benefit of the proposed program is that it can be applied to various analyses of images and for the image compression tasks. The algorithms described in this paper for coding and decoding are fast and unique.
As mentioned above the data compression technique into fewer bits is called image compression. The main purpose of image compression is to minimize irrelevant and redundant data of the images to be able to store or transfer data in an effective way. Therefore, we can increase the speed of transmission and help in the reduction of the transmission time over the network by using the image compression technique. After having implemented the compression technique the image does not experience any data loss in lossless image compression. This research paper has highlighted a proposal by the authors which is basically a unique lossless compression technique called the Huffman Based LZW Lossless Image Compression using Retinex Algorithm. This technique has been represented in three stages. The first stage is where the image is compressed using the Huffman coding. In the second stage they proposed to use the LZW coding and decoding technique to compress all Huffman code words that are joined together. In the last stage the Retinex algorithm are used on the compressed image for the enhancement of the image contrast and improvement of the quality of the image. This proposed technique has been proved to increase the Peak signal of Noise Ratio (PSNR), the compression ratio (CR), and Mean Square Error (MSE) in the MATLAB Software.
In another paper the authors have proposed a new image multiresolution transform that can be applied to both lossless (reversible) and lossy compression. This new transformation that can be processed with only integer addition and bit shift operations uses a technique that is like sub band decomposition. During the calculation of the transform the number of bits required to show the transformed image is kept small by using careful scales and truncations. Numerical results show that when compared with predictive coding the entropy obtained with the new transform is smaller than that obtained with predictive coding of similar complexity. They have also proposed entropy coding methods that use the multiresolution structure and can successfully compress the transformed image for progressive transmission up to exact recovery. The paper has shown how the lossless compression ratios are the best in literature surveys.
It has been shown that for the effective and proper lossless image compression predictive coding has proven to be the most unique. The process of estimating a pixel color value based on the pixel color values of the neighboring pixels is called predictive coding. In this paper the authors have put forward a unique and easy predictive coding that estimates the pixel color value based on the pixel colors of three neighboring pixels which are quantized. This proposal is used to increase the accuracy of the particular pixel color. It can also help reduce the upper limit of the errors from the prediction of the color. The paper has shown experiments that were conducted on a set of true color 24-bit images, whose pixel colors were quantized into 2, 4, 8 and 16 colors. The results show how successful the proposed algorithm is and how it is better than some known lossless image compression techniques such as JPEG-LS and PNG. The results also show that when colors were quantized into two colors the proposed technique gives the best compression ratios.
This thesis studies five various image scanning processes and how their relationship with several images scanning processes and scanning resolution affects or disturbs the total shrinking that can be obtained with run length coding. Run length coding is a famous process used for compressing binary images. Recently we have seen though the popularity of the run length coding a line by line scanning process is always interpreted and very little attention has been to the opportunities of making use of other techniques to scan an image while it is getting encoded. This thesis has also wonderfully described a comparison between the performances of run length coding and the features of Huffman coding for the compression of binary images. To obtain the goals as mentioned in the thesis a well-organized structure of computer rules namely the Image, Scanning and Compression (ISC) system has been established in this thesis. There are several researches in the field of binary image compression that are going on this system.
Every digital image has a vast amount of data associated with them. Therefore, for the transmission and storage of digital images, image compression is the most important technology. This research paper has suggested a new image compression technique with trimming proposal based on discrete wavelet transformation (DWT). This algorithm based on the discrete wavelet transformation has shown its effectiveness over some real images, and its performance is comparable with other common compression standards. This algorithm has basically been implemented using Visual C++ and tested on a Pentium Core 2 Duo 2.1 GHz PC with 1 GB RAM. Based on experimental results it has been shown that the proposed algorithm gives more compression ratios compared to other compression techniques.
Fractal image compression is a relatively recent image compression method. Its extension to a sequence of motion images is important in video compression applications. There are two basic fractal compression methods, namely the cube-based and the frame-based methods, being commonly used in the industry. However, there are advantages and disadvantages in both methods. This paper has proposed a hybrid algorithm highlighting the advantages of the two methods in order to produce a good

compression algorithm for video industry. Experimental results show the hybrid algorithm improves the compression ratio and the quality of decompressed images.

In another paper called the fractal coding of video sequence another algorithm was proposed for fractal video sequence coding, based on the circular prediction mapping and the no contractive interframe mapping. The proposed algorithm can effectively exploit the temporal correlation in real image sequences, since each range block is approximated by the domain block in the adjacent frame, which is of the same size as the range block. The computer simulation results demonstrate that the proposed algorithm provides very promising performance at low bit rate, ranging from 40-250 Kbyte/s.

In another paper a video coding method is proposed which is based upon fractal block coding. The method utilizes a novel three-dimensional partitioning of input frames for which several efficient block-matching search methods can be used and permits spatio-temporal splitting of the input blocks to improve overall-encoding quality. After describing the basic fractal block coding algorithm, the details of the proposed three-dimensional algorithm are presented along with encoding and decoding results from two standard video test sequences, representative of video-conferencing data. These results indicate that average compression rates ranging from 40 to 77 can be obtained with subjective reconstruction quality of video-conferencing quality. The results also indicate that, in order to meet the compression rates required for very low bit rate coding, it is necessary to employ additional techniques such as entropy encoding of the fractal transformation coefficients.

III. METHODOLOGY

The following two methodologies were used in the project:

i) **Fractal Image Compression:**

To do fractal compression, the image is divided into sub-blocks. Then for each block, the most similar block found in a half size version of the image is stored. This is done for each block. Then during decompression, the opposite is done iteratively to recover the original image.

Fractal compression is a lossy compression method for digital images, based on fractals. The method is best suited for textures and natural images, relying on the fact that parts of an image often resemble other parts of the same image. Fractal algorithms convert these parts into mathematical data called "fractal codes" which are used to recreate the encoded image.

ii) **Compression using Huffman Coding**

In computer science and information theory, a Huffman code is a particular type of optimal prefix code that is commonly used for lossless data compression. The process of finding and/or using such a code proceeds by means of Huffman coding, an algorithm developed by David A. Huffman while he was a Sc.D. student at MIT, and published in the 1952 paper "A Method for the Construction of Minimum-Redundancy Codes".

The output from Huffman's algorithm can be viewed as a variable-length code table for encoding a source symbol (such as a character in a file). The algorithm derives this table from the estimated probability or frequency of occurrence (*weight*) for each possible value of the source symbol. As in other entropy encoding methods, more common symbols are generally represented using fewer bits than less common symbols. Huffman's method can be efficiently implemented, finding a code in time linear to the number of input weights if these weights are sorted. However, although optimal among methods encoding symbols separately, Huffman coding is not always optimal among all compression methods.

IV. IMPLEMENTATION

```
%program for fractal image compression using Quadtree decomposition and huffman coding clc;clear;closeall;
%%1.input Image
fname=uigetfile('*.jpg');%opens ui for select image files I=imread(fname); %reads selected image
I=imresize(I,[128 128]);%resize image into 256*256 figure,imshow(I);title('Original Image');drawnow; tic;%record time
%% 2.Quadtree Decomposition
s=qtdecomp(I,0.2,[2 64]);%divides image using quadtree decomposition of
%threshold .2 and min dim =2 ,max dim =64
[i,j,blksz] = find(s); %record x and y coordinates and blocksize blkcount=length(i); %no of total blocks
avg=zeros(blkcount,1);%record mean values for k=1:blkcount avg(k)=mean2(I(i(k):i(k)+blksz(k)-1,j(k):j(k)+blksz(k)-
1));%find mean %value end avg=uint8(avg);
figure,imshow((full(s)));title('Quadtree
Decomposition');drawnow;
%% 3.Huffman Encoding
%prepare data
i(end+1)=0;j(end+1)=0;blksz(end+1)=0;%set boundary elements data=[i;j;blksz;avg];%record total information
data=single(data); %convert to single
symbols= unique(data);% Distinct symbols that data source can produce counts = hist(data(:), symbols);%find counts of symblos
in given data
p = counts./ sum(counts);% Probability distribution sp=round(p*1000);% scaled probabilities
dict = huffmandict(symbols,p'); % Create dictionary. comp = huffmanenco(data,dict);% Encode the data.
 %% 4.Compressed
%Time taken for compression t=toc; fprintf('Time taken for compression = %f seconds\n',t);
%compression ratio bits_in_original=8*256*256;
bits_in_final=length(comp)+8*length(symbols)+8*length(sp); %Compression Ratio = total number of bits in original file,
divided by
%number of bits in final file
```

CR= bits_in_original/bits_in_final; fprintf('compression ratio= %f\n',CR);

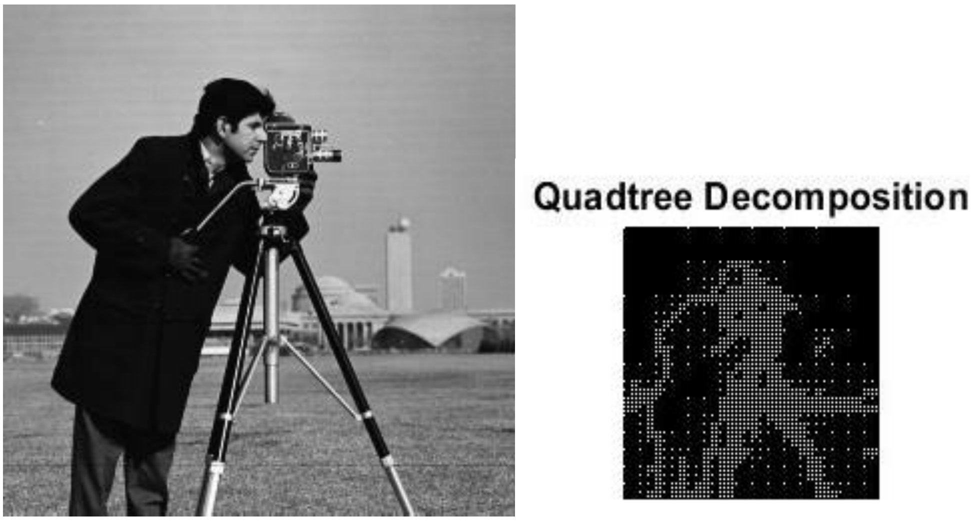

Fig 1

10.4kb 9.94kb

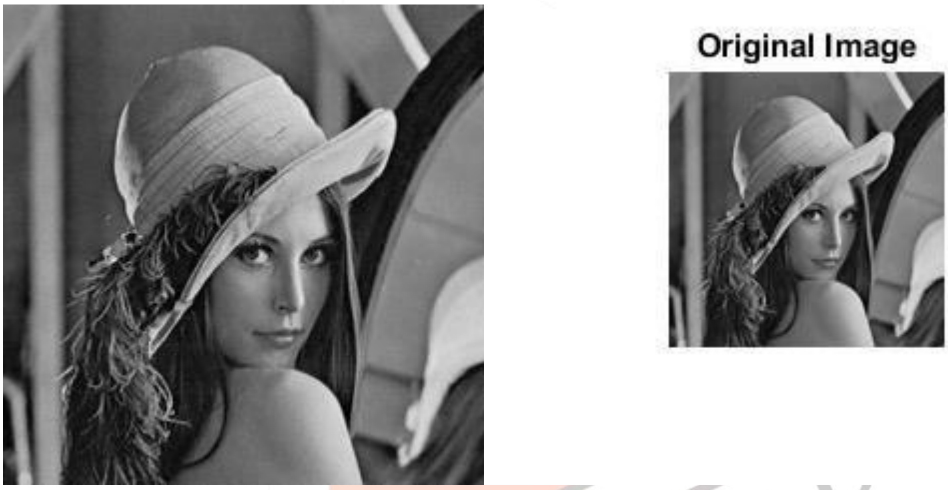

Fig 2

42.1 kb 6.59 kb

```
%Huffman Coding on image
%clearing all variableas and screen clearall; closeall; clc;
%Reading image a=imread('k.JPG'); figure,imshow(a)
%converting an image to grayscale
I=rgb2gray(a);
%size of the image
[m,n]=size(I);
Totalcount=m*n;
%variables using to find the probability cnt=1; sigma=0;
%computing the cumulative probability. fori=0:255 k=I==i; count(cnt)=sum(k(:))
%pro array is having the probabilities pro(cnt)=count(cnt)/Totalcount; sigma=sigma+pro(cnt); cumpro(cnt)=sigma; cnt=cnt+1;
end;
%Symbols for an image
symbols = [0:255];
%Huffman code Dictionary dict = huffmandict(symbols,pro);
%function which converts array to vector vec_size = 1; for p = 1:m for q = 1:n newvec(vec_size) = I(p,q);
vec_size = vec_size+1; end end
%Huffman Encodig
hcode = huffmanenco(newvec,dict);
%Huffman Decoding
dhsig1 = huffmandeco(hcode,dict);
%convertign dhsig1 double to dhsig uint8 dhsig = uint8(dhsig1);
%vector to array conversion dec_row=sqrt(length(dhsig)); dec_col=dec_row;
```

```
%variables using to convert vector 2 array arr_row = 1; arr_col = 1; vec_si = 1;  for x = 1:m for y = 1:n
back(x,y)=dhsig(vec_si);
arr_col = arr_col+1; vec_si = vec_si + 1; end arr_row = arr_row+1; end
%converting image from grayscale to rgb
[deco, map] = gray2ind(back,256); RGB = ind2rgb(deco,map); imwrite(RGB,'decoded.JPG');
%end of the huffman coding
```

Fig 3
12.6 kb, 7.87kb

Fig 4
212 KB, 69.7 KB

IV. RESULTS AND DISCUSSION

4.1 Results of Implementations

Fig No.	Original Image Size	Compressed Image Size	Compressed Ratio
1	10.4	9.94	0.95
2	42.1	6.59	0.15
3	12.6	7.87	0.62
4	212	69.7	0.32

Table 4.1: Descriptive Analysis

Table 4.1 displayed figure numbers above, original image size, compressed image size and the compressed ratio.

Advantages and Disadvantages of Fractal Compression

If the image to be compressed contains sharp edges, which are mapped to higher frequency range, then the DCT-processed file will have very noticeable ripples spreading from the edges. Furthermore, the JPEG method is resolution dependent. Higher resolution graphics have more pixels than the same images at lower resolution, hence they partition into more 8 by 8 blocks of pixels. The result is either longer compression and decompression times, or, if one chooses to fix the compression and decompression times by increasing the size of the blocks (so the 64 cosine functions will now represent more than 64 pixels), the output graphics will be very blocky due to pixel replication. Fractal image compression does not suffer from either of the above problems. Furthermore, the fractal method has the benefit of faster decompression speed, having done most of the computation during the compression step, while giving equal or better compression ratio. These advantages mean that fractal image compression is well suited for applications requiring fast access to high-quality images.

Fractal image compression is not without its limitations. Its lengthy compression step will preclude it from being used in applications where it is essential to be able to send out the compressed images with minimal delay, such as live broadcast of video over a network, tele-conferencing, and videophone.

Advantages and Disadvantages of Huffman Coding

Lossless encoding, or lossless compression, refers to the process of encoding data more efficiently so that it occupies fewer bits or bytes but in such a way that the original data can be reconstructed, bit-for-bit, when the data is decompressed. The advantage of lossless encoding techniques is that they produce an exact duplicate of the original data. Hence they are useful in compressing text and data files.

However Huffman coding isn't the best technique for compressing images. They cannot achieve high levels of compression and are extremely slow while reading and writing files.

Since Huffman coding is producing a lower compression ratio from the implementation (shown above), and after going through the advantages and disadvantages of the two compression techniques we conclude that the Huffman coding is a better compression technique.

Optimization of Huffman Coding.

Huffman encoding can be optimized in two different ways:

- Adaptive Huffman code dynamically changes the code words according to the change of probabilities of the symbols.
- Extended Huffman compression can encode groups of symbols rather than single symbols.

V. CONCLUSION

We thus conclude from the literature review and the implementations and results that Huffman coding is a better technique for image compression. Moreover, it can also be extended for compression of videos by converting the videos into a series of frames in Adobe Photoshop and then compressing each frame using Huffman compression technique.

REFERENCES

[1] A hybrid fractal video compression method M Wang, CH Lai - Computers & Mathematics with Applications, 2005 - Elsevier

[2] Fractal coding of video sequence using circular prediction mapping and non-contractive inter frame mapping CS Kim, RC Kim, SU Lee - IEEE Transactions on Image

[3] Fractal block coding of digital video MS Lazar, LT Burton

[4] MPEG-4 video verification model: A video encoding/decoding algorithm based on content representation T Ebrahimi - Signal Processing: Image Communication, 1997 – Elsevier

[5] A new lossless method of image compression and decompression usingHuffman coding techniques. JH Pujar, LM Kadlaskar - Journal of Theoretical & Applied