A Brief Study of Different Techniques used for Image and Video Compression

Rajat Kumar (201811024) Taher Makda (201811031) V. V. Vinay (201811043)

***Abstract*—Image compression and therefore video compression is categorized broadly into two parts i.e. lossy and lossless compression. Each type of compression may be distinct for different types of video dataset. The type of compression is also dependent on the user, whether he wants to decompress to original or near to original image/video. Generally lossless compression is requuired to save hue amount of data i.e. like youtube video dataset, goole images dataset, satellite images etc.**

Keywords—lossless, lossy, compression

I. INTRODUCTION

Image compression is a type of data compression technique in which the image is encoded into some form of shorter size so that it can be decompressed to original size when needed. The compressed image is represented by less number of bits compared to original. Hence, the required storage size will be reduced, consequently maximum images can be stored and it can transferred in faster way to save the time, transmission bandwidth.[1] Image compression algorithms take advantage of properties of image like color variation, size, texture, redundancy. Removing redundancy may reduce the image size but still is a lossy compression. Some examples of lossless image compression are Run-length Encoding, DPCM (Differential Pulse-code modulation), Entropy encoding, Huffman encoding, Chain codes etc. Other examples of lossy image compression include Chroma subsampling, Transform coding and fractal compression etc.

II. LOSSY COMPRESSION METHODS

A. *Transformation Coding- Discrete Cosine Transform (DCT)*

DCT is very popular due to its computational efficiency. It is used in lossy compression because it has very strong energy compaction.[2][3] The key to the JPEG baseline compression process is a mathematical transformation known as the Discrete Cosine Transform (DCT). The DCT is in a class of mathematical operations that includes the well known Fast Fourier Transform (FFT), as well as many others. It take a signal as a input and represented it from one type of representation to another. The DCT is used at the heart of the international standard lossy image compression algorithm known as JPEG. As shown in figure 1, the original imae is of a cameraman is first converted to grayscale so as to apply DCT on a 2-D matrix of image. After doing decompression, the size of the image reduces to 81.5 KB to 32.5 KB. Thus a lossy technique but still a better one.

Fig. 1. DCT compression

B. *Vector quantization*

Quantization is a process of approximating the discrete values into continuous value.[4] A vector quantizer maps k-dimensional vectors in the vector space Rk into a finite set of vectors Y = {yi: i = 1, 2, ..., N}. Each vector yi is called a code vector or a codeword. and the set of all the codewords is called a codebook. Most of time, the input to the vector quantization is discrete value but it is much more finer than that of output resolution. VQ is a powerful method for lossy compression of data such as because their vector representations often occupy only small fractions of their vector spaces. As shown in figure 2, the original image of cat is shown and 3 decompressed images are also shown. The original image size is 940 KB and decomressed image sizes are 38 KB, 47 KB and 46 KB respectively.

Fig. 2.Image Compression Using Vector Quantization with LBG algorithm

C. *Fractal Compression*

Fractal image compression (FIC) is used not only in image coding but also in many important image processing algorithms. There are two main disadvantages which restrained the development and application of FIC. First, the encoding phase of FIC is time-consuming. Second, the quality of the reconstructed images for some images which have low structure-similarity is usually unacceptable.The idea of fractal image compression, as is to find subspaces (or sub-images) of the original image space, which can be regenerated using an ifs. Where possible, if one IFS can be used in place of several IFS's which reproduce similar sub-images. [5] As can be seen in the figure 3, the image is compressed to lossy image but still quite a good technique. The decompressed image can be uniquely identified. The Quad tree approach divides a square image into four equal sized square blocks, and then tests each block to see if meets some criterion of homogeneity. If a block meets the criterion it is not divided any further, and the test criterion is applied to those blocks. This process is repeated iteratively until each block meets the criterion. The result may have blocks of several different sizes [6][7][8].

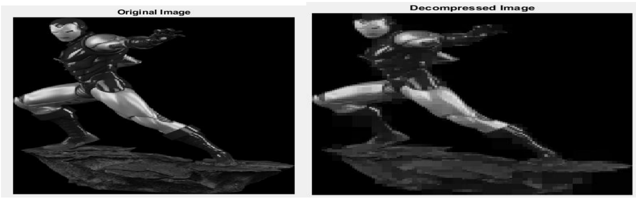

Fig.3. Fractal image compression using Quadtree decomposition and huffman coding

III. LOSSLESS COMPRESSION METHODS

A. Run Length Encoding

RLE is the easiest algorithm for compression.if same character is coming repeatedly,then it replace the same character by a value of occurance. RLE is very easy to implement and does not require much CPU horsepower.The disadvantage of this algorithm is that it can be only use in the files which have lots of repetitive data. These can be text files if they contain lots of spaces for indenting but line-art images that contain large white or black areas are far more suitable.

Fig.4. Image Compression using Run Length Encoding

B. Huffman Encoding

Huffman coding is basically an entropy based algorithm. It is a prefix free code used to compress data. This algorithm uses a tabular approach to find out the probability of symbols occuring in the particular data. It needs to know the probability of symbols in advance to work upon ints algorithm. After knowing probabilites, huffman tree is constructed using priority queue data structure and particular bits are alloted to each symbol. In images, the data has symbols from 0 to 255 for each color intensity. Therefore huffman coding can be used to compress images. One main advantage is that it is lossless and hence encoded image can be fully decompressed. The algorithm goes like this. Assume you have a source generating 4 different symbols {a1,a2,a3,a4} with probability {0.4;0.35;0.2;0.05}. Generate a binary tree from left to right taking the two less probable symbols, putting them together to form another equivalent symbol having a probability that equals the sum of the two symbols. Keep on doing it until you have just one symbol. Then read the tree backwards, from right to left, assigning different bits to different branches.[9]

Fig.5. Image Compression using Huffman Encoding.

C. Lempel-Ziv-Welch (LZW) encoding

LZW or Lempel-Ziv-Welch algorithm works on the fact of constructing a dictionary on the symbols of the data. Suppose given a binary seuence, it starts from entering the NULL value in the dictionary. Then whatever symbol it encounters, it tries to find it in dictionary. If it is able to find, it takes a symbol combining current and previous one. Again finds out in dictionary and keep doing the same. If it fails to find it in dictionary, it treats it as a new symbol. The process is completed when dicitonary is prepeared. After that it is put in encoding form.

LZW compression Algorithm[10]

Start

lastcode=NIL

LOOP

read the input image c

if ((lastcode ,c) in the dictionary

lastcode=location of (lastcode,c)

else

output=lastcode

add the lastcode and c in the dictionary

lastcode=c

End

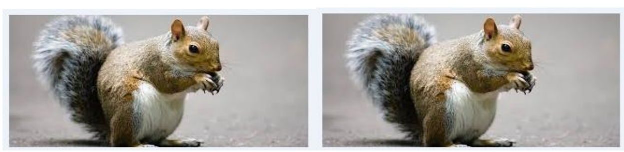

Fig.6. Image compression using LZW algorithm

D. Entropy encoding

In information theory an entropy encoding is a lossless data compression scheme that is independent of the specific characteristics of the medium.

One of the main types of entropy coding creates and assigns a unique prefix-free code to each unique symbol that occurs in the input. These entropy encoders then compress data by replacing each fixed-length input symbol with the corresponding variable-length prefix-free output codeword. The length of each codeword is approximately proportional to the negative logarithm of the probability. Therefore, the most common symbols use the shortest codes.[11]

IV. PROPOSED APPROACH

Our approach aims at compressing the image with lossy compression technique. The algorithm works on the fact that in an image, the locality of colors can be tapped. This can be explained by taking an example. Suppose there is n x m matrix of a grayscale image. Some part of image is formed with red color, some with white and some with blue. We can tap the red locality by storing the average or mean value of pixels corresponding to red in each pixel position. In this manner each cluster of locality can be replaced by the mean values and hence image can be encoded. But definitely it will be lossy. Figure 7 illustrates the image we get after decompression.

Fig.7. Image compression using proposed approach

Proposed algorithm steps are as follows.

- First convert rgb image to grayscale to get a 2-D matrix of pixels.
- Convert 2-D matrix to a 1-D matrix by reshaping it.
- Travere the image array and compare the adjacent elements to each other.
- If the absolute difference is more than a threshold like 10 or 20, the elements belong to different locality.
- Locality set of elements are replaced by mean of the elements of locality. Thus compression can be tone just by tapping the locality.
- Decompression is done by reshaping the new array and converting it to uint8.

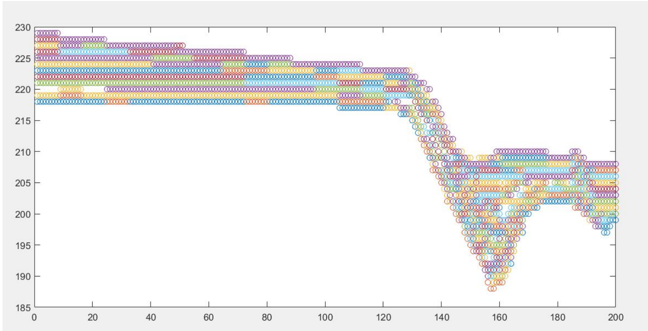

Fig.8. Locality of pixels can be seen over short interval of grayscale image

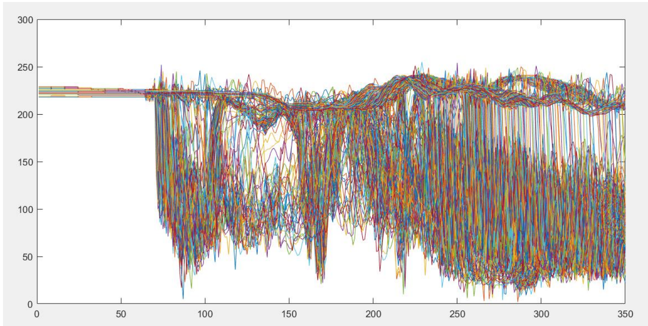

Fig.9. Pixels correspondin to grayscale image

V. CONCLUSION

Image compression is a very interesting topic. Genreally lossless compression is preferred over lossy methods. Some algorithms of lossless method may give a very good compression depending on the type of image. Compression is always linked with minimizing the redundancy. Entropy or randomness is also a good factor to determine for achieving good compressions. Our proposed approach works only with some particular dataset. With other datasets, it may or may not work well. Tapping a locality is a good step but the way to tap the locality of colors also matters. Like with grayscale, there are only two colors black and white. But with colorful image, identification of clusters of domainating pixels can compress the image sufficiently. Machine learning can be used to identify clusters and there is a very good future scope of image compression.

REFERENCES

[1] Sonal Chawla, Meenakshi Beri, Ritu Mudgil, “Image Compression Techniques: A Review”, IJCSMC, Vol. 3, Issue. 8, August 2014, pg.291 – 296

[2] Ahmed, N.; Natarajan, T.; Rao, K. R. (January 1974), "Discrete Cosine Transform", IEEE Transactions on Computers, C-23 (1): 90–93, doi:10.1109/T-C.1974.223784

[3] Yip, P (1990), Discrete Cosine Transform: Algorithms, Advantages, Applications, Boston: Academic Press, ISBN 978-0-12-580203-1

[4] Mukesh Mittal, Ruchika Lamba, “Image Compression Using Vector Quantization Algorithms: A Review”, IJARCSSE, Volume 3, Issue 6, June 2013

[5] Hitashi, Gaganpreet Kaur, Sugandha Sharma, “Fractal image compression- a review ”, IJARCSSE, Volume 2, Issue 2, February 2012

[6] Bohong Liu and Yung Yan, (2010) “An Improved Fractal Image Coding Based on the Quadtree”, IEEE 3rd International Congress on Image and Signal Processing, pp. 529-532.

[7] Hui Yu, Li Li, Dan Liu, Hongyu Zhai, Xiaoming Dong, (2010) "Based on Quadtree Fractal Image Compression Improved Algorithm for Research", IEEE Trans, pp.1-3.

[8] Dr. Muhammad Kamran, Amna Irshad Sipra and Muhammd Nadeem, (2010) "A novel domain Optimization technique in fractal image compression", IEEE Proceedings of the 8th World Congress On Intelligent Control and Automation, pp. 994-999.

[9] Mamta Sharma, "Compression Using Huffman Coding", IJCSNS International Journal of Computer Science and Network Security, VOL.10 No.5, May 2010

[10] Saumya Mishra ,.Avinash Singh,"Image compression and enhancement by using the LZW and BHEPL",International Journal of Scientific and Research Publications, Volume 7, Issue 5, May 2017 235 ISSN 2250-3153

[11] Sandeep Kaur, Sukhjeet Singh, "Entropy Coding and Different Coding Techniques", Volume 6, Issue 5, May (2016) Volume 6, Issue 5, May (2016)