HUFFMAN CODING BASED LOSSLESS IMAGE COMPRESSION

HIMANSHU SHEKHAR [1], HITESH PANT[2], RITANSHU TYAGI[3], ABHIGYAN SINGH[4]

1. *Main author, Electronics and Communication, SRM IST, Tamil Nadu, India.*
2. *Co-Author, Electronics and Communication, SRM IST, Tamil Nadu, India.*
3. *Co-Author, Electronics and Communication, SRM IST, Tamil Nadu, India.*
4. *Co-Author, Electronics and Communication, SRM IST, Tamil Nadu, India.*

ABSTRACT

Image compression is the technique that deals with the problem of reducing the amount of data required to represent a digital image. Image compression is achieved by removal of one or three basic data redundancies: (1) coding redundancy, (2) spatial redundancy, (3) irrelevant information. This can be done by Huffman coding technique. In computer and information theory, a Huffman code is a particular type of optimal prefix code that is commonly used for lossless data compression. Prefix code means that the code assigned to one character is not a prefix of code assigned to any other character. The idea is to assign variable-length codes to input characters, lengths of assign codes are based on the frequencies of corresponding characters. The most frequent occurring character gets the smallest input code and the most occurring character gets the largest code. This project is aimed at optimizing the source file by using Huffman code (lossless data compression) in today's vastly expanding technical environment where quality data transmission has become necessary. It has application in fields where it is important that the original and decompressed data be identical, like in zip file format and is often used as a component within lossy data compression techniques like mp3 encoder and other lossy audio encoder. For this we are using MATLAB R-2015 where, the result from Huffman's algorithm is viewed as a variable code table. This algorithm derives the table from an estimated probability or frequency of occurrence (weight) for each possible value of source symbol.

Keyword : - *Image compression, Huffman coding, Coding redundancy, Lossy , Lossless, Spatial redundancy,*

CHAPTER 1

INTRODUCTION

1.1 PREVIEW

A commonly image contain redundant information that is because of neighboring pixels which are correlated and contain redundant information. The main objective of image compression is redundancy it needs to represent an image by removing redundancies as much as possible, while keeping the resolution and visual quality of compressed image as close to the original image. Decompression is the inverse processes of compression that is get back the original image from the compressed image. ratio is defined as the ratio of information units an original image and compressed

Compression is performed by the three kinds of redundancies:

[1] Coding redundancies

[2] Spatial redundancies

[3] Psycho redundancies

Compression further divided into predictive and transform coding. Transform coding means, the large amount of information is transferred into very small number of blocks. One of the best examples of transformed coding technique is wavelet transform. Predictive means based on the training set (neighbors), reduce some redundancies context based compression algorithms are used predictive technique.

1.2 BACKGROUND

Image compression systems are composed of two distinct systems structural blocks encoder and a decoder. Image f(x,y) is fed into the encoder , which creates a set of symbols from the input data and uses them to represent the image . if we let n_1 and n_2 denote the number of information carrying units (usually bits) in the original and encoded images, respectively the compression that is achieved can be quantified numerically via compression ratio

$$C_R = n_1/n_2$$

A compression ratio like 10 or (10:1) indicates that the original image has 10 information carrying units (e.g., bits) for every 1 unit in the compressed data set.

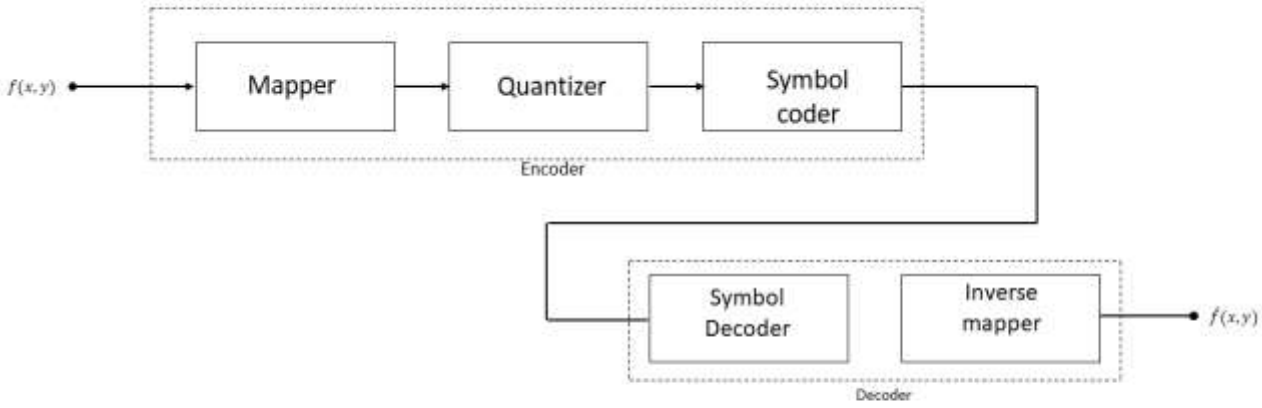


Fig 1: Compression Diagram

CHAPTER 2

LOSSLESS IMAGE COMPRESSION

2.1 CONCEPT OF LOSSLESS COMPRESSION

In the technique of Lossless compression with the compressing of data that is when get decompressed, will be the same replica of actual data. In this case, when the binary data like the documents, executable etc. are get compressed. This required to be reproduced exactly when get decompressed again. On the contrary, the images and the music also required not to be generated 'exactly'. A resemblance of the actual image is sufficient for the most objective, as far as the error or problems between the actual and compressed image is avoidable or tolerable. These types of compression are also known as noiseless as they never add noise to signal or image. It is also termed as the entropy coding as it uses the techniques of decomposition/statistics to remove/reduce the redundancy. It is also used only for some specific applications along with the rigid needs like a medical-imaging.

Below mentioned techniques consist the lossless image compression:

1. Huffman encoding

2. Run length encoding

3. Arithmetic coding

4. Dictionary Techniques

2.2 BENEFITS OF IMAGE COMPRESSION

Below are few benefits of the Image compression technique:

- It enables a reliable cost of savings that is included with the sending of less data on the network of switched telephone in which the cost of call is normally dependent on its duration.
- It is not only to decrease the requirements of storage but also decrease the entire time of execution.
- It decreases the chances of the error's transmission.
- It enables a level of the security against monitoring the unlawful Activities.

CHAPTER 3

EXISTING SYSTEM

1.Color image converted from RGB to YCbCr.

2. The resolution of chromo components is reduced by a factor of 2 or 3. This reflects that the eye is less sensitive to fine color details than to brightness details.

3. Image is converted into 8X8 matrix.

4. DCT, Quantization is applied and finally compressed.

5.Decoding is reversible process except quantization is irreversible.

3.1 BASIC CONCEPT

Morse code, invented in 1838 for use in telegraphy, is an early example of data compression based on using shorter codeword's for letters such as "e" and "t" that are more common in English. Modern work on data compression began in the late 1940s with the development of information theory. In 1949 Claude Shannon and Robert Fano devised a systematic way to assign codeword's based on probabilities of blocks. An optimal method for doing this was then found by David Huffman in 1951.In 1980's a joint committee, known as joint photographic experts group (JPEG) developed first international compression standard for continuous tone images. JPEG algorithm included several modes of operations.

3.2 CODING REDUNDANCY

R_k is distinct random variable, for k=1,2,…L with related probability

$$P_r(r_k)=n_k/n$$

where k=1,2,…L, Kth gray level in an image is represented by 'nk' and 'n' is used to specify total number of pixels in the image. (r_k) is the total number of bits used to symbolize each pixel in the still image then the average number of bits required to represent each pixel [4] is:

$$L_{avg} = \Sigma L\ l(r_k)p_r(r_k)$$

$$K=1$$

The average length of the code words is calculated by summing the number of bits used to represent the gray level and the probability that the gray level which placed in an image.

Coding redundancy is always present when the gray levels of an image are coded using a binary code. In that table 1, both a fixed and variable length encoding of a four-level image is shown. Column two represents the gray level distribution. The 2-bit binary encoding (code1) is shown in column

3. It has an average length of 2 bits. The average number of bits required by code2 (in column 5) is:

$$L_{avg}=\Sigma 4\ l2(k)p_r(r_k)$$

$$=3(0.1875) +1(0.5) +3(0.125) +2(0.1875) =1.8125 \text{ compression ratio is}$$

$$C_r =2/1.8125=1.103$$

3.3 INTER PIXEL REDUNDANCY

In order to reduce inter pixel redundancy, the 2-D pixel array normally used for human viewing and interpretation must be transformed into a more efficient format. For example, the difference between adjacent pixels can be used to represent an image. It is called mapping. A simple mapping procedure is lossless predictive coding, eliminates the inter pixel redundancies of closely spaced pixels by extracting and coding only the new information in each pixel. The new information of a pixel is defined as the difference between the actual and predicted value of that pixel.

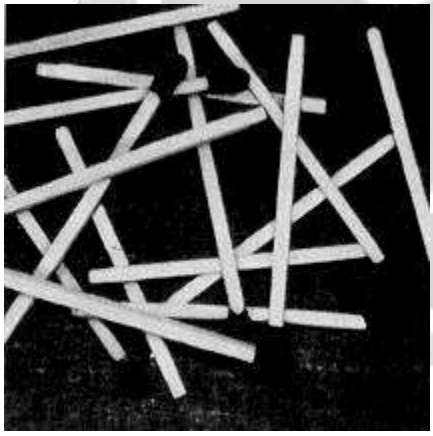


Fig 2: Sample (a)

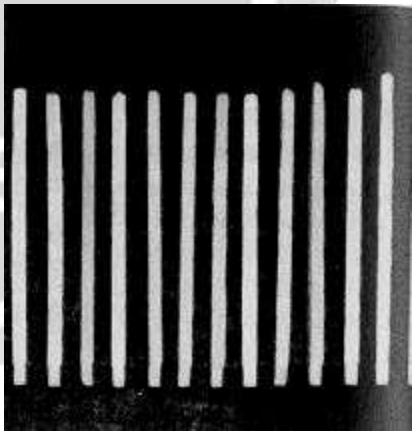


Fig 3: Sample (b)

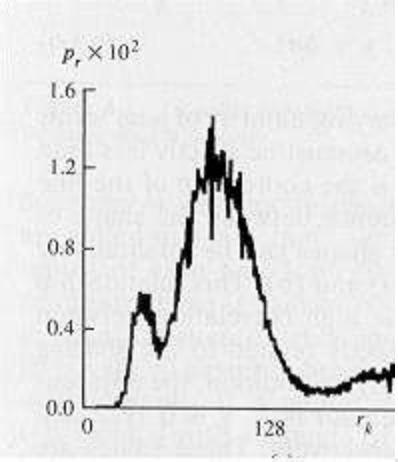


Fig 4: Histogram of sample(a) and sample(b)

These observations highlight the fact that variable – length coding is not designed to take advantage of the structural relationship s between the aligned matches in Fig 4(b). Although the pixel to pixel correlations are more important in the Fig 4(a). Because the values of the pixels in either image can be predicted by their neighbors, the information carried by individual pixels is relatively small. visual contribution of a single pixel to an image is redundant; it could be predicted by their neighbors.

3.4 PSYCHO VISUAL REDUNDANCY

In general, an observer searches for distinguishing features such as edges or textual regions and mentally combines them into recognizable groupings. The brain then correlates these groupings with prior knowledge in order to complete the image interpretation process. Thus, eye does not respond with equal sensitivity to all visual information. Certain information simply has less relative importance than other information in normal visual processing. This information is said to be psycho visually redundant. Unlike coding and inter pixel redundancy psycho visual redundancy is associated with real or quantifiable visual information. Its elimination is desirable because the information itself is not essential for normal visual processing. Since the elimination of psycho visually redundant data results in a loss of quantitative information, it is called quantization.

Fig 5: Original (a)

Fig 6: Gray level (b)

Fig 7 Gray levels/random noise (c)

Consider the image in Fig 5, shows a monochrome image with 256 gray levels. Fig 6 it is the same image after the uniform quantization to four bits or 16 possible levels. Fig 7 is called improved gray-scale quantization. It recognizes the eye's inherent sensitivity to edges and breaks them up by adding to each pixel a pseudorandom number, which is generated from the low-order bits of neighboring pixels, before quantizing the result. In fig(c) removes a great deal of psycho visual redundancy with little impact on perceived image quality.

CHAPTER 4

PROPOSED SYSTEM

4.1 CONCEPT OF HUFFMAN CODING

The proposed system is based on the lossless data compression algorithm. The idea is to assign the variable length codes to input characters. The length of assigned codes are based on the frequencies/ probabilities of corresponding characters, the most frequent character get the smallest code and the least frequent character gets the largest code .

The variable length code assigned to input characters are prefix codes ,means the codes (bit sequence) are assigned in such way that the codes assigned to one character is not prefix of code assigned to another character . This is how Huffman coding makes sure that there is no ambiguity when decoding the generated bit stream
There are mainly two part in Huffman coding :-

[1] Build a Huffman tree

[2] Traverse through the Huffman tree and assign codes to the characters

Steps to Huffman tree :-

[1] Create leaf node for each unique character and build a minimum heap of all leaf nodes (minimum heap is used as a priority queue.

[2] Extract a new internal node with them minimum frequency from the maximum heap.

[3] Create a new internal node with the frequency equal to the sum of the two nodes frequencies. make the first extracted node as its left child and the other extracted node as right child add this to the minimum heap.

[4] Repeat step 2 and step 3 until the heap contains only one node. The remaining node is the root node and the tree is complete.

4.2 ALGORITHM

In 1952 David Huffman, a graduate student at the famous Massachusetts Institute of Technology developed an elegant algorithm for lossless compression as part of his schoolwork. The algorithm is now known as Huffman coding.

Huffman coding can be used to compress all sorts of data. It is an entropy-based algorithm that relies on an analysis of the frequency of symbols in an array.

Huffman coding can be demonstrated most vividly by compressing a raster image. Suppose we have a 5×5 raster image with 8-bit color, i.e. 256 different colors. The uncompressed image will take 5 x 5 x 8 = 200 bits of storage.

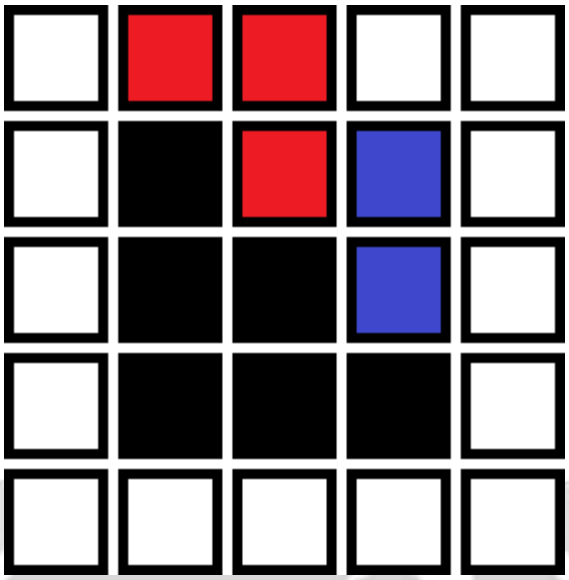
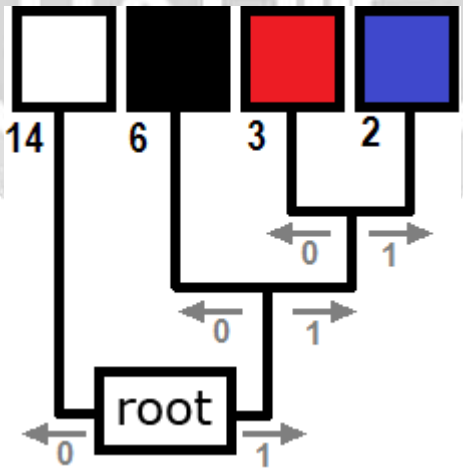


Fig 8: Grid of pixels

First, we count up how many times each color occurs in the image. Then we sort the colors in order of decreasing frequency. We end up with a row that looks like this

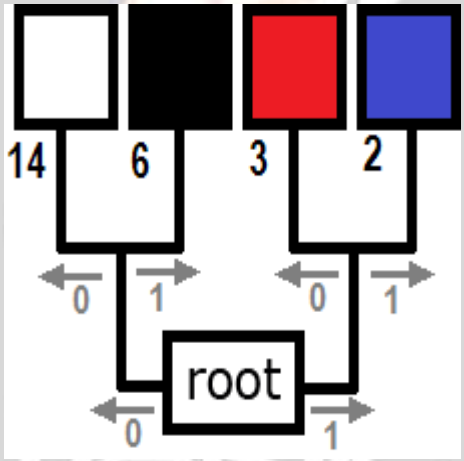
Now we put the colors together by building a tree such that the colors farthest from the root are the least frequent. The colors are joined in pairs, with a node forming the connection. A node can connect either to another node or to a color. In our example, the tree might look like this:



The result is known as a Huffman tree. It can be used for encoding and decoding. Each color is encoded as follows. We create codes by moving from the root of the tree to each color. If we turn right at a node, we write a 1, and if we turn left – 0. This process yields a Huffman code table in which each symbol is assigned a bit code such that the most frequently occurring symbol has the shortest code, while the least common symbol is given the longest code.

color	freq.	bit code
	14	0
	6	10
	3	110
	2	111

The Huffman tree and code table we created are not the only ones possible. An alternative Huffman tree that looks like this could be created for our image:



The corresponding code table would then be:

color	freq.	bit code
	14	00
	6	01
	3	10
	2	11

Using the variant is preferable in our example. This is because it provides better compression for our specific image.

Because each color has a unique bit code that is not a prefix of any other, the colors can be replaced by their bit codes in the image file. The most frequently occurring color, white, will be represented with just a single bit rather than 8 bits. Black will take two bits. Red and blue will take three. After these replacements are made, the 200-bit image will be compressed to 14 x 1 + 6 x 2 + 3 x 3 + 2 x 3 = 41 bits, which is about 5 bytes compared to 25 bytes in the original image.

Of course, to decode the image the compressed file must include the code table, which takes up some space. Each bit code derived from the Huffman tree unambiguously identifies a color, so the compression loses no information.

CHAPTER 5

SOFTWARE

5.1 Introduction

MATLAB (matrix laboratory) is a multi-paradigm numerical computing environment. A proprietary programming language developed by MathWorks, MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++, C#, Java, Fortran and Python.

Although MATLAB is intended primarily for numerical computing, an optional toolbox uses the MuPAD symbolic engine, allowing access to symbolic computing abilities. An additional package, Simulink, adds graphical multi-domain simulation and model-based design for dynamic and embedded systems.

As of 2017, MATLAB has roughly 1 million users across industry and academia. MATLAB users come from various backgrounds of engineering, science, and economics.

5.2 History

The chairman of the computer science department at the University of New Mexico, started developing MATLAB in the late 1970s. He designed it to give his students access to LINPACK and EISPACK without them having to learn Fortran. It soon spread to other universities and found a strong audience within the applied mathematics community. Jack Little, an engineer, was exposed to it during a visit Moler made to Stanford University in 1983. Recognizing its commercial potential, he joined with Moler and Steve Bangert. They rewrote MATLAB in C and founded MathWorks in 1984 to continue its development. These rewritten libraries were known as JACKPAC. In 2000, MATLAB was rewritten to use a newer set of libraries for matrix manipulation, LAPACK.

MATLAB was first adopted by researchers and practitioners in control engineering, Little's specialty, but quickly spread to many other domains.

5.3 Syntax

The MATLAB application is built around the MATLAB scripting language. Common usage of the MATLAB application involves using the Command Window as an interactive mathematical shell or executing text files containing MATLAB code.

5.4 MATLAB OPERATORS AND SYMBOLS

ARITHMETIC SYMBOLS:

Symbol	Role	More Information
+	Addition	plus
+	Unary plus	uplus
-	Subtraction	minus
-	Unary minus	uminus
.*	Element-wise multiplication	times
*	Matrix multiplication	mtimes
./	Element-wise right division	rdivide
/	Matrix right division	mrdivide
.\	Element-wise left division	ldivide
\	Matrix left division (also known as *backslash*)	mldivide
.^	Element-wise power	power
^	Matrix power	mpower
.'	Transpose	transpose
'	Complex conjugate transpose	ctranspose

RELATIONAL OPERATORS:

Symbol	Role	More Information
==	Equal to	eq
~=	Not equal to	ne
>	Greater than	gt
>=	Greater than or equal to	ge
<	Less than	lt
<=	Less than or equal to	le

SYMBOLS:

Symbol	Symbol Name	Role	Description	Examples

/
\
	Slash and Backslash	File or folder path separation	In addition to their use as mathematical operators, the slash and backslash characters separate the elements of a path or folder. On Microsoft® Windows® based systems, both slash and backslash have the same effect. On The Open Group UNIX® based systems, you must use slash only.	On a Windows system, you can use either backslash or slash:
				`dir([matlabroot '\toolbox\matlab\elmat\shiftdim.m'])`
				`dir([matlabroot '/toolbox/matlab/elmat/shiftdim.m'])`
				On a UNIX system, use only the forward slash:
				`dir([matlabroot '/toolbox/matlab/elmat/shiftdim.m'])`

..	Dot dot	Parent folder	Two dots in succession refers to the parent of the current folder. Use this character to specify folder paths relative to the current folder. • cd	To go up two levels in the folder tree and down into the `test` folder, use: `cd ..\..\test`
*	Asterisk	Wildcard character	In addition to being the symbol for matrix multiplication, the asterisk * is used as a wildcard character. Wildcards are generally used in file operations that act on multiple files or folders. MATLAB matches all characters in the name exactly except for the wildcard character *, which can match any one or more characters.	`dir('january_*.mat')` locates all files with names that start with `january_` and have a `.mat` file extension.
@	At symbol	Class folder indicator	An @ sign indicates the name of a class folder. • Class and Path Folders	Refer to a class folder: `\@myClass\get.m`

+	Plus	Package directory indicator	A + sign indicates the name of a package folder.
			• Packages Create Namespaces
				Package folders always begin with the + character:
				`+mypack`
				`+mypack/pkfcn.m % a package function`
				`+mypack/@myClass % class folder in a package`

Character Formatting Symbol:

Symbol	Effect on Text
''	Single quotation mark
%%	Single percent sign
\\	Single backslash
\a	Alarm
\b	Backspace
\f	Form feed
\n	New line
\r	Carriage return
\t	Horizontal tab
\v	Vertical tab
\xN	Hexadecimal number, N
\N	Octal number, N

CHAPTER 6

SIMULATION

6.1 Huffman Image compression:

6.1.1 Compared output of greyscale image:

6.1.2 Original Image:

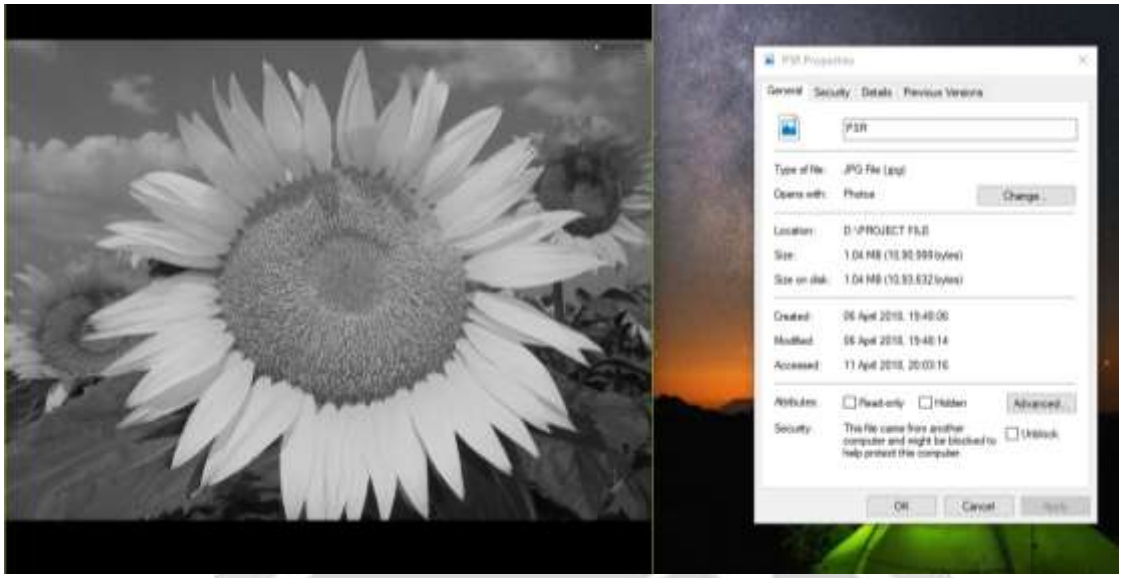


Fig 9: Original Grey-Scale Image.

The Fig 9 shows the original grey-scale image which was used for further compression. The details of the original image are as follows:

- Size of the image : 1.04MB
- Dimensions : 2048x1356
- Width : 2048pixels
- Height : 1536pixels;
- Resolution : 96dpi

6.1.3 Compressed Image:

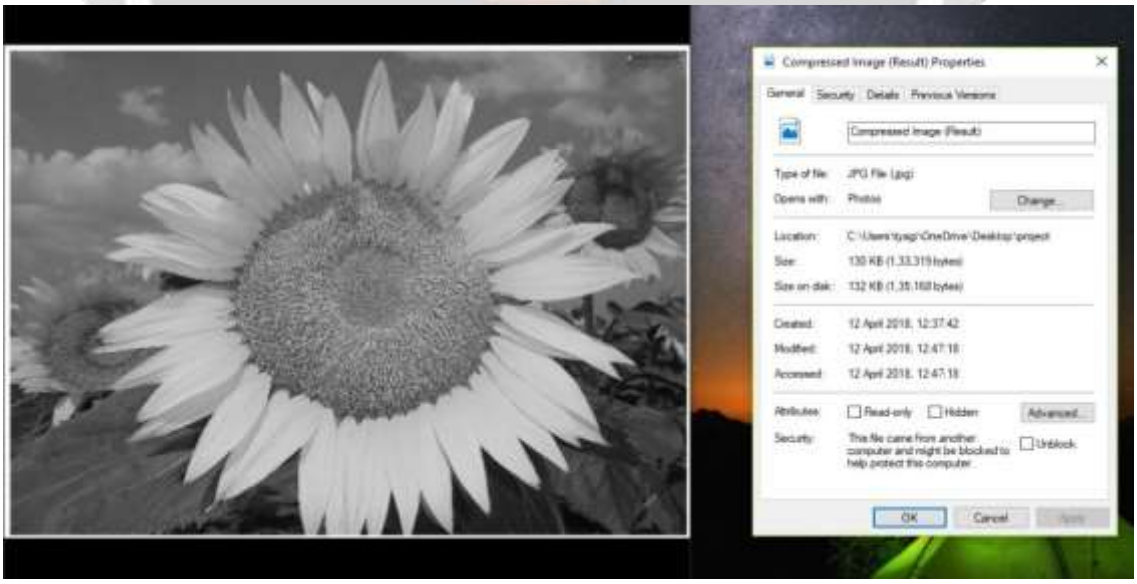


Fig 10 Compressed Grey-Scale Image.

The Fig 10 shows the same image which has been compressed using the Huffman Algorithm using MATLAB. This image is an example of lossless compression as there is no loss in resolution of the image when compared to the original image. The details of the compressed image are as follows:

- Size of the image : 310 KB
- Dimensions : 1366x756
- Width : 1366pixels
- Height : 756pixels;
- Resolution : 96dpi

From the two images compared it is clearly evident that the Huffman algorithm/code as worked as expected. The size of the image has reduced by **30%**. The major parameters that have been compressed are size, dimensions, width and height. Since the resolution of both the, original image and compressed image is same i.e. **96dpi** (dots per inch), it proves that lossless image compression has taken place**.**

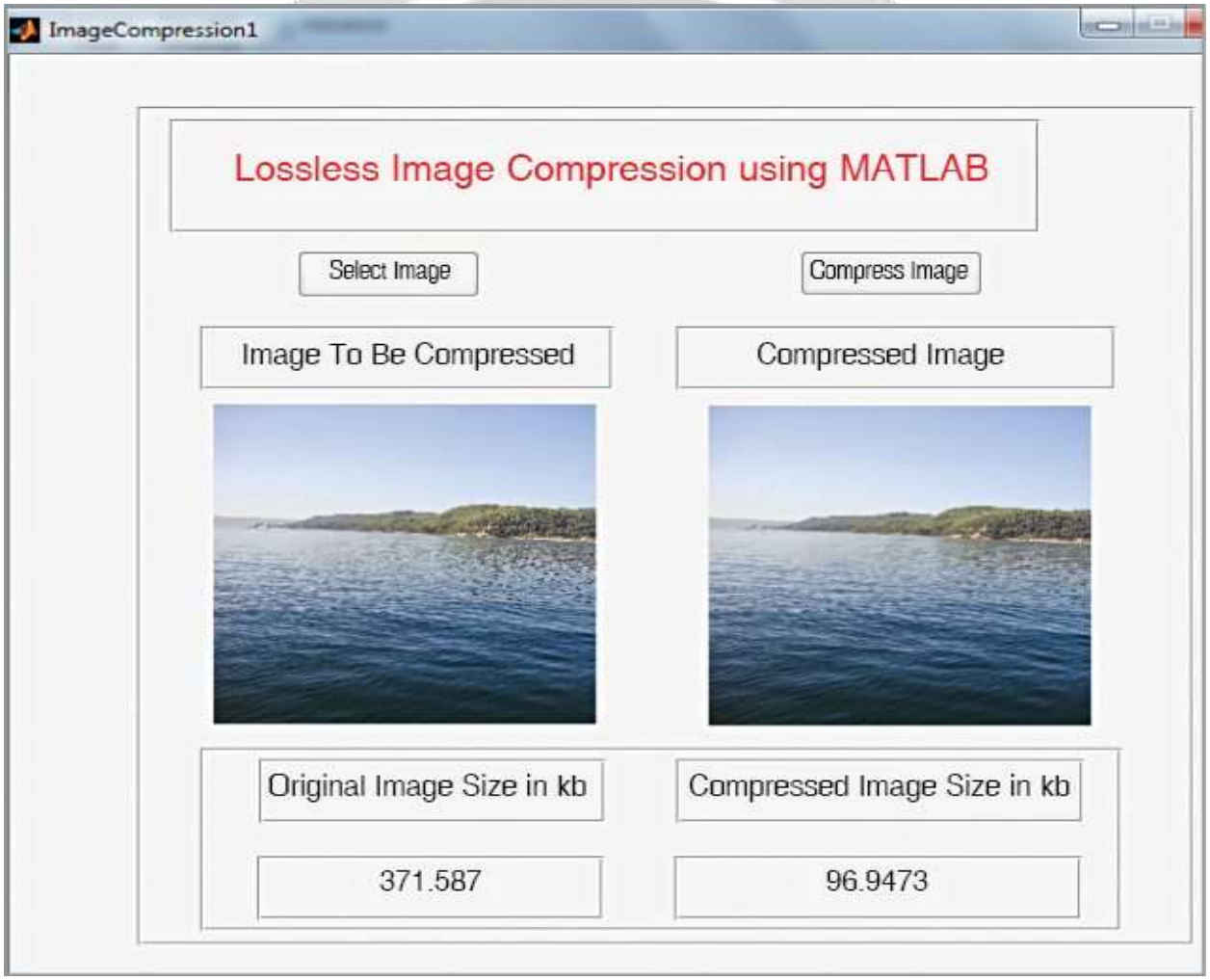


Fig 11: Lossless compression

A similar simulation is done for the colored image Fig 6.3 using the same Huffman image compression code. From the compared images it is evident that the original image which was of the size of **371.587 kb** is reduced to **96.9473 kb**. The original image as been reduced by **27%**. This process is also lossless as the resolution of both the images remain same before and after compression.

CHAPTER 7

CONCLUSION AND FUTURE ENHANCEMENT

7.1 CONCLUSION

With this we can conclude that our algorithm is doing more compression by using normalized data the reason behind this is because it is very easy to distribute frequency in normalized data then normal data due to this reason we are achieving more compression ratio. By using MATLAB as development language, we are having so advantages which are: MATLAB is general-purpose mathematical software that is user friendly and very useful for data analysis, simulations, presentation of results, and more. The software runs on both UNIX and WINDOWS. MATLAB is an interpreter, which means that it performs every command line immediately after it is entered. So you can either enter your commands line by line (prompt mode), or prepare a script file first, and then run a program. Other advantage of MATLAB is the natural notation used. It looks a lot like the notation that you encounter in a linear algebra course. This makes the use of the program especially easy and it is what makes MATLAB a natural choice for numerical computations.

7.2 FUTURE ENHANCEMENT

Data compression is most consideration thing of the recent world. We have to compress a huge amount of data so as to carry from one place to other or in a storage format. That is why data has to compress. This proposed compression technique has improved the efficiency of file (like .txt, .docx, .pptx) compression using Huffman Approach with the concepts of Typecasting and Data Normalization. For the further research in future you can try this approach to compress file like image etc. to improve the compression efficiency.

REFRENCES

[1] Enhancement in File compression using Huffman approach, IJRASET International Journal For Research In Applied Science And

Engineering Technology , Vol. 2 Issue II, Feb. 2014.

[2] Ternary Tree & A new Huffman Technique, IJCSNS International Journal of Computer Science and Network Security, Vol.10 N0.3,

March 2012.

[3] Typecasting, Legitimation, and Form Emergence: A Formal Theory Greta Hsu Univ. of California at Davis Michael T. Hannan Stanford

University László Pólos Durham University Running head: Typecasting, Legitimation, and Form Emergence March, 2012.

[4] A Study and implementation of the Huffman Algorithm based on Condensed Huffman Table, 2008 International Conference on Computer

Science and Software Engineering.

[5] A Method for the Construction of Minimum-Redundancy Codes David A. Huffman, Associate, IRE, (1952).

[6] Typecasting, Legitimating: A Formal Theory Greta Hsu Univ. Of California at Davis Michael t. Hannan Stanford University.