NUMERICAL ANALYSIS

IMAGE COMPRESSION

Project report

Submitted by:

Mamona Zahoor 2016-CS-353
Zuhha Azhar 2016-CS-361

Submitted to:

Ma'am Sahar Waqar

TABLE OF CONTENTS

IMAGE COMPRESSION

CHAPTER: 1

- ### Background:

  I. #### Image compression:

  Term "image compression" refers to a data compression process used for images. It is used to minimize the storage by images. Different algorithms are used to perform this task, these algorithms work based on statistical properties of data. We need compression to be lossless i.e. without demolishing the quality of picture. As our reliance on computer grows so is the advancement in the efficiency of storing large amount of Data.

  II. #### Methods:

  Here are a few methods to compress images;

  - Run-length encoding
  - Entropy coding
  - DEFLATE

And there are many other types but we choose according to our desired quality of output.

- ### Problem statement

We want to present a MATLAB project that will compress images with extension ".jpeg". This will help us out to perform our different tasks which demand lesser storage for images. To perform these tasks e.g. upload, download through internet, file sharing, etc. of our routine we compress image using built-in functions. But here we will be able to compress any image using an easy approach. This algorithm uses simple tools of MATLAB GUI.

CHAPTER: 2

Literature survey

There are two types for techniques of image compression:

- Lossless compression
- Lossy compression
  The technique we used is "Lossless compression".

Lossless compression:

Using this technique, the resulted image (compressed image) is numerically identical to the original one, in other words, the matrix of such image has no variation. It has many applications as discussed before. This process consists of two basic steps:

- Generation of statistical model for input image
- Mapping of data to bit sequence such that data will produce lesser output than improbable input.

Now, let's have a look for some commands used in this project:

- ### Uigetfile instruction:
  This instruction is used to open a dialogue box for data selection. Here, it is used for input image selection. It takes a filter for specific extension of image files and a string to display on dialogue box, as arguments.

- ### dir instruction:
  This instruction returns the information of file i.e. name, size, folder, modification date, datenum, isdir.

- ### Image read/write/show:
  Instructions imread, imwrite and imshow are used for image file reading, writing to a file and displaying of image.
  Other instructions include:
  - Im2double(Image datatype conversion)
  - DCT (discrete cosine transform)
  - Blockproc (Process image by applying a function), etc.

CHAPTER: 3

Methodology/Algorithm

It simply compresses the image by using DCT algorithm. In this process of image compression the size of data required to represent a given quantity of information is reduced without effecting the quality of image. On running the code a screen appears with 2 buttons. The "SELECT IMAGE" button let user to select the image of his own choice (but there is a restriction that image extension should be with ".jpeg" or ".bmp" or ".png" or ".tiff") When user clicks on "SELECT IMAGE" button uigetfile function is called which allows the user to select a particular image and this command is also responsible for selecting images with particular extensions. Then dir command is used to get complete information about image (It returns image size, name, folder, modification date, datenum, isdir) and then the size of image in terms of bytes is calculated and the string on UI is set to the size of actual image. Now we are ready to compress the image. Let's move to the next "COMPRESS IMAGE" button (what is happening at its back end).

On clicking compress image button user gets the size of compressed image on screen and the compressed image itself get saved in the same folder from where the original image was taken with name "imagecompression1.xxx"(Here xxx is not specifying any extension although it means that the compressed image will be with extension similar to extension of original image). On the click event of "COMPRESS IMAGE" button it is assured that image has been selected or not. If image is not selected an error message with text "Please Select an image" appears on the screen and if the image has been selected then the procedure is further proceeded.

Image is actually compressed by DCT algorithm. It works by using equation:

$$D(i,j) = \frac{1}{\sqrt{2N}} C(i)C(j) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} p(x,y) \cos\left[\frac{(2x+1)i\pi}{2N} \right] \cos\left[\frac{(2y+1)j\pi}{2N} \right]$$

Where p(x, y) represent a particular pixel located at (x, y) position and N represents size of block on which DTC is performed. Equation actually calculates one entry which is from transformed image to actual image matrix. It works by separating different parts of image on the basis of their frequencies then the frequencies which are of less importance are discarded and storage space is reduced in this way. As it uses cosine function so its resulting matrix highly depend upon vertical frequencies and horizontal diagonal. That is why a complete black image has random looking matrix whereas image which contains only one color has first element quite large and other elements are zeros.

After DTC the image is finally compressed by "QUANTIZATION" process. Through specific quantization matrices, the compression according to the users wish is performed. It enables the user to decide that he wants the quality of 0 or 100 or somewhere in between. 0 represent lowest quality and 100 is high quality image. At value 50 matrix has both high compression quality as well as best decompression quality.

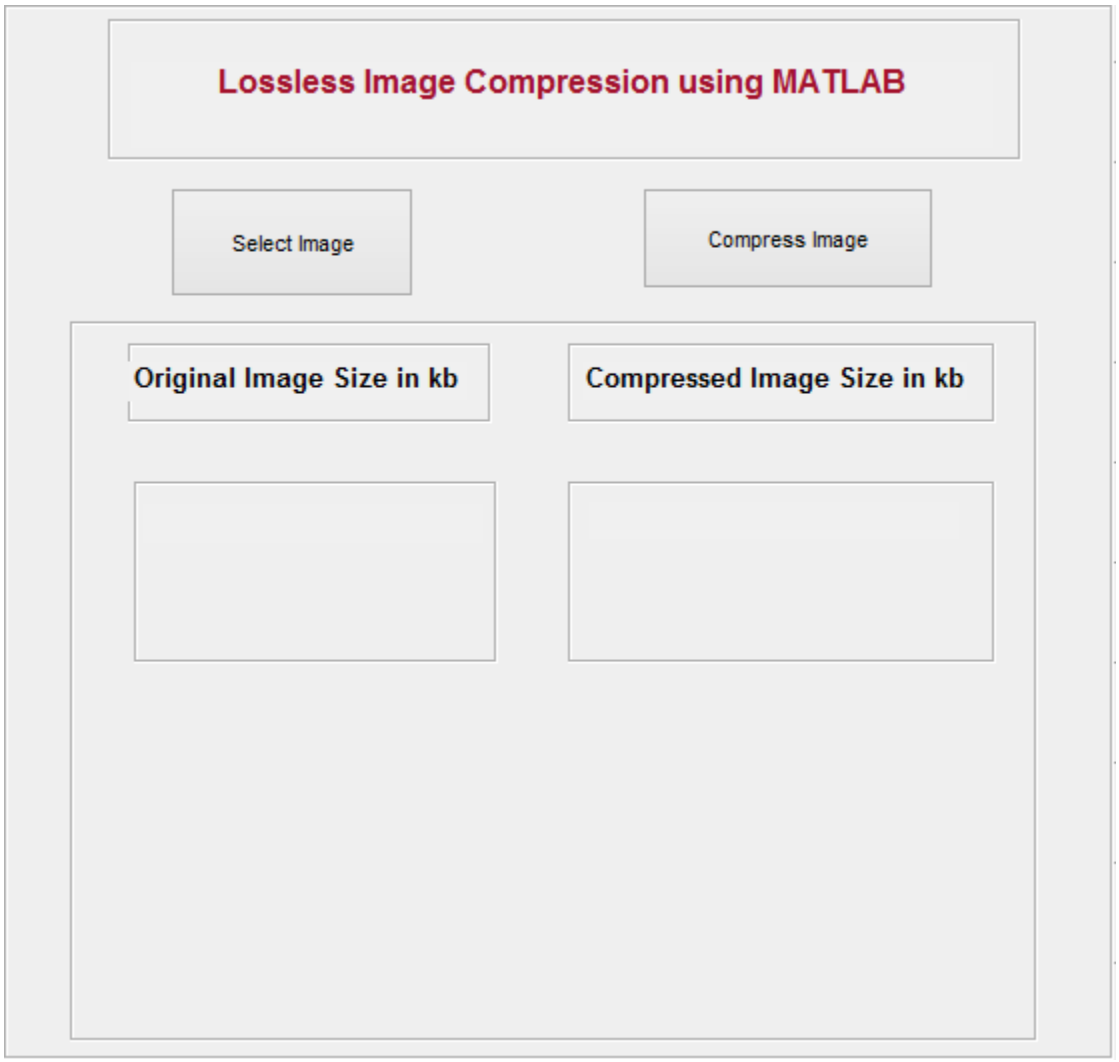
Elements in transformed matrix are divided by corresponding actual matrix and then rounding to nearest value in quantization.

$$C(I, j) = \text{round}(D(I, j)) / Q(I, j))$$

Where D is transformed matrix and Q is quantized matrix. The zeroes in Transformed matrix represent less important higher frequencies which are discarded and image is compressed.

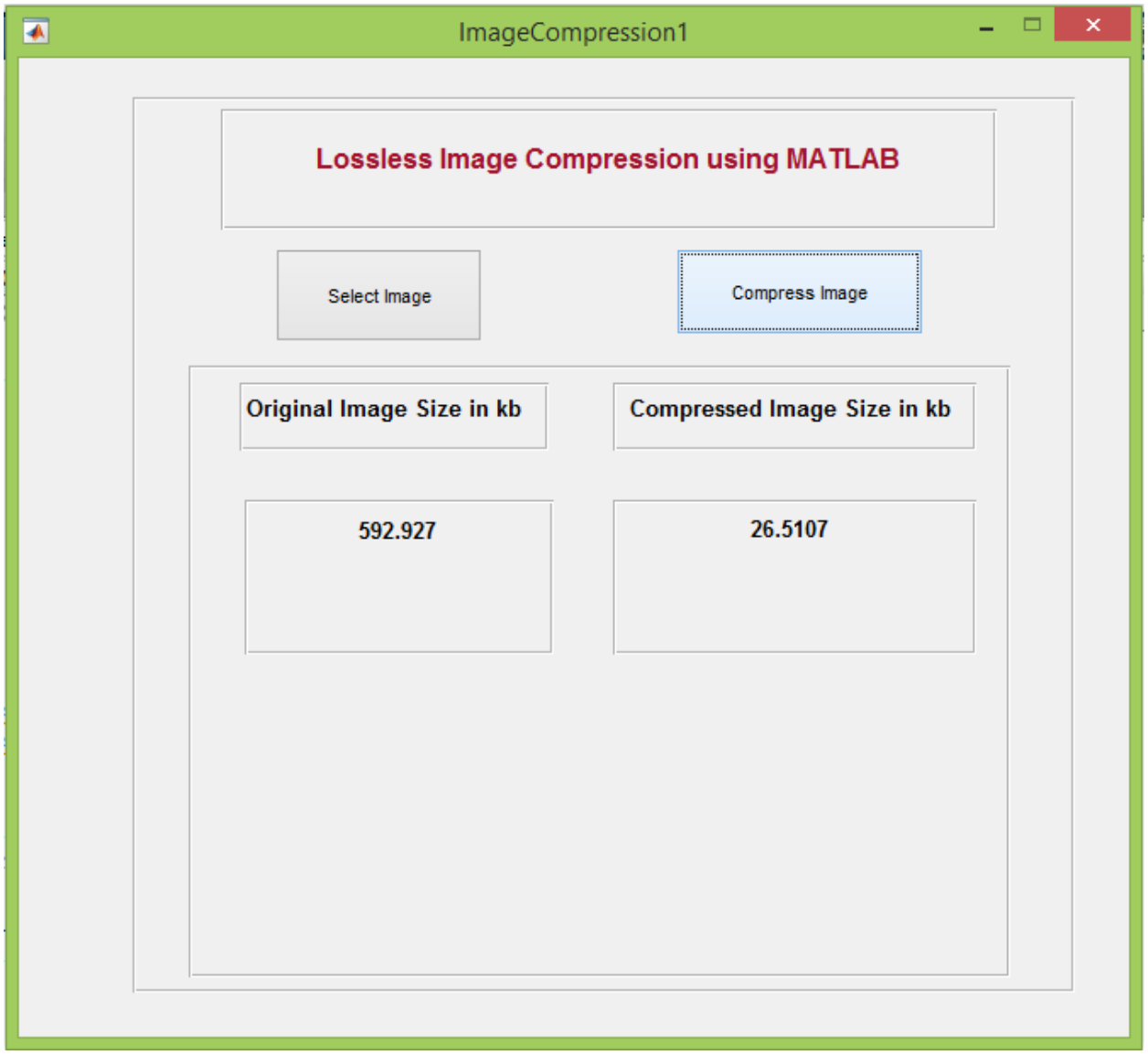
CHAPTER: 4

Graphic User Interface



CHAPTER: 5

Results



CHAPTER: 6

Future Directions

Our project compresses '.jpeg' images specifically. So many advancements can be made in this aspect. It is not the best definitely and there is a great room for improvement. For future working on this project we can extend it for other extensions like .tiff, .fig, .bmp and so on.

Compression of videos:

Compression of videos is also an important task now a days. These compressions are very useful in our different workings. With help of such a convenient and easy to handle tool one can compress required files immediately. Similarly, the decompression of images can be achieved using matlab tools and instructions. These operations produce low cost memory usage which is a basic need now a days, especially, for working with internet where we want speedily uploading and downloading of files.

For medical field:

Image compression can be extended to the medical field as medical images are significantly different from ordinary images. Generally, the medical images are not colored because colored images may not carry the significant clinical attributes. So, one may need to use different functionalities and other approaches to compress such image.

Moreover, image compression can be beneficial in industrial work.

CHAPTER: 7

References

- researchoutput.csu.edu.au
- ch.mathworks.com
- www.sciencedirect.com
- http://citeseerx.ist.psu.edu
- www.ijser.org