Department

Of

Electrical Engineering and Computer Science

South Dakota State University

Computer Assignment 1

EE 575

Submitted By:

Yugeen Chaulagain

Submitted To:

Dr. Songxin Tan

September 8, 2016

1 Introduction

In this era machines are getting smarter and there is rapid evolution in information technology. There are various vital tools and technologies being catalyst in this growth and among all image processing is one of the vital. It has become backbone in different fields as security and navigation, advanced medical technologies, forensics research etc. In general it can be said that image processing provides vision to machines. We have seen some very smart projects on fusing processed image data with speech and other information from various sensors.

Being this smart it is not that easy to fuse processed image data in our system. Image processing requires sophisticated mathematical tools and multidimensional matrix for storing color information. Before getting into the vague part of the technology firstly we have learned about some common image file formats and then we started using the well-known tool MATLAB with common commands and functions to manipulate image file with different extensions.

2 Summary: common commands used

Here I have summarized some of the common commands used in this assignment from the beginning referencing MATLAB help and doc file.

2.1 File Operation

```
file_id = fopen(‘file_name’, mode)    % open files with different mode
data_arry = fread(file_id, inf, 'uint8'); %Read file indicated by id,
save data bytes into data_arry
fclose(file_id);      % Close the opened file
```

2.2 Matrix/Array manipulation

```
num_elements = numel(mat);  % find number of elements available in an
array or matrix
% reshape vector or matrix into new row x comln matrix and return to
new_matrix
new_matrix = reshape(mat_vect, row, comln);
my_matrx = randn(m, n); % create m x n random matrix.
my_mat= ones([m n]); % Create mXn matrix with all the elements being 1
%  do concatenation of mat_1 and mat_2 horizontally vertically or
diagonally as specified by
% variable orient and save new matrix into new_val
new_val = cat(orient, mat_1, mat_2);
```

2.3 Common Image related commands

```
image(mat) % display image from the bytes stored at mXn matrix 'mat'.
imagesc(dat) % isplay image from the bytes stored at mXn matrix 'mat'
stretching to colormap formed from the particular 'dat' image matrix.
% Read graphics file and return the data bytes of the file into matrx
matrx = imread(grp_file)
colormap(map) % map each pixel represented by a byte to the colormap
value stored at 'map'
```

3 Methodology

After studying about the problem statement and instructions provided with the assignment I started studying and practicing the commands and syntax required to solve the problem. Firstly I went through all the commands required to manipulate a file in different modes. I failed learning about the commands used for manipulation of images. Immediately I went for knowing image formats, structures and arrangements of color information and then again I continued learning the commands necessary to manipulate image. Understanding the commands used for manipulating an image stored in any format was painful and I have invested lots of time understanding the command colormap and handling it in proper way. After it I made general practice in useful commands used for manipulation of matrix and arrays and started solving problems.

4 Solutions

Here are the solutions to the problems done on my own. About reference, off course I have referred some texts from some popular books, suggestions made by an unknown guy at matlab forum, lots from matlab documentation and help etc.

a. Load a square 8-bit image (file name: lena.dat) calculate its height and width plot the image and also plot the image adjusting 256 grayscale for 8-bit image.

```
% Clean previous variables, figures and results
clear all;
close all;
clc;

% Read raw File (begining of file operation)
file_id = fopen('images\lena.dat', 'r');
% save uint8 data from the image.dat file
mat_img = fread(file_id, inf, 'uint8');
% Close the file (end of file operation)
fclose(file_id);
```

```
% Since the image is square
% Image Height = image widht = sqare root of (no of bytes)
size_of_array = numel(mat_img);
img_size  = sqrt(size_of_array);

% Construction of matrix as that of data arranged in square image
% without any headers.
fill_img = reshape(mat_img, img_size, img_size);

% plot original image
figure('Name','Image: original');
image(fill_img);

% plot grayscale adjusted image
% using 'gray' colormap command
figure('Name','Image: grayscale adj');
image(fill_img);
colormap(gray);
```

b. Load image sundial_bw.tif, display as 256 grayscale, convert it to binary data file (.dat extenshion) and save it.

```
% Clean previous variables, figures and results
clear all;
close all;
clc;

% Read known type graphics image
read_into_matrix = imread('images\sundial_bw.tif');

% plot grayscale adjusted image
% using 'gray' colormap command
figure('Name','Image: Grayscale sundial ');
image(read_into_matrix);
colormap(gray);

% Find no. of rows and columns
[img_h, img_w] = size(read_into_matrix);

% Reshape matrix to vector for storing to a binary data file
array_vlues = reshape(read_into_matrix, img_h * img_w, 1);

% create file with write and append permission
write_file_id = fopen('outputs\sundial_bw.dat','a');
% save uint8 data from the image.dat file
fwrite(write_file_id, array_vlues,'uint8');
% Close the file (end of file operation)
fclose(write_file_id);
```

c. Create the checkerboard. Size: 64x64 pixel with each black/white square being size of 8x8 pixel. Dark square color value: 10 and Bright square color value: 200. Later save it into a .dat format as in previous problem

```
% Clean previous variables, figures and results
close all;
clear all;
clc;

% Create square boxes
black_square = 10 * ones([8 8]);    % dark square value 10, area: 8x8
px
white_square = 200 * ones([8 8]);   % bright square value 200, area:
8x8 px

%concatination for making a black and white square
pair_w_first = cat(2, white_square, black_square); % White square
first
pair_b_first = cat(2, black_square, white_square);  % Black square
first

% Create a row of 8 b/w square boxes, white square first
for i = 2:7
    if(rem(i, 2))
        pair_w_first = cat(2, pair_w_first, black_square);
    else
        pair_w_first = cat(2, pair_w_first, white_square);
    end
end

% Create a row of 8 b/w square boxes, black square first
for i = 1:6
    if(rem(i, 2))
        pair_b_first = cat(2, pair_b_first, black_square);
    else
        pair_b_first = cat(2, pair_b_first, white_square);
    end
end

% Create 8x2 boxes. white square first in first row, blck box first in
2nd
whole_img = cat(1, pair_w_first, pair_b_first);

% create whole checker box with above 8x2 pair
for i = 2:7
    if(rem(i, 2))
        whole_img = cat(1, whole_img, pair_b_first);
    else
        whole_img = cat(1, whole_img, pair_w_first);
    end
```

```
end

% The matrix is created corrosponding to the image
% show the image from 64x64 matrix using image function
% Uncomment this and fn below (colormap) to see the preview of image
formed
image(whole_img);

% Command for using only black and white color
colormap(bone);

%---------------------------------------------------------------------
-----
% binary file creation

img_h = 64; % Known to us
img_w = 64;

% Reshape matrix to vector for storing to a binary data file
array_vlues = reshape(whole_img, img_h * img_w, 1);

% create file with write and append permission
write_file_id = fopen('outputs\checkerboard.dat','a');

% save uint8 data from the image.dat file
fwrite(write_file_id, array_vlues,'uint8');

% Close the file (end of file operation)
fclose(write_file_id);
```

d. Create a function named ‘loadim’ which takes arguments as file name, image height and width and creates and saves an image of any size with any file in the same matlab workspace.

```
function loadim(image_name, image_w, image_h)
    % clean
    close all;

    % Creating a matrix with random values and known dimensions
    image_raw = 255 * abs(randn(image_h, image_w));

    % plot image name
    figure('Name',image_name);
    imagesc(image_raw);


    % Reshape matrix to vector for storing to a binary data file
    array_vlues = reshape(image_raw, image_h * image_w, 1);
    % create file with write and append permission
    write_file_id = fopen(strcat(image_name, '.dat'),'a');

    % save uint8 data from the image.dat file
    fwrite(write_file_id, array_vlues,'uint8');

    % Close the file (end of file operation)
    fclose(write_file_id);
end
```

5 Colculusion

This assignment was very much needed for getting started into image processing in matlab. This also helped me to know some common formats in which images are stored, concept of 24-bit color of a pixel etc. While solving for the checker board I found my confident raised in problem solving technique too. On the other hand exposure to more handy and useful data manipulation functions of matlab was even more beneficial to have for further study.