**Extraction of bit planes in a grayscale image using MATLAB**

Let’s read image:

% Section 1

myimage = imread('cameraman.tif');

imshow(myimage)



Let’s add a threshold value as 120 whatever pixel is greater than 120, store it as 1 else 0. It can be 256/2 as well.

%% Section 2

a = myimage>120;

imshow(a)

Generated image includes only ones and zeros. So it is a binary image.



Let’s convert image into double before processing. Myimage\_double values are between 0 and 1.

%% Section 3

% Convert your images to type double before processing.

myimage\_double = double(myimage);

% double is between 0 and 1 so we have to divide this to 255

imshow(myimage\_double/255)

% or , you can directly use this function

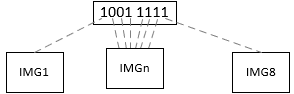
% imshow(im2double(myimage))



Let’s say that we have a grayscale image and it has 8 bits value for each pixel which is double for matlab. (myimage\_double)

If a pixel is dec 159 then binary version is 10011111.

Let’s say that we want to generate image from each bit like this :



To be able to extract each bit from 8 bit value we need to use mod function.

For example:

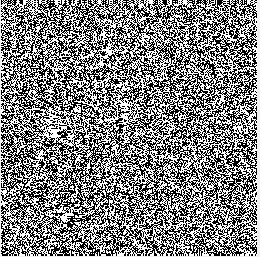
LSB Value: Bit 0 : 159 **mod** 2 = 1

Let’s do this for the whole image and get only bit 0 for all like this. So we have LSB values for each pixel of the image. They are only ones and zeros.

%% Section 4

c0 = mod(myimage\_double,2);

imshow(c0)



To be able to get the next bit, bit 1, we have to shift it by 1 and apply mod 2. To be able to do right shift we have to divide it by 2.

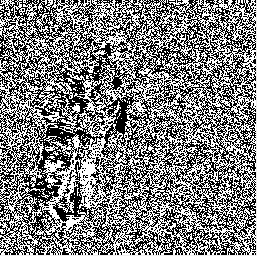
%% Section 5

% Y = floor( X ) rounds each element of X to the nearest integer less than

% or equal to that element

c1 = mod(floor(myimage\_double/2),2);

imshow(c1)



Let’s get other bits as well.

%% Section 6

c0 = mod(myimage\_double,2);

c1 = mod(floor(myimage\_double/2),2);

c2 = mod(floor(myimage\_double/4),2);

c3 = mod(floor(myimage\_double/8),2);

c4 = mod(floor(myimage\_double/16),2);

c5 = mod(floor(myimage\_double/32),2);

c6 = mod(floor(myimage\_double/64),2);

c7 = mod(floor(myimage\_double/128),2);

subplot(1,8,1); imshow(c0); title('[0]');

subplot(1,8,2); imshow(c1); title('[1]');

subplot(1,8,3); imshow(c2); title('[2]');

subplot(1,8,4); imshow(c3); title('[3]');

subplot(1,8,5); imshow(c4); title('[4]');

subplot(1,8,6); imshow(c5); title('[5]');

subplot(1,8,7); imshow(c6); title('[6]');

subplot(1,8,8); imshow(c7); title('[7]');



Let’s recreate this image again. We need to convert double into 8 bits unsigned integer (uint8).

%% Section 7

c0\_r = uint8(c0\*1);

c1\_r = uint8(c1\*2);

c2\_r = uint8(c2\*4);

c3\_r = uint8(c3\*8);

c4\_r = uint8(c4\*16);

c5\_r = uint8(c5\*32);

c6\_r = uint8(c6\*64);

c7\_r = uint8(c7\*128);

c\_recreated = c0\_r+ c1\_r+ c2\_r+ c3\_r+ c4\_r+ c5\_r+ c6\_r+ c7\_r;

imshow(c\_recreated)

