Assignment 2 (40 points)

Sebastian Zdarowski

October 6, 2019

CSC 481

From the text:

3.4(a) (4)

Propose a set of intensity slicing transformation functions capable of producing all the individual bit planes of an 8 bit monochrome image.

Reconstruction is done by multiplying the pixels of the nth plane by the constant 2^(n-1)

8 layers have 128 = (2^(8-1))

7 Layers have 64 = (2^(7-1)

6 layers have 32 = (2^(6-1)

5 layers have 16 = (2 ^ (5-1)

4 layers have 8 = (2^(4-1)

3 layers have 4 = 2^(3-1)

2 layers have 2 = 2 ^(2-1)

1 layer has 1 = 2 ^(1-1)

Odd intervals get values of 255 and even intervals get values of 0.

Function T(r) = { 255 for 2^(n) <= mod(r,2^n)

0 otherwise }

n = plane

So for example if it was seven t(r) = {255 for 2^6 <= mod(r,2^6)}

481 Students: 3.6(a) (2/0)

In general what effect would setting to zero the lower order bit planes have on the histogram of an image?

Ans: Since we are setting lower order bit planes to zero. Histogram would have wider gaps between the bars, we can't change the amount of pixels within the image, but we would lose out on high frequency details of the image.

3.34 (4)

Two images shown below are quite different, but their histograms are the same. Suppose that each image is blurred using a 3 x 3 box kernel.

A) Would the histograms of the blurred images still be equal?

Ans: No the histograms would not be equal, When using a 3x3 filter mask across the image on the left side , the pixel values wont change values until you get about halfway into the black side of the image. When working on the right image you have avg black/white pixel squares so the probabilities of the new pixel value will be much higher.

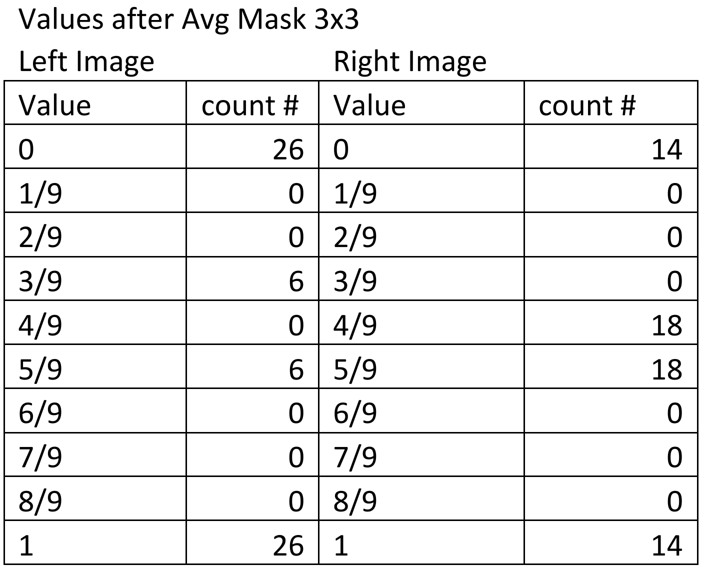
B) If your answer is no, either sketch the two histograms or give two tables detailing the histogram components.

Ans: Right image is 8 x 8, assume left one is 8 x 8 as well.

32 black pixels, 32 white pixels

Apply Avg Mask:

Histogram Table



Values after Avg Mask 3x3

Left Image Right Image

Value	count #	Value	count #
0	26	0	14
1/9	0	1/9	0
2/9	0	2/9	0
3/9	6	3/9	0
4/9	0	4/9	18
5/9	6	5/9	18
6/9	0	6/9	0
7/9	0	7/9	0
8/9	0	8/9	0
1	26	1	14

3.37 (2)

Discuss the limiting effect of repeatedly filtering an image with a 3 x 3 low pass filter kernel.

Ans: Lowpass filter will remove higher frequency features. Since it is a smoothing filter, it will replace the value of every pixel in an image by the average of the intensity levels. By doing this in repeat the method would zero or set pixel values close to 0.

481 Students: 3.60(a) (3/0)

Use the definition of fuzzy union and the membership functions in Fig 3.66 to construct the membership functions show in the following figures.

Image contains two trapezoids and one triangle union

U(z) = 0.5 – (z - a-c)/(a-b – a-c)

$1 - (z - a-b)/(a+c)$

$1 - (z-a) / (a+c)$

$.5 - (z -a+b)/ (a+c - a+b)$

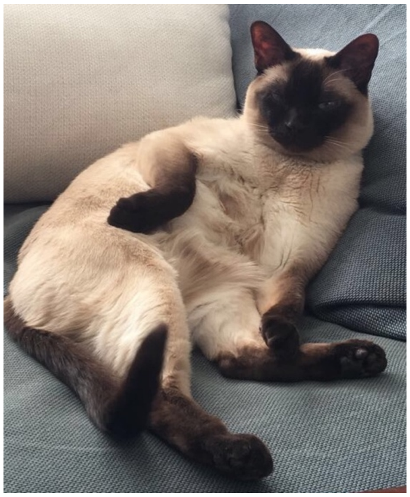
$.5 - (z - a+c) / (a+c - a-c)$

381 Students: 3.60(c) (0/5)

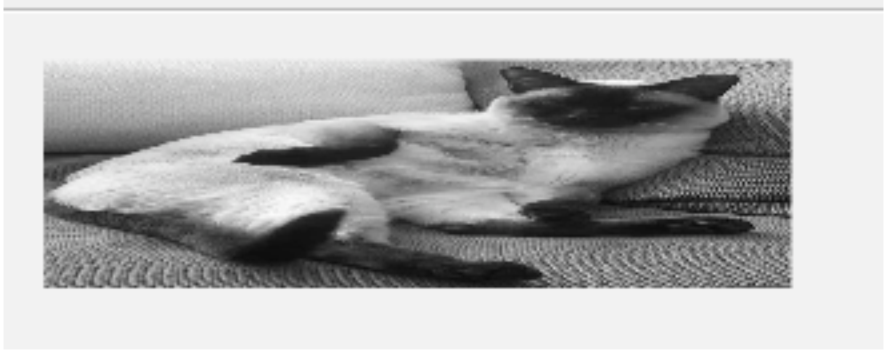
Programming:

Problem 1: *Zooming/Oversampling* and *Shrinking/Subsampling Images* by Pixel Replication (15 points)

Write your own function capable of shrinking and zooming an image by pixel replication and decimation. Assume that the desired zoom/shrink factors will be the inputs for your function and will have integer values: a negative input means shrink and a positive input means expand. Do not use MatLab built in functions (or other language libraries) for decimation and replication, although you can use other MatLab functions.

Orginal image: 

(a) Use your program to shrink an image by a factor of 4 in each dimension. Show the shrunk image.

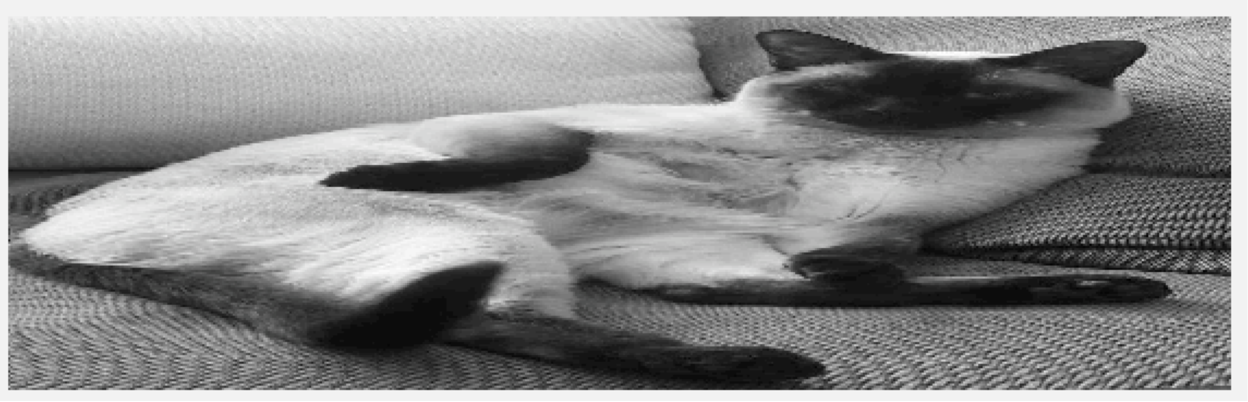


```
runshrink = imageZoom(Y,-4);


function [ output ] = imageZoom(input, change)
%a = rgb2gray(input);
[x,y] = size(input); %obtain size of image
if change > 0
    %Zoom function
    hold = []; %Hold values
    for i = 1:y
        hold = cat(2,hold,ones(1,change)*i); %combine matrix
    end
    output = input(:,hold);
    hold = []; %Hold Values
    for i = 1:x
        hold = cat(2,hold,ones(1,change)*i); %combine matrix
    end
    output = output(hold, :);
imshow(output);
elseif change < 0
    %shrink function
    z = abs(change);
    output = input([1:z:end], [1,z:end]);
imshow(output);

end
end
```

(b) Use your program to zoom the image back to its original size. Show the zoomed image and explain how and why the original image and the shrunk/zoomed images are different.



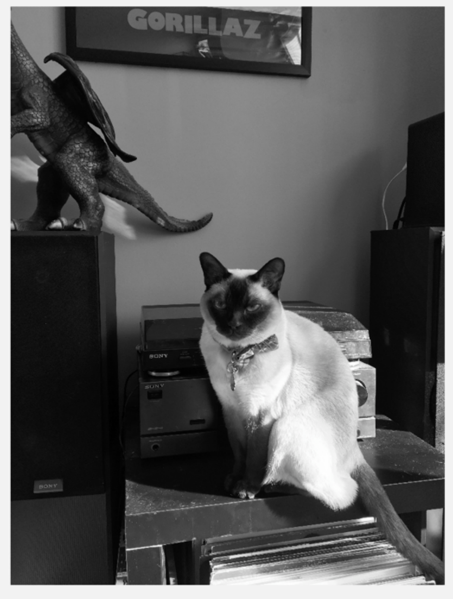
```
final = imageZoom(runshrink,4);
```

With the image shrinking we removed pixels and caused us to lose detail in the original photo. Image was shrunk literally. Smaller than original image, since we deleted rows and columns from the original picture. When applying the Zoom to try to get the image back, we lost the important detail from the shrinking and only applied a 4 factor of pixel replication, it was used the same way, but since we replicated over the left over pixels, the zoomed image will not look like the original.

Problem 2: Basic Grey Level Transformations (5 points)

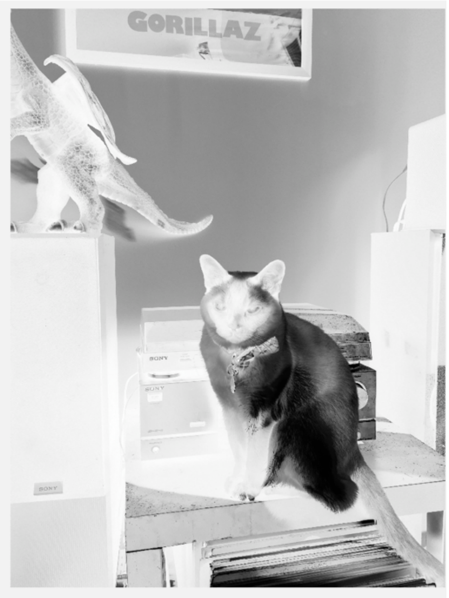
(a) Read and display an image.

```
X = imread('collins2.jpg');
Y = rgb2gray(X);
imshow(Y);
```



(b) Calculate the negative of the image and display it.

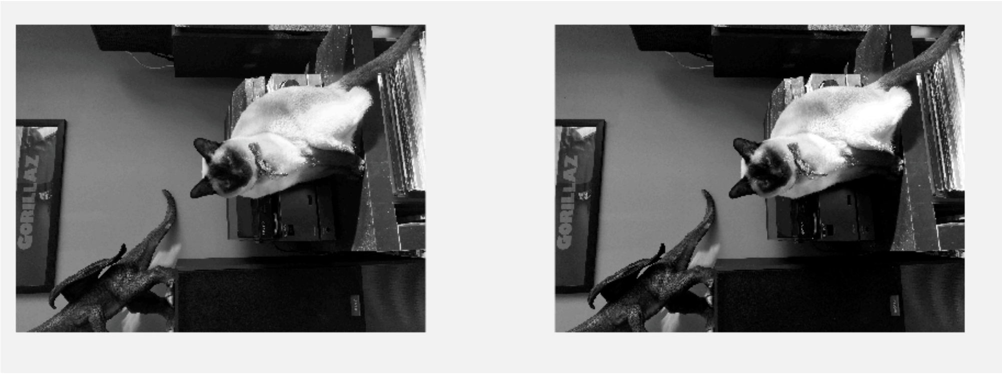
```
Z = imadjust(Y,[0 1],[1 0]);
imshow(Z);
```



(c) Perform contrast stretching using the contrast stretching technique in the section titled "Piecewise Linear Transformation Functions". You can use the imadjust.m function to perform the image transformation.

```
%Piecewise contrast strech
% Find limits to contrast strech image
A = imadjust(Y,stretchlim(Y),[]);
subplot(1, 2, 1)
imshow(Y);
subplot(1, 2, 2)

imshow(A);
```



Work by hand:

Some basic relationships between pixels (5 points) Note: You do not need to use Matlab code to solve this problem.

Consider the image segment shown below (the values in blue represent the p and q pixels):

	3	1	2	1	(q)
	2	2	0	2	
	1	2	1	1	
(p)	1	0	1	2	

(a) Let V= {0,1} and compute the lengths of the shortest 4-, 8- paths between p and q. If a particular path does not exist between these two points, explain why.

8 path distance is 4, I have shown this in the yellow highlighted numbers within the square.

4 Path does not exist in this pixel chart between p and q. q has no 4 neighbors that are 1 or 0. Cant connect p to q.

(b) Calculate the D_4 distance (city-block distance) and the D_8 distance (chessboard distance) between pixels p and q. Do these two distances depend on which path you choose between p and q? Explain your answer.

D4 Distance: Also known as Manhattan distance Sigma Sum (|xa – xb|)

P = (0,0), Q = (3,3) D4= |3 – 0 | + |3 + 0| = 6

D8 Distance Chess board distance: DChess = max (| x2 – x1| , | y2 – y1|)

Max(|3-0| , |3-0|) = 3

Distance does not depend on the path, since we are only calculating the coordinates of p and q. Coordinates are used to do the calculation.

General submission instructions:

(a) Be kind to your aging, over-worked professor and submit only a single document. This can be pdf, MS Word, OpenOffice, etc. Do not submit a zip file.

(b) Your single document should include the input image for your problem, if required, and answers to each of the sub-problems (text, image or both, as appropriate). Your document should also include code that you wrote to generate your answers.

(c) You may use any images you like for the programming; I encourage you to use images that might be useful/interesting for your final project.

(d) Feel free to use whatever functions MatLab supplies, except where otherwise specified. Also feel free to write your own, if you are so inclined; it will take more time, but you will gain a deeper understanding of the material. It is one thing, for example, to use imadjust.m and quite another to write your own contrast stretching function.

(e) Point values for each question are indicated as ***x/y*** in which ***x*** is the point value for 481 students and ***y*** is the point value for 381 students.