Ray Li
CSC 481
Dr. Furst
9/21/16

2.2 When you enter a dark theater on a bright day, it takes an appreciable interval of time before you can see well enough to find an empty seat. Which of the visual processes explained in Section 2.1 is at play in this situation?

The visual process is called brightness adaption. While the human vision can adapt a very wide range of brightness ranging from scotopic threshold to glare limit, it cannot operate over such a range simultaneously. It indicates that the range of distinct intensity levels human vision can discriminate is less than the total adaption range. Therefore, it takes some time for eyes to adjust to the brightness level in a dark theater when it is a bright day.

2.3 Although it is not shown in Fig. 2.10, alternating current certainly is part of the electromagnetic spectrum. Commercial alternating current in the United States has a frequency of 60 Hz. What is the wavelength in kilometers of this component of the spectrum?

$$\lambda = \frac{c}{v}$$

$$\lambda = \frac{\frac{2.998 \times 10^8\ m/s}{60\ Hz}}{1000} \approx 4{,}996.67\ km$$

As shown above, the wavelength of this component of the spectrum is around 4,996.67 kilometers.

2.9 A common measure of transmission for digital data is the baud rate, defined as the number of bits transmitted per second. Generally, transmission is accomplished in packets consisting of a start bit, a byte (8 bits) of information, and a stop bit. Using these facts, answer the following:

a) **How many minutes would it take to transmit a 1024 * 1024 image with 256 intensity levels using a 56K baud modem?**

$$\frac{1024^2 \times (1 + 8 + 1)}{56{,}000} \approx 187.25\ seconds \approx 3.12\ minutes$$

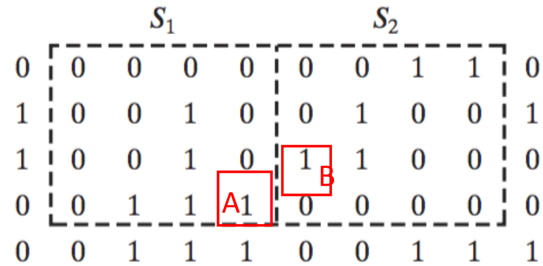
As shown above, it takes approximately 3.12 minutes to transmit a 1024x1024 image using a 56K baud modem.

b) **What would the time be at 3000K baud, a representative medium speed of a phone DSL (Digital Subscriber Line) connection?**

$$\frac{1024^2 \times (1 + 8 + 1)}{300{,}000} \approx 34.95\ seconds \approx 0.58\ minutes$$

As shown above, it takes approximately 0.58 minutes to transmit using a 3000K baud.

2.11 Consider the two image subsets, S_1 and S_2, shown in the following figure. For V = {1}, determine whether these two subsets are



a) **4-adjacent**
In these two image subsets, pixels A and B are used to determine the adjacency. Because B is not in any of four neighbors of A and neither A is in any of four neighbors of B, these two image subsets are not 4-adjacent.

b) **8-adjacent**
A is in the eight neighbors of B and so is B in eight neighbors of A. Therefore, these two image subsets are 8-adjacent.

c) **m-adjacent**
A and B are both in the diagonal neighbors of each other and at the same time, no A nor B's four neighbors is in the same class. Therefore, these two images are m-adjacent.

2.19 The median, ζ, of a set of numbers is such that half the values in the set are below ζ and the other half are above it. For example, the median of the set of values {2, 3, 8, 20, 21, 25, 31} is 20. Show that an operator that computes the median of a subimage area, S, is nonlinear.

As stated in equations 2.6-1 and 2.6-2, a linear operator can be represented as

$$H[f(x,y)] = g(x,y)$$
$$H\left[a_i f_i(x,y) + a_j f_j(x,y)\right] = a_i H[f_i(x,y)] + a_j H\left[f_j(x,y)\right]$$
$$= a_i g_i(x,y) + a_j g_j(x,y)$$

For simplicity, assuming a_i and a_j are both 1. Also we assign g_i and g_j as

$$g_i = \{1,3,2\}$$
$$g_j = \{4,6,8\}$$

The median for g_i is 2 and the median for g_j is 6. Using the linear representation, we can calculate the median of the entire subimage area as

$$H\left[a_i f_i(x,y) + a_j f_j(x,y)\right] = H[\{1,3,2\} + \{4,6,8\}]$$
$$= H[\{5,9,10\}]$$
$$= 9$$

As shown above,

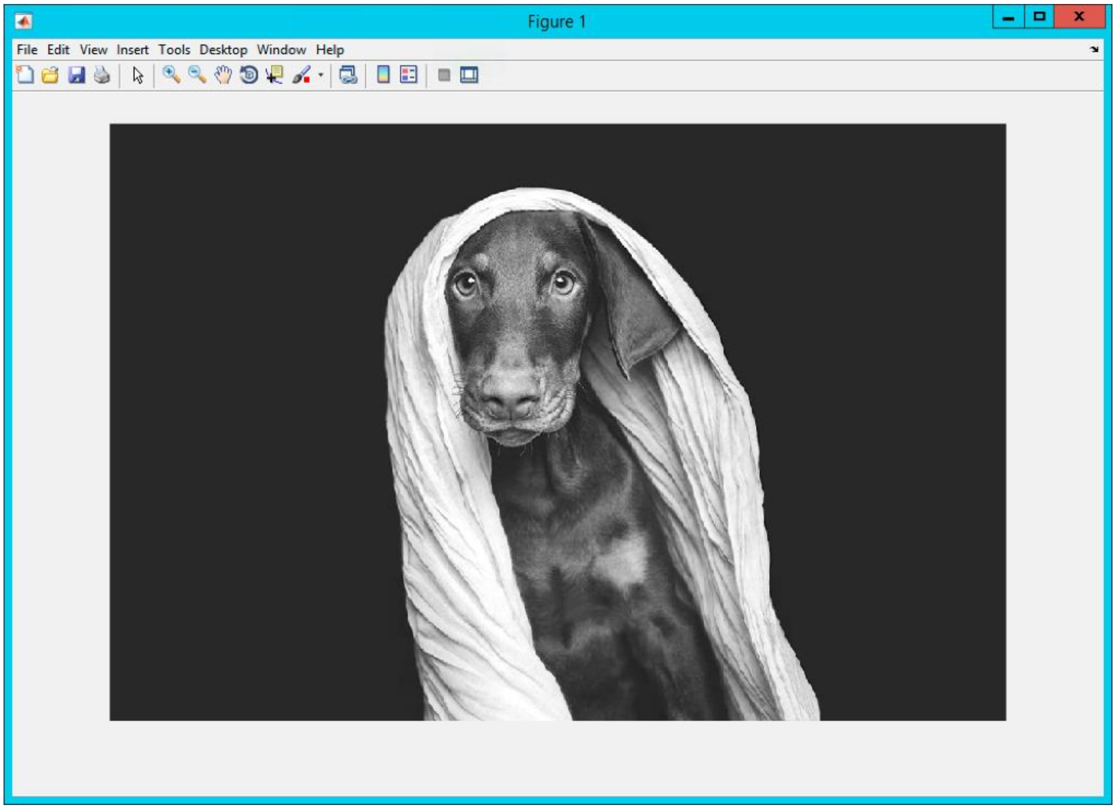
$$H\left[a_i f_i(x,y) + a_j f_j(x,y)\right] = 9$$
$$a_i g_i(x,y) + a_j g_j(x,y) = 2 + 6 = 8$$
$$H\left[a_i f_i(x,y) + a_j f_j(x,y)\right] \neq a_i g_i(x,y) + a_j g_j(x,y)$$

Therefore, the operator that computes the median of a subimage area is nonlinear.

Programming (Matlab)

1. Write a program/function that will:
   (a) Read and display an image

```
>> A = imread('dog.jpg');
>> B = rgb2gray(A);
>> imshow(B);
```



   (b) Calculate the size of the image

```
>> size(B)

ans =

   801        1200
```

   (c) Calculate the maximum pixel value

```
>> max(B(:))

ans =

  255
```

   (d) Calculate the mean pixel value

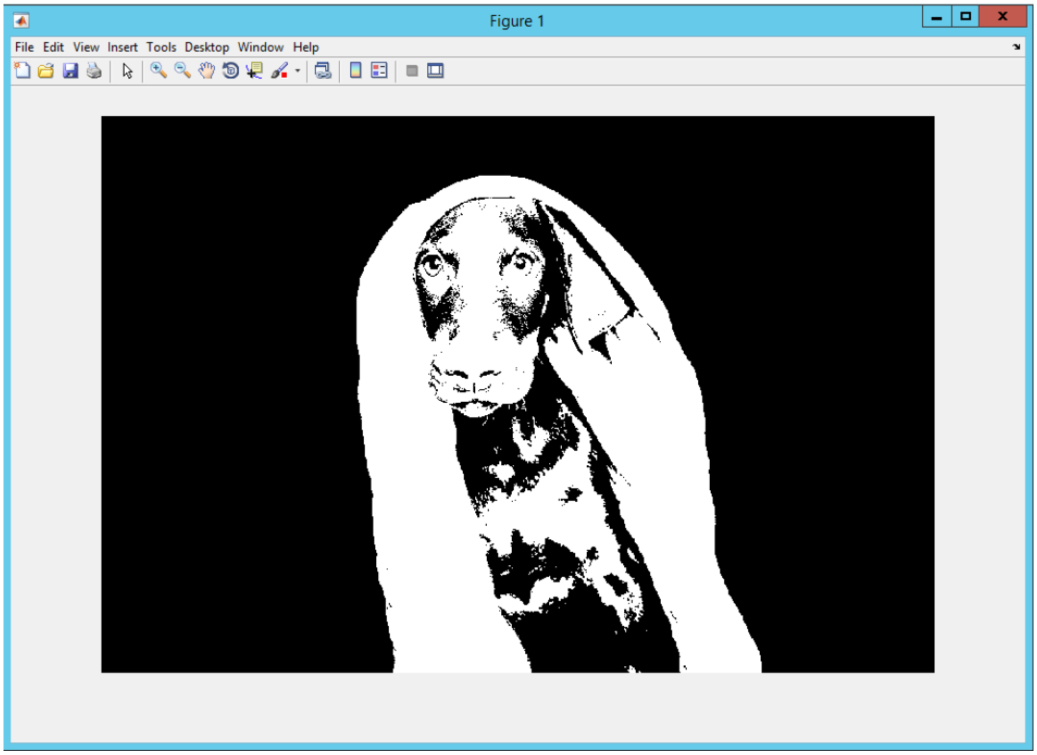
```
>> mean(B(:))

ans =

   79.8240
```

   (e) Change the pixel values of the image in the following way: all pixels' values less than the average calculated at (d) will be equal to 0 and all the others will be equal to 1. What type of image is the new generated image?

```
>> C = zeros(size(B));
>> C(B>mean(B(:))) = 1;
>> imshow(C)
```



This image is a binary image, meaning the image is made of two colors, which in this case are black and white.

2. Write a computer program to produce results comparable to Figure 2.24. That is, given an input image, reduce its spatial resolution, and then return it to its original resolution. Use all of nearest neighbor, bilinear and bicubic interpolation to do this. **481 Students (2/0):** Design your program such that the desired change in spatial resolution (e.g. 0.5, which will halve the image in each dimension, or 2.0, which will double the image in each dimension) is a variable input to your program. Show an example run of your code.

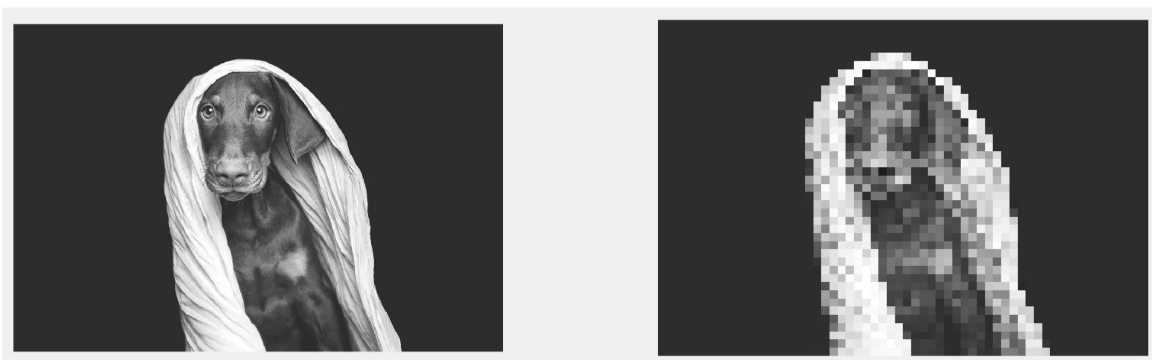
Here is the Matlab program,

```
function [ out_img ] = questionTwo(in_img, method, alpha)
g = rgb2gray(in_img);
new_img = imresize(g, double(alpha), method);
out_img = imresize(new_img, 1.0/double(alpha), method);
subplot(1, 2, 1)
imshow(in_img)
subplot(1, 2, 2)
imshow(out_img)
end
```

Here is the code to test the Matlab program,

```
>> A = imread('dog.jpg');
>> questionTwo(A, 'nearest', 0.05);
```

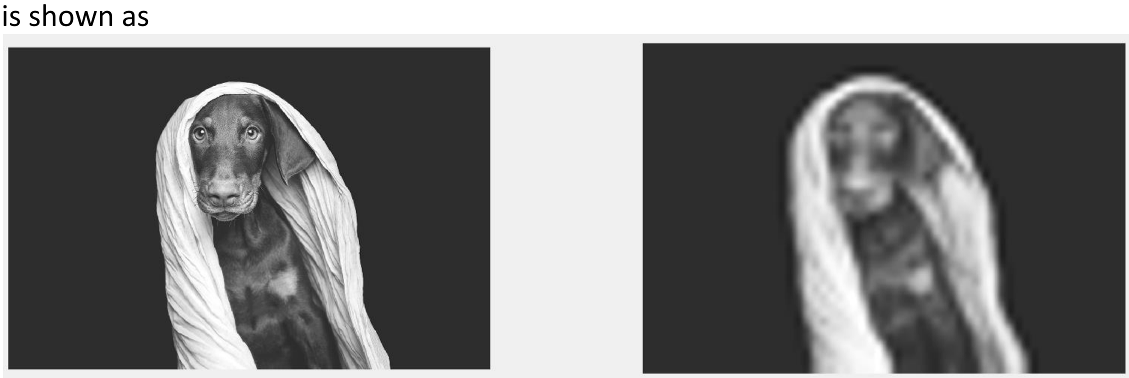
The output of the program is shown as



Here is another code example,

```
>> A = imread('dog.jpg');
>> questionTwo(A, 'bicubic', 0.05);
```

The output is shown as



3. Write a computer program capable of reducing the number of gray levels in an image from 256 to 2, in integer powers of 2. **481 Students (*2/0*):** Design your program such that the desired number of gray levels (256, 128, 64, 32, 16, 8, 4, 2) is a variable input to your program. Show an example run of your code.

Here is the program,

```
function [ out_img ] = questionThree(in_img, levels)
d = 256.0/double(levels);
g = rgb2gray(in_img);
out_img = g/double(d);
subplot(1, 2, 1)
imshow(in_img)
subplot(1, 2, 2)
imshow(out_img)
end
```

Here is the code to test it, assuming we want the gray level to be 128,

```
>> A = imread('dog.jpg');
>> questionThree(A, 128);
```

Here are the output images,

Assuming we want the gray level to be 32, we can run the following code,

```
>> A = imread('dog.jpg');
>> questionThree(A, 32);
```

Here are the output images