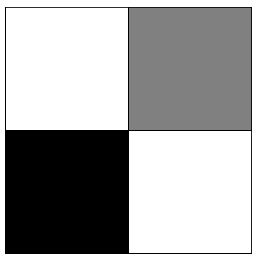
Radiography and Tomography
Lab 1: Digital Images

Grayscale Images

A grayscale image can be associated with a set of numbers representing the brightness of each pixel. For example, the 2×2 pixel image below can be associated with the corresponding array of numbers, where the black squares represent pixels with intensity 0 and the white squares represent pixels with intensity 16:



$$\sim \begin{array}{|c|c|} \hline 16 & 8 \\ \hline 0 & 16 \\ \hline \end{array}$$

Note that this interpretation gives us a way to think about multiplying an image by a constant and adding two images. For example, suppose we start with the three images (A, B, and C) below.

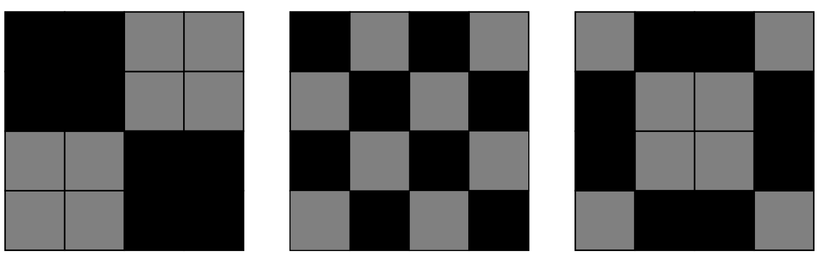


Image A Image B Image C

Then multiplying Image A by 0.5 results in Image 1 below. Note that the maximum intensity is now half what it previously was, so the all pixels have become darker gray (representing their lower intensity). Adding Image 1 to Image C results in Image 2 below; so Image 2 is created by doing arithmetic on Images A and C (try writing out the matrices corresponding to these images and checking the arithmetic).

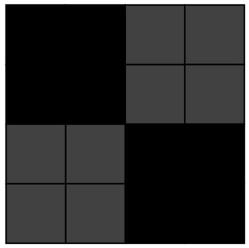


Image 1

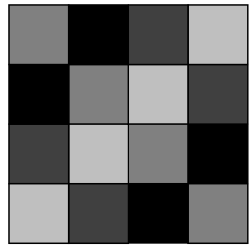


Image 2

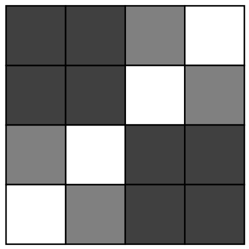


Image 3

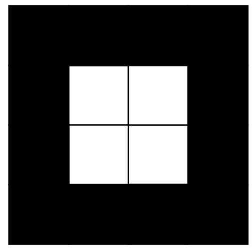


Image 4

The above examples lead us naturally to the following two definitions.

Definition: Given two images $\mathbf{x}$ and $\mathbf{y}$ with (ordered) intensity values $(x_1, x_2, \cdots, x_n)$ and $(y_1, y_2, \cdots, y_n)$, respectively, the **sum**, written $\mathbf{z} = \mathbf{x} + \mathbf{y}$ is the image with intensity values $z_i = x_i + y_i$ for all $i \in \{1, 2, \cdots, n\}$. That is, the sum of two images is the image that results by pixel-wise addition.

Definition: Given scalar a and image $\mathbf{x}$ with (ordered) intensity values $(x_1, x_2, \cdots, x_n)$, the **scalar product**, written $\mathbf{z} = a\mathbf{x}$ is the image with intensity values $z_i = ax_i$ for all $i \in \{1, 2, \cdots, n\}$. That is, scalar multiplication with an image is the image that results by pixel-wise multiplication.

Task 1: Decide whether or not Image 3 and Image 4 can be written using arithmetic operations on Images A, B, and C. In both cases either produce these arithmetic operations or justify why this can't be done.

Images in Matlab

We'll now see how to input matrices, convert them into grayscale images, and display them in Matlab. Instructions to install Matlab have been provided in a separate document. To begin, input the red lines below into the command window of Matlab, making sure to take time to understand what each line of code is doing.

Notes:
1. Ending a line with a semicolon suppresses the output (i.e. it stops Matlab from printing the result of the calculation done in that line of code). If you want to show the result of a computation, delete the semicolon at the end of its line.
2. In order to display an image, we have to know the minimum and maximum pixel intensities. The imagesc function allows the user to specify these values (this is the [0,16] in the code, indicating that we want 0 to correspond to black and 16 to correspond to white). If values outside of this range appear in the matrix, then the corresponding box will be assigned the same colour as that of the nearest number on the scale. So in this case, any number greater than 16 appears as white and any number less than 0 appears as black.
3. In this case, the scale [0,16] was chosen because of the range of values appearing in these matrices. The range could just as easily involve negative integers and does not need to include zero (for instance we could choose [-15,-2] if it made sense for the matrix entries).

```
M_A = [0 0 8 8; 0 0 8 8; 8 8 0 0; 8 8 0 0];
M_B = [0 8 0 8; 8 0 8 0; 0 8 0 8; 8 0 8 0];
M_C = [8 0 0 8; 0 8 8 0; 0 8 8 0; 8 0 0 8];
figure;
subplot(1,3,1),imagesc(M_A, [0, 16]),title('Image A');
subplot(1,3,2),imagesc(M_B, [0, 16]),title('Image B');
subplot(1,3,3),imagesc(M_C, [0, 16]),title('Image C');
```

Task 2: Enter the following lines of code one at a time and state what each line does.

```
M_A
M_1 = .5*M_A
M_2 = M_1 + M_C
figure;
subplot(1,2,1);
imagesc(M_1,[0, 16]);
title('Image 1');
subplot(1,2,2);
imagesc(M_2,[0, 16]);
title('Image 2');
colormap(gray)
```

Task 3: If you said either Image 3 or Image 4 could be obtained using arithmetic operations on images A, B and C, justify your answer by writing your own code to produce these images. Be sure to submit this code with the lab.

Task 4: Discuss with your group and answer the following questions:

1. Is it always true that adding an entirely black image to another image does not change the second image? Justify your answer.

2. We can choose any range we would like in the `imagesc` function. Why may we not want to choose a range that is (i) too small relative to the data present or (ii) too large relative to the data present? (Hint: try adjusting the range in the above code and see what the result is)

3. It is important to note that if we use two different scales, it is possible for two non-identical matrices to produce the same image. Produce two such matrices in which no black or white boxes appear.

4. In what ways are a set of images similar to a vector space? What might a zero image look like etc.?

5. In your own words and as concisely and precisely as possible **what linear algebra concepts did you learn from taking this lab?** Can you translate those concepts into the language of this lab?