Activity 2 – Calculating Change in an Image **Name_____**

The goal of this activity is to use Octave to analyze the change in one of your own personal images.

1 In Octave, load the image package and change to your working directory.

2 Click the Editor tab at the bottom of the page. This is where we can write scripts, which are basically multiple commands that can be executed together.

A useful way to begin a script is by clearing the command window, clearing the variables, and closing all open figures. On the first line, type

```
clear; clc; close all;
```

3 Make sure that the image that you want to analyze is in your working directory and read the image, storing the value as a matrix with a specific variable name.

```
myImage = imread('chimp.jpg');
```

***It is recommended to use an image with lower resolution so that the data you obtain will be easier to interpret, and will not take a long processing time. In the example, the image was 100 by 100.

4 Change the image to be in double form, and then change it to grayscale.

```
myImage = im2double(myImage);
myGrayImage = rgb2gray(myImage);
```

***Notice that as you enter these commands, the workspace has not changed. It will not execute these until you press run, which is the blue "play" triangle at the top of the screen. A shortcut is to type F5. You can do this at any time to make sure that your syntax is correct. You will have to click over to the command window to check for errors. The first time you run it, you will be prompted to save the file.

5 Use imshow to display your grayscale image. By hovering the mouse over the image you will be able to see the row and column of any location. Choose a row that you would like to analyze.

***Once you have chosen the row to use, you may not want the image to display every time that you run the program. You can delete the line with imshow, or a better solution may be to "comment it out". You can enter comments by using a percent sign at the beginning, and Octave will skip past that command. To comment out a block of code, highlight it and type ctrl-R. To remove a comment, use ctrl-shift-R.

6 Create a vector containing the values in the row that you have chosen. Recall that to isolate a single row of a matrix, you choose the row value, followed by a colon to include all columns.

```
myRow = myGrayImage(50,:);
```

7 We will now plot the values in this row. Octave has a built in "plot" command, which takes two parameters. The first is the set of input values (the x-axis) and the second is the output values (y-axis). The input will just be the integers from 1 up to the width of your image. You can find the width by looking at the workspace, or by using the length function.

```
x = 1:length(myRow);
```

8 This will be the first figure of our program, so we will create if by calling figure(1), then we can plot the values of x and y that we created.

```
figure(1);
plot(x,myRow);
```

If you run the script now, you should see a graph of the pixel values.

9 Now we will create a vector with the values of the first derivative approximation. To start with, we will make a new vector with the appropriate size, with zero at every element.

```
dmyRow = zeros(1,length(myRow));
```

Each value in the derivative is the difference between successive values from the row of the image that we are analyzing. Since each term is dependent on the next term, we will have one fewer elements in the derivative. For the sake of this exercise, we will leave the last term as zero.
To assist in our calculations, we will create a "helper vector" that has sequential values from 1 up to one less than the row length. This will be used for indexing.

```
helper = 1:length(myRow)-1;
dmyRow(helper) = myRow(helper+1) - myRow(helper);
```

10 We will make a second figure to plot the first derivative. We will repeat the process from step 7.

```
figure(2);
plot(x,dmyRow);
```

*** The *y*-axis will not show automatically. You need to make a vector of zeros that is the same size as *x* and plot it in the same graph. To do so, use the "hold on" command before the plots and "hold off" afterward.

11 Following the instructions from the previous two steps, make a vector for the values of the second derivative. Since this calculation requires the previous and subsequent term, the helper vector will begin at 2 and end at one less than the number of terms in the row.

Plot these values as well in a third figure.

12 In a fourth figure use imshow to display the row that you have been analyzing. This will be one pixel tall, so you may need to "stretch" the image some to be able to see it.

13 Now that you have observed the entire row, we will focus on a small portion to make our analysis. Look at the original image, and choose a segment that is about 20 to 30 pixels wide where there is a lot of change. If you have followed the previous step, you will only have to make a change at step 6 by replacing the colon with a range of values.

14 Complete Image Analysis Report and submit electronically. Look at sample for details.