

# Computing for Scientists - Lab 8

CS 1340 — Dr. Mihail  
Department of Computer Science  
Valdosta State University

November 13, 2018

## 1 Introduction

In this lab, you will compute and display partial image derivatives. The image derivatives (or gradients) are useful for many image processing tasks such as edge detection, maps, elevation, etc.

An image can be interpreted as a two dimensional function. Each pixel in an image has a coordinate ( $x$  and  $y$ ), and a color. The color is a 3-tuple of (red, green and blue) values, referred to as the *color channels*. For most images, each color channel is a 2D matrix of the same size as the image (e.g.: rows—height by columns—width) and of `uint8` (8 bits, or a byte) data type. Each color channel for each pixel is able to encode 256 distinct values of red, green and blue. The total number of colors is then  $256^3 = 16777216$  possible colors for a particular pixel.

You will use the MATLAB Image Processing Toolbox for this assignment. Specifically, the function `imread(filename)` will return a matrix of type `uint8` and of size  $h$  by  $w$  by 3, where  $h$  is the height of the image, and  $w$  is the width of the image. For example:

```
>> im = imread('donkey.jpg');
>> size(im)

ans =

      742      1200         3

>> whos
  Name      Size      Bytes  Class  Attributes
  im       742x1200x3    2671200  uint8
```

`im` is a 3D matrix of type `uint8`, which contains the image data. You can visualize the image using the `image(im)` command, as seen in Figure 1. Next, you will convert the image to the

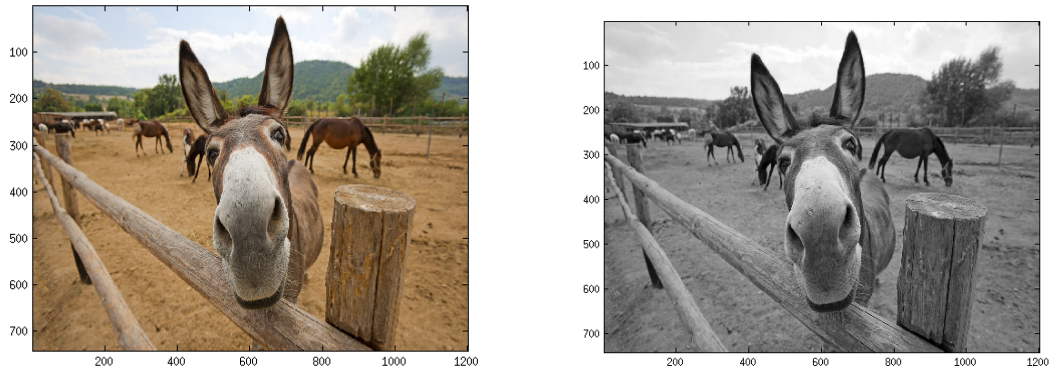


Figure 1: Left: result of `image(im)` call. Right: after conversion to grayscale.

data type `double`, using the `im2double` function, which will map each byte into the range  $[0, 1]$ , as follows: `imd = im2double(im);`

To compute the partial derivatives of a color image, you must first convert it to grayscale. This process involves a weighted average of the red, green and blue channels. Each gray pixel is compute from the  $(r,g,b)$  tuple as follows:

$$gray\_im = 0.21 * R + 0.72 * G + 0.07 * B$$

You have to extract each channel and assign it to a variable. For example:

```
>> RED = im(:, :, 1);
>> size(RED)
```

ans =

```
742      1200
```

RED is a 2D matrix. You have to compute the grayscale image using matrix operations, as shown on the right of Figure 1. **The use of Image Processing Toolbox functions to convert to grayscale is not allowed in this assignment.** After you computed the grayscale image, you are now ready to compute partial image derivatives.

## 2 Image Derivatives

The partial derivative of a continuous function  $I(x, y)$  with respect to the horizontal variable  $x$  is defined as the local slope of the plot of the function along the  $x$ -direction, or more formally, by the following limit:

$$\frac{\partial I(x, y)}{\partial x} = \lim_{\Delta x \rightarrow 0} \frac{I(x, +\Delta x, y) - I(x, y)}{\Delta x}$$

An image is not a continuous function, instead it is a function of a discrete variable, so we cannot make  $\Delta x$  arbitrarily small. Instead, we can go as small as a pixel:  $\Delta x = 1$ . This

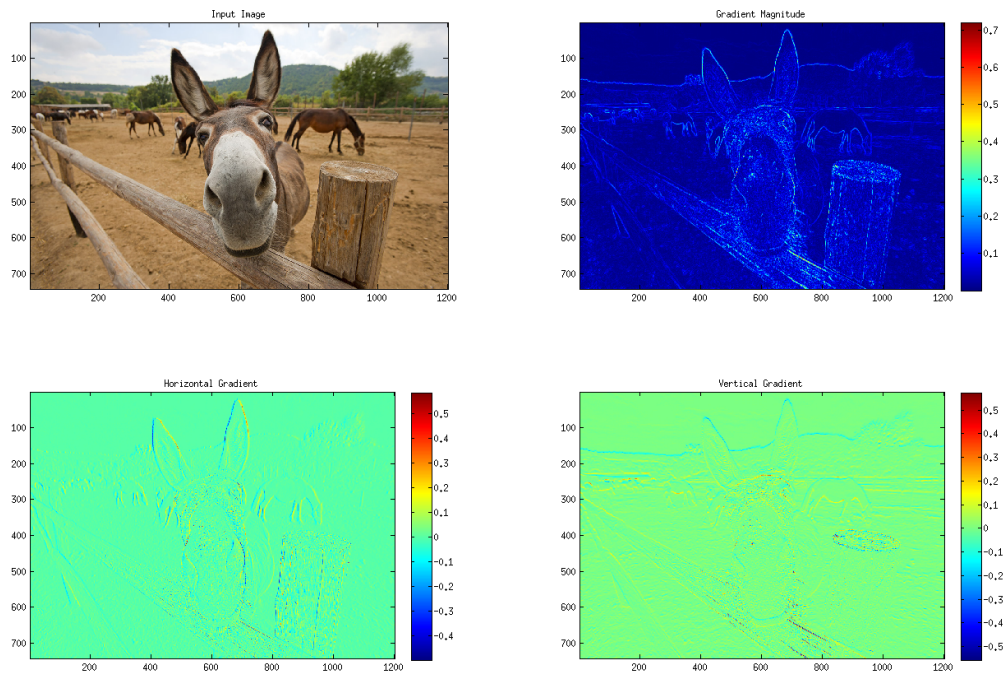


Figure 2: Final results.

results in an approximation of the true partial derivatives, computed as follows using finite differences ( $I$  is your grayscale image matrix):

$$\frac{\partial I(x, y)}{\partial x} \Big|_{x=j, y=i} \approx I(i, j + 1) - I(i, j)$$

$$\frac{\partial I(x, y)}{\partial y} \Big|_{x=j, y=i} \approx I(i + 1, j) - I(i, j)$$

Each partial derivative will be a 2D matrix. The size of the partial derivative should be equal to that of the original image. You have to use nested `for`-loops to compute the partial derivatives. **You cannot use MATLAB built-in functions to compute the partials. If you do, you will receive a zero on the assignment.**

After you computed the partial derivatives  $\frac{\partial I(x, y)}{\partial x}$  and  $\frac{\partial I(x, y)}{\partial y}$ , you will compute the gradient magnitude, with the following formula:

$$M = \sqrt{\left(\frac{\partial I(x, y)}{\partial x}\right)^2 + \left(\frac{\partial I(x, y)}{\partial y}\right)^2}$$

You will display all the results (original image, partials and magnitude) in a subplot, exactly as seen in Figure 2.

### 3 Grading Rubric

#### Program Correctness (100 points)

Input image read correctly and converted to double  
Image converted to grayscale  
Horizontal partial derivative correct (using nested for-loops)  
Vertical partial derivative correct (using nested for-loops)  
Magnitude compute correctly  
Subplot is 2 by 2  
Colorbars displayed  
Titles correct

### 4 Hints

- Do not use built-in functions that are not mentioned in this assignment prompt.
- Pay close attention to the for-loop indices in each direction (horizontal and vertical).
- Use `zeros` to initialize your partial derivative matrices.
- Use `imagesc` to display partials and magnitude.

### Due dates

1. Code: due Friday, November 16th, before midnight.