EEEE 709 – Advanced Engineering Mathematics
MATLAB Project #2 Problem 2

Author: Aharon Sebton
Affiliation: Student
Email: ams9265@rit.edu
Phone: 484-680-0143

Date: 10/14/2021

Problem 2: This problem serves to visually demonstrate the operations of horizontal/vertical gradients and laplacians.

Create a script in MATLAB to perform the following tasks:

a) Read in the grayscale image named "cameraman.tif" which is available in MATLAB and display it.
b) Compute the horizontal gradient of the image by convolving it with the 3x3 kernel Gx shown below and display the output. The Gx kernel represents a first difference approximation to the partial derivative $\frac{\partial f}{\partial x}$.
c) Compute the vertical gradient of the image by convolving it with the 3x3 kernel Gy shown below and display the output. The Gy kernel represents a first difference approximation to the partial derivative $\frac{\partial f}{\partial y}$.
d) Compute the magnitude of the gradient and display it.
e) Compute the Laplacian of the image by convolving it with the 3x3 kernel H shown below and display your results. The 3x3 kernel H given below provides a second difference approximation to the Laplacian operator defined as $\nabla^2 f(x,y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$.
f) Provide an explanation of your observations on the above steps in your report and display all images. Clearly label each image/figure.
g) Read in the color image named "peppers.png" which is available in MATLAB and convert it to grayscale colormap [hint: use rgb2gray()]. Display the color and grayscale images side by side.
h) Repeat steps (b) through (e) on the grayscale version of "peppers.png" and provide the results. Did the gradient computation performed on the grayscale image properly account for all of the edges in the color image? Elaborate on this point. Under what conditions would the gradient computation of edges on the grayscale version not account for the color edges? Can you illustrate this in MATLAB with a properly structured example?

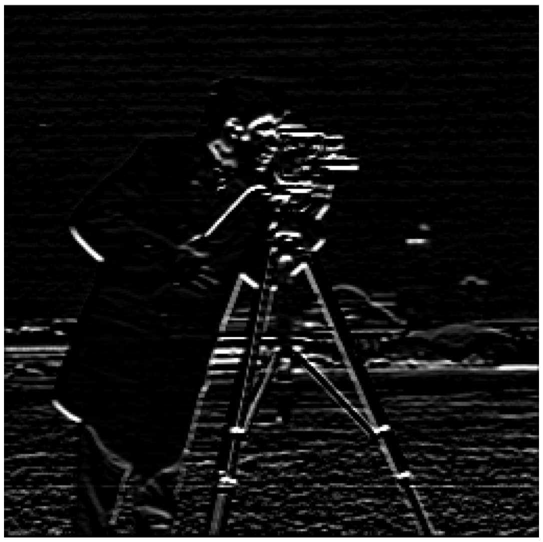
Note: In performing the convolutions, constraint the output to the same size as the input [hint: imfilter() may be useful to perform the above operations]. You may need to rescale the output to display it or use MATLAB to automatically do the proper rescaling. The following functions in MATLAB may be useful: imread, imfilter, rgb2gray and imshow.

$$G_x = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \qquad G_y = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \qquad H = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

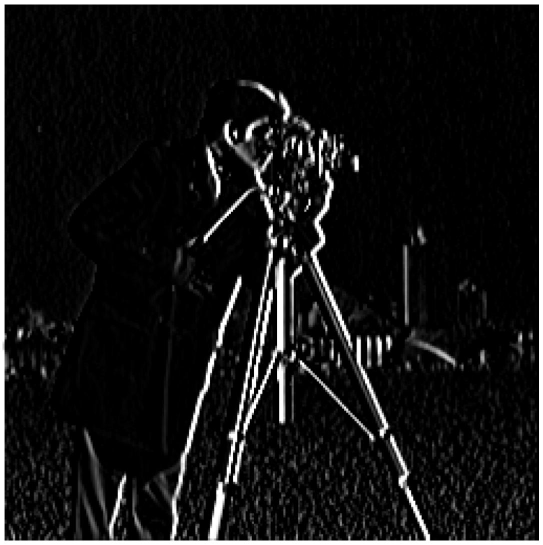
Problem 2 Solution:

a) **Figure 1: Original cameraman.tif Grayscale Image**

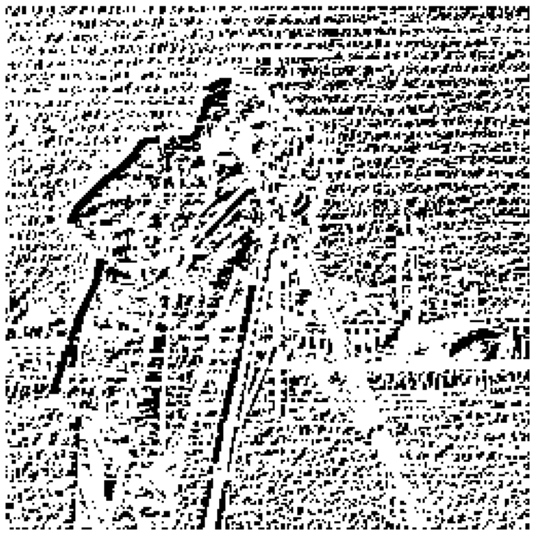
b) Figure 2: Horizontal Gradient of Image



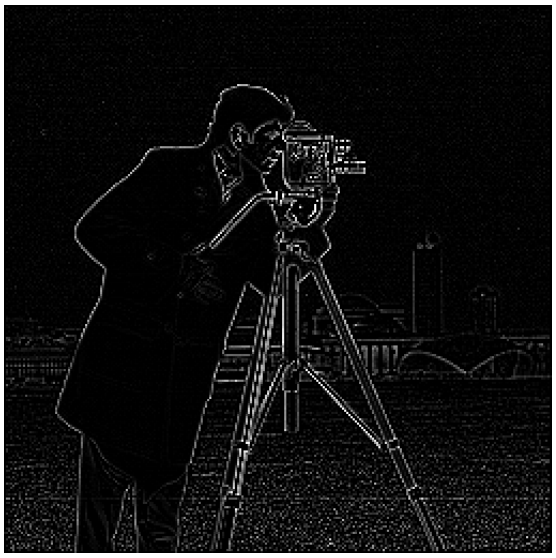
c) Figure 3: Vertical Gradient of Image



d) Figure 4: Magnitude of Gradient of Image

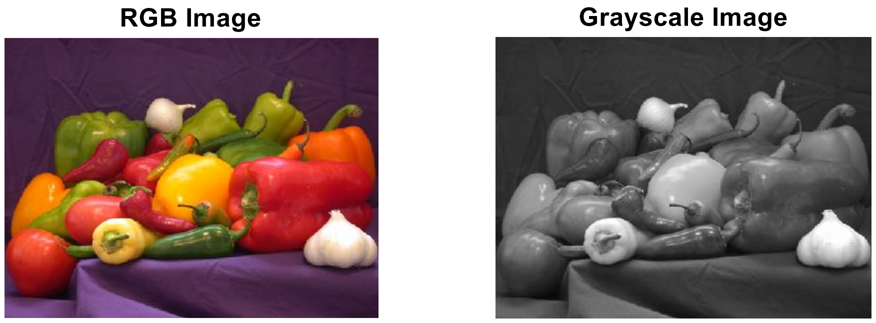


e) Figure 5: Laclacian of Image



f) The horizontal gradient image shows the rate at which pixel values change when sifting across the original image in the horizontal (x) direction. The vertical gradient image shows the rate at which pixel values change when sifting across the original image in the vertical (y) direction. The magnitude of the gradient image shows the maximum rate at which pixel values change in any given direction. The Laplacian image shows the rate at which the gradients change in any given direction, hence the edge detection capabilities of the Laplacian operation.

g) Figure 6: peppers.png RGB Image vs. Grayscale Image



h) Figure 7: Horizontal Gradient of Image

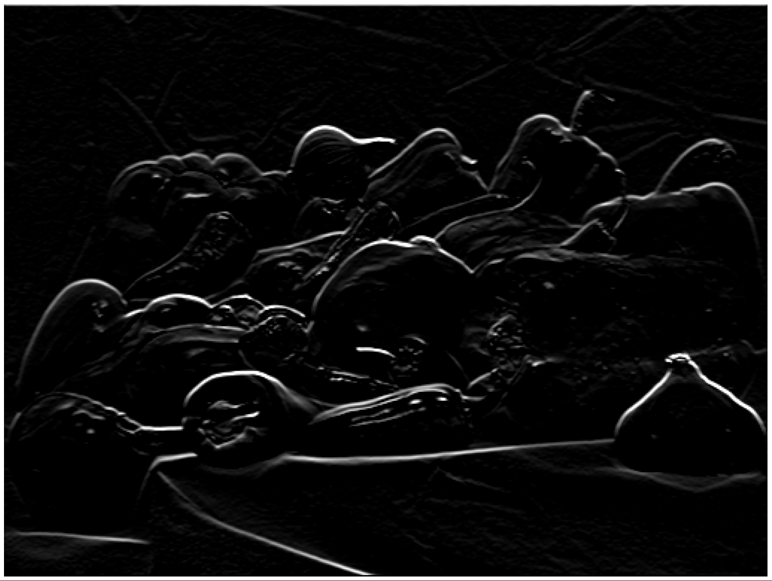


Figure 8: Vertical Gradient of Image

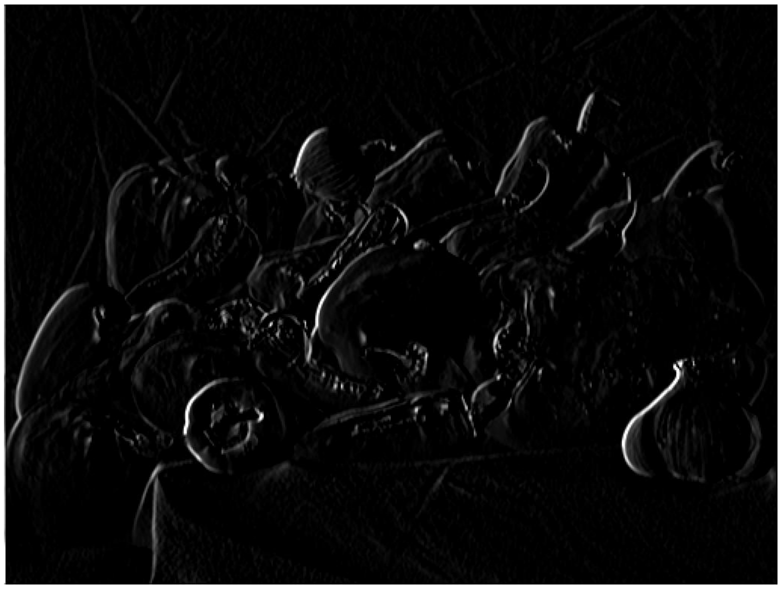


Figure 9: Magnitude of Gradient of Image

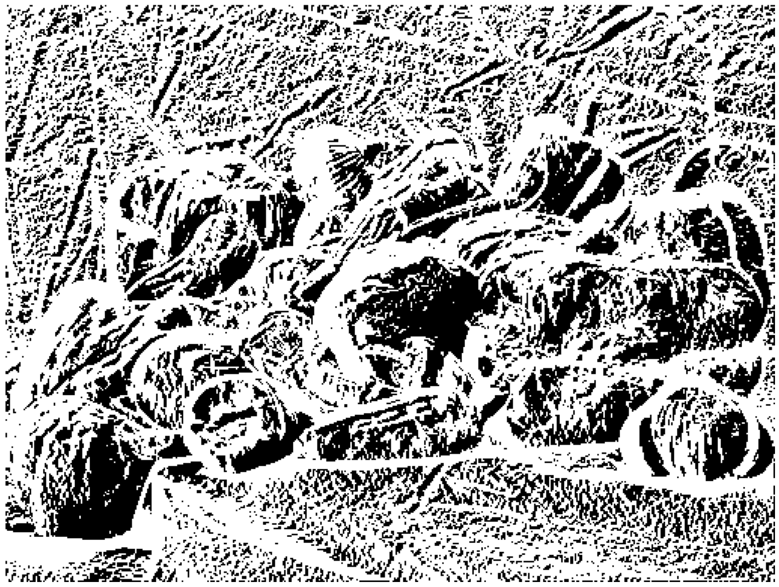
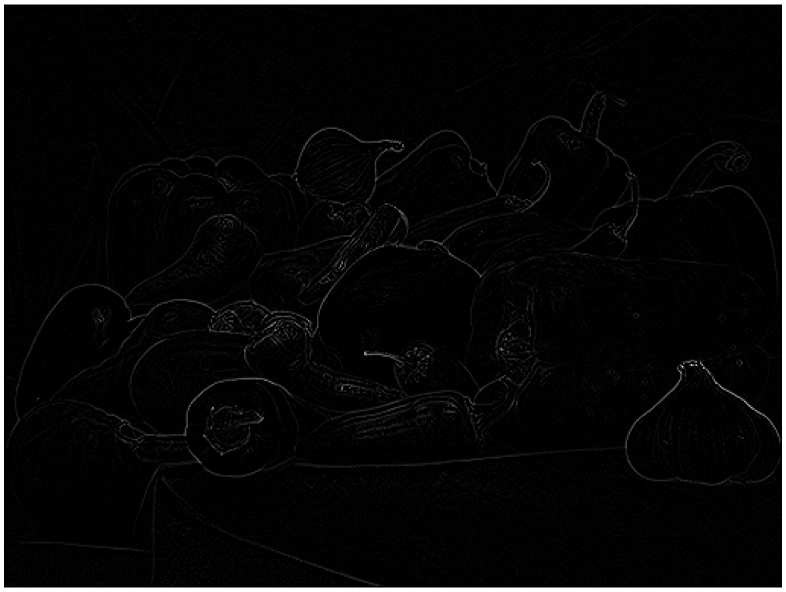


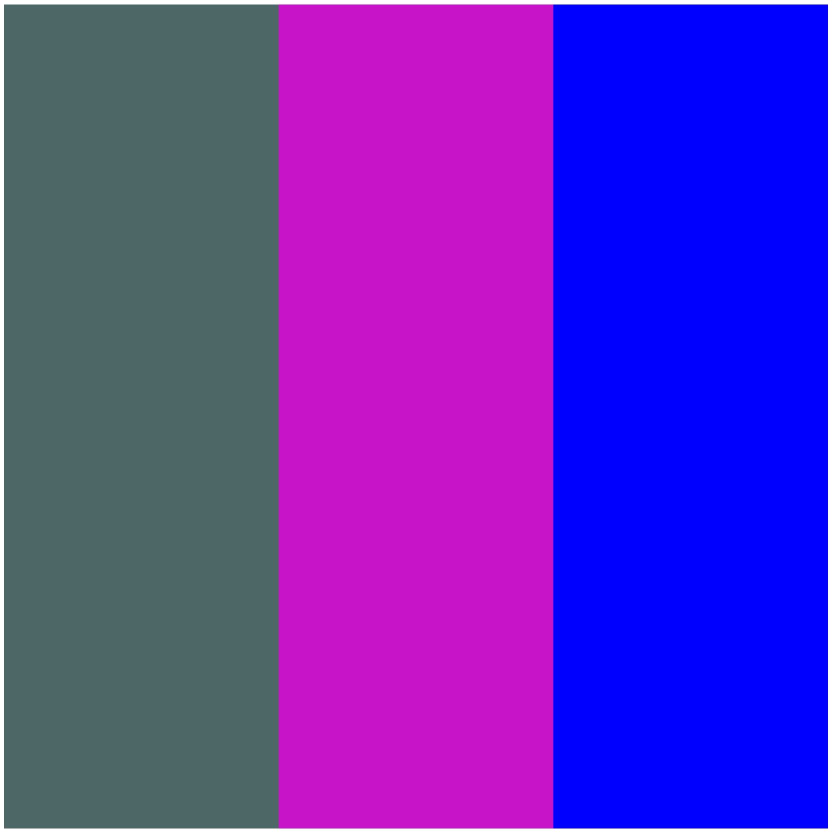
Figure 10: Laclacian of Image



The gradient computation performed on the grayscale image did not properly account for all of the edges in the color image. This is because the rgb2gray function (and any other conversion algorithm, for that matter) maps 256^3 RGB combinations to only 256 grayscale values. Since this is far from a one-to-one mapping, there are plenty of unique colors that have the same grayscale value post-conversion. The Laplacian of a grayscale image will not detect a change in intensity between two pixels with the same grayscale value, so the Laplacian of a grayscale image cannot detect edges between colors that map to the same grayscale value.

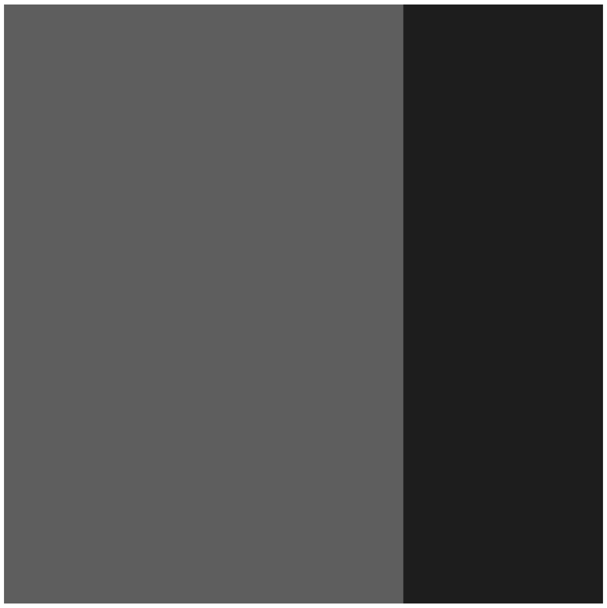
In order to prove this phenomenon, an arbitrary RGB combination was chosen and displayed in the left column of Figure 11 below. The rgb2gray algorithm found in MATLAB's documentation was then used to compute a new RGB combination that, when transformed by the rgb2gray function, should map to the same grayscale value. That computed RGB combination is displayed in the middle column of Figure 11 below. Finally, a column of pure blue was displayed in the right column of Figure 11 below as an experimental control.

Figure 11: Constructed Image with Chosen RGB1, Computed RGB2, and Pure Blue for Control



After converting the entire constructed image in Figure 11 to grayscale using the MATLAB rgb2gray function, Figure 12 shows that the left and middle columns mapped to the same grayscale value as expected. The pure blue column mapped to a darker grayscale value than the other two RGB combinations.

Figure 12: Grayscale Image Derived from Figure 11 RGB Image



Computations of the magnitude of the gradient of the grayscale image (shown in Figure 13 below) and the Laplacian of the grayscale image (shown in Figure 14 below) prove that the color edge between the left and middle columns is no longer detectable. The color edge between the middle and right columns is detectable by both computations, as expected.

Figure 13: Magnitude of Gradient of Image

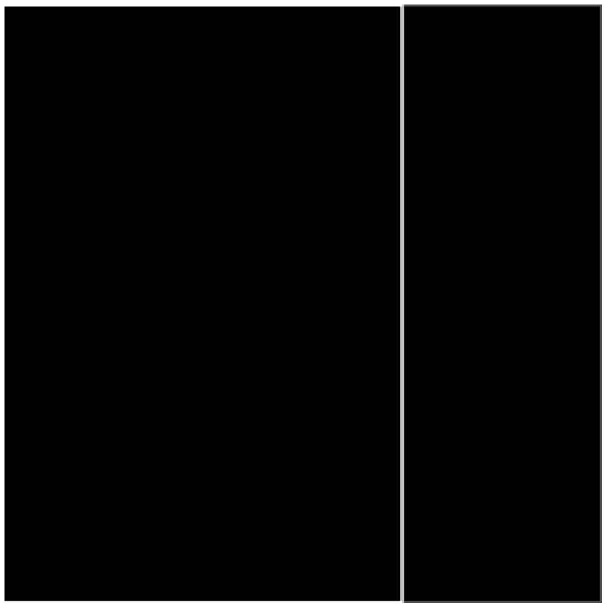
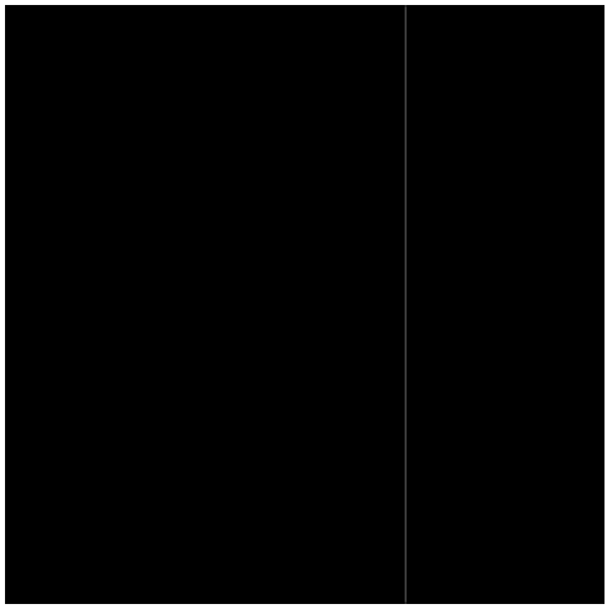


Figure 14: Laplacian of Image



Appendix 2: MATLAB Code for Problem 2

```
%% Aharon Sebton - Advanced Engineering Mathematics Project 2 - Problem 2
%% Problem 2 - Part a
clear all; clc;                                   % Clear all variables and command
window
I=imread('cameraman.tif');                        % Store image from graphics file as
matrix of grayscale values
figure(1)                                         % Create new figure window
imshow(I)                                         % Display grayscale image
title('Figure 1: Original cameraman.tif Grayscale Image')    % Give the image a title
%% Problem 2 - Part b
Gx=[-1 -1 -1;0 0 0;1 1 1];                        % Define Gx kernel
PDx=imfilter(I,Gx);                               % Compute horizontal gradient of
image (partial derivative wrt x) by convolving with Gx
figure(2)                                         % Create new figure window
imshow(PDx)                                       % Display horizontal gradient
title('Figure 2: Horizontal Gradient of Image')   % Give the image a title
%% Problem 2 - Part c
Gy=[-1 0 1;-1 0 1;-1 0 1];                        % Define Gy kernel
PDy=imfilter(I,Gy);                               % Compute vertical gradient of image
(partial derivative wrt y) by convolving with Gy
figure(3)                                         % Create new figure window
imshow(PDy)                                       % Display vertical gradient
title('Figure 3: Vertical Gradient of Image')     % Give the image a title
%% Problem 2 - Part d
magdel=sqrt(double(PDx).^2+double(PDy).^2);       % Compute magnitude of the gradient
figure(4)                                         % Create new figure window
imshow(magdel)                                    % Display magnitude of the gradient
title('Figure 4: Magnitude of Gradient of Image') % Give the image a title
%% Problem 2 - Part e
H=[0 1 0;1 -4 1;0 1 0];                           % Define H kernel
lap=imfilter(I,H);                                % Compute laplacian of image by
convolving with H
figure(5)                                         % Create new figure window
imshow(lap)                                       % Display laplacian of the image
title('Figure 5: Laclacian of Image')             % Give the image a title
%% Problem 2 - Part g
I=imread('peppers.png');                          % Store image from graphics file as
matrix of RGB values
Ig=rgb2gray(I);                                   % Convert RGB image matrix to matrix
of grayscale values and store
figure(6)                                         % Create new figure window
subplot(1,2,1)                                    % Format figure as side-by-side
comparison between two images and select left side
imshow(I)                                         % Display RGB image on the left
title('RGB Image')                                % Give the RGB image a title
subplot(1,2,2)                                    % Select right side of comparison
figure
imshow(Ig)                                        % Display grayscale image on the
right
title('Grayscale Image')                          % Give the grayscale image a title
sgtitle('Figure 6: peppers.png RGB Image vs. Grayscale Image','FontWeight','bold')    %
Give the overall image a title
%% Problem 2 - Part h
PDx=imfilter(Ig,Gx);                              % Compute horizontal gradient of
image (partial derivative wrt x) by convolving with Gx
figure(7)                                         % Create new figure window
imshow(PDx)                                       % Display horizontal gradient
title('Figure 7: Horizontal Gradient of Image')   % Give the image a title
```

```
PDy=imfilter(Ig,Gy);                                  % Compute vertical gradient of image
(partial derivative wrt y) by convolving with Gy
figure(8)                                             % Create new figure window
imshow(PDy)                                           % Display vertical gradient
title('Figure 8: Vertical Gradient of Image')         % Give the image a title
magdel=sqrt(double(PDx).^2+double(PDy).^2);           % Compute magnitude of the gradient
figure(9)                                             % Create new figure window
imshow(magdel)                                        % Display magnitude of the gradient
title('Figure 9: Magnitude of Gradient of Image')     % Give the image a title
lap=imfilter(Ig,H);                                   % Compute laplacian of image by
convolving with H
figure(10)                                            % Create new figure window
imshow(lap)                                           % Display laplacian of the image
title('Figure 10: Laplacian of Image')                % Give the image a title

image=zeros(300,300,3);                               % Create blank image
RGB1=[0.3 0.4 0.4];                                   % Choose arbitrary RGB color values
between 0 and 1

% MATLAB rgb2gray documentation states the following:

% rgb2gray converts RGB values to grayscale values by forming a weighted sum of the R, G,
and B components:
% 0.2989 * R + 0.5870 * G + 0.1140 * B

% To find another RGB value combination with the same grayscale value, use rgb2gray
algorithm equation

RGB2=[RGB1(2)*0.587/0.2989 RGB1(3)*0.114/0.587 RGB1(1)*0.2989/0.114]; % Compute new R
from old G, new G from old B, and new B from old R

image(:,1:100,1)=RGB1(1);                             % Add R portion of first color
(arbitrarily chosen) to image
image(:,1:100,2)=RGB1(2);                             % Add G portion of first color
(arbitrarily chosen) to image
image(:,1:100,3)=RGB1(3);                             % Add B portion of first color
(arbitrarily chosen) to image

image(:,101:200,1)=RGB2(1);                           % Add R portion of new color
(carefully computed) to image
image(:,101:200,2)=RGB2(2);                           % Add G portion of new color
(carefully computed) to image
image(:,101:200,3)=RGB2(3);                           % Add B portion of new color
(carefully computed) to image

image(:,201:300,3)=1;                                 % Add 100% blue to image as a control

figure(11)                                            % Create new figure window
imshow(image)                                         % Display RGB image
title('Figure 11: Constructed Image with Chosen RGB1, Computed RGB2, and Pure Blue for
Control')    % Give the image a title
g=rgb2gray(image);                                    % Convert RGB image matrix to matrix
of grayscale values and store
figure(12)                                            % Create new figure window
imshow(g)                                             % Display grayscale image
title('Figure 12: Grayscale Image Derived from Figure 11 RGB Image')    % Give the image
a title
PDx=imfilter(g,Gx);                                   % Compute horizontal gradient of
image (partial derivative wrt x) by convolving with Gx
```

```
PDy=imfilter(g,Gy);                              % Compute vertical gradient of image
(partial derivative wrt y) by convolving with Gy
magdel=sqrt(double(PDx).^2+double(PDy).^2);      % Compute magnitude of the gradient
figure(13)                                       % Create new figure window
imshow(magdel)                                   % Display magnitude of the gradient
title('Figure 13: Magnitude of Gradient of Image')  % Give the image a title
lap=imfilter(g,H);                               % Compute laplacian of image by
convolving with H
figure(14)                                       % Create new figure window
imshow(lap)                                      % Display laplacian of the image
title('Figure 14: Laplacian of Image')           % Give the image a title
```