CS-512 Assignment 2: Report

Gady Agam
Department of Computer Science
Illinois Institute of Technology

Abstract

This is a report for Assignment 2 for CS512. The task given at hand is to perform single image manipulation corresponding to specified keys that must modify the image being read accordingly. The report talks about the implementation of the problem statement and the correctness of the results obtained.

1. Problem Statement

The program must read an image from an input file and must perform tasks corresponding to the keys on the keyboard mentioned below:

- 'i' - reload the original image (i.e cancel any previous processing)
- 'w' - save the current (possibly processed) image into the file 'out.jpg'
- 'g' - convert the image to grayscale using the openCV conversion function.
- 'G' - convert the image to grayscale using your implementation of conversion function.
- 'c' - cycle through the color channels of the image showing a different color channel every time the key is pressed.
- 's' - convert the image to grayscale and smooth it using the openCV function. Use a track bar to control the amount of smoothing.
- 'S' - convert the image to grayscale and smooth it using your function which should perform convolution with a suitable filter. Use a track bar to control the moth of smoothing.
- 'd' - downsample the image by a factor of 2 without smoothing.
- 'D' - downsample the image by a factor of 2 with smoothing
- 'x' - convert the image to grayscale and perform convolution with an x derivative filter. Normalize the obtained values to the range [0,255].
- y' - convert the image to grayscale and perform convolution with a y derivative filter. Normalize the obtained values to the range [0,255].
- 'm' - show the magnitude of the gradient normalized to the range [0,255]. The gradient is computed based on the x and y derivatives of the image.
- 'p' - convert the image to grayscale and plot the gradient vectors of the image every N pixels and let the plotted gradient vectors have length K. Use a track bar to control N. Plot the vectors as short line segments of length K.
- 'r' - convert the image to grayscale and rotate it using angle theta. Use a track bar to control the rotation angle.
- 'h' - Display a description of the program and the keys it uses to run.

2. Proposed Solution

For solving this problem statement an event loop needs to be created that takes the character input from the user to perform the corresponding functions on the input image.

3. Implementation Details

Steps to run the program:

- The program must be run with an input file name on the command line.
- Once the program runs press the keys to perform corresponding functions
- Enter 'h' on the keyboard to get a help menu that guides which key performs which function and to terminate the program use the key 'esc' on the keyboard

For key 'i' :
The original image is reloaded to cancel out any previous processing done on the image. The image displayed is resized.

For key 'w' :
The image at any stage of the program i.e with processing can be stored in the current working directory as 'out.jpg'.

For key 'g' :
The original image is converted to grayscale using the OpenCV function cvtColor().

For key 'G' :
The original image is converted to grayscale by the function RGB2Grey(image). This function takes the image as it's parameter. The function splits the input image into it's R, G, B color channels and takes a weighted average of each R, G, B pixel value to produce a grayscale pixel value.

For key 'c' :
The original image split into it's R, G, B color channels by using the OpenCV function split. And each time the key is pressed it displays each color channel of the input image.

For key 's' :
The original image is first converted to grayscale using the OpenCV function cvtColor(), then a trackbar is created which uses the function smoothingSlider() to smooth according to trackbar. In the smoothingSlider() function, the position of the trackbar is captured using OpenCV function getTrackbarPos() and the smoothing is done by the OpenCV function blur().

For key 'S' :
The original image is first converted to grayscale using the OpenCV function cvtColor(), then a trackbar is created which uses the function smoothingSlide2r() to smooth according to trackbar. In the smoothingSlider2() function, the position of the trackbar is captured using OpenCV function getTrackbarPos() and the smoothing is done by the function convolve(image,kernel) which takes the image and the kernel used as parameters. In this function an area of interest of the image is captured

which is similar to the kernel dimensions and then this area of the image and the kernel are convolved and the result is stored in an output image.

For key ‘d’ :
The original image is down sampled by a factor of 2 by the function downsample(image) which takes the image as a parameter. This function returns an image by throwing every second pixel from it’s rows and columns.

For key ‘D’ :
The original image is smoothed and down sampled by a factor of 2 by the OpenCV function pyrDown() which implements a Gaussian image pyramid that smooths the image first and then down samples the image by a factor of 2.

For key ‘x’ :
The original image is first converted to grayscale using the OpenCV function cvtColor(), then a sobelX filter is defined using numpy arrays. This filter is then convolved with the image using OpenCv function filter2D() to get the X-derivatives of the image. The obtained values are then normalized on the range [0,255] and displayed.

For key ‘y’ :
The original image is first converted to grayscale using the OpenCV function cvtColor(), then a sobelY filter is defined using numpy arrays. This filter is then convolved with the image using OpenCv function filter2D() to get the Y-derivatives of the image. The obtained values are then normalized on the range [0,255] and displayed.

For key ‘m’ :
The original image’s gradient magnitude is displayed using the function magnitude(image) which takes the image as a parameter. In this function, first the X and Y derivatives of the image are taken by convolving the image with the sobelX and sobelY filters specified respectively using the OpenCV function filter2D(). Once the X and Y derivatives are found, a loop is used to loop over the image and take the magnitude of the gradient at each pixel using it’s respective X and Y derivative value. This is stored in an output image which is then displayed.

For key ‘p’ :
The original image is first converted to grayscale using the OpenCV function cvtColor(), then the X and Y derivatives are take using the sobel() function from OpenCV. Then, the gradient vectors are plotted using the quiver() function from matplotlib and displayed.

For key ‘r’ :
The original image is first converted to grayscale using the OpenCV function cvtColor(), then the angle of the image rotation is controlled using a trackbar. This trackbar uses the function rotationSlider() to perform image rotation. In the rotationSlider() function, the angle of rotation from the trackbar is captured and then the OpenCV function getRotationMatrix2D() is used to obtain the corresponding rotation matrix. Finally, the OpenCV function wrapAffine is used to apply the rotation matrix to the image and also perform an inverse mapping to avoid any holes in the image.

For key ‘h’ :
The help menu for the program is displayed on the terminal. It describes the function each key performs on the input image given on the command line.

4. Results and Discussion

Once the program has executed, the following are the results obtained on the input image on the various input commands by the user

Original image:

For key ‘g’:

For key ‘G’:

For key ‘c’:

Red channel

Green channel

Blue channel

For key ‘s’:

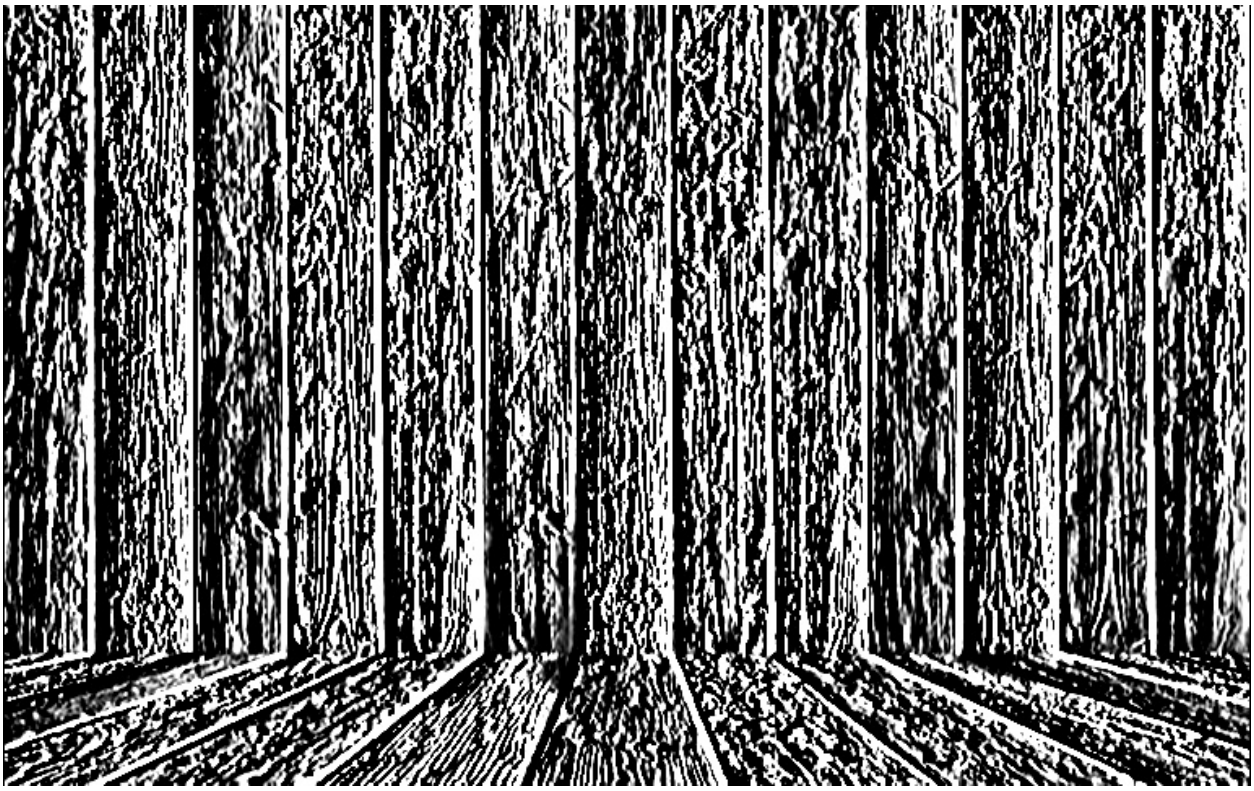
Smmothing level 2

Smoothing level 8

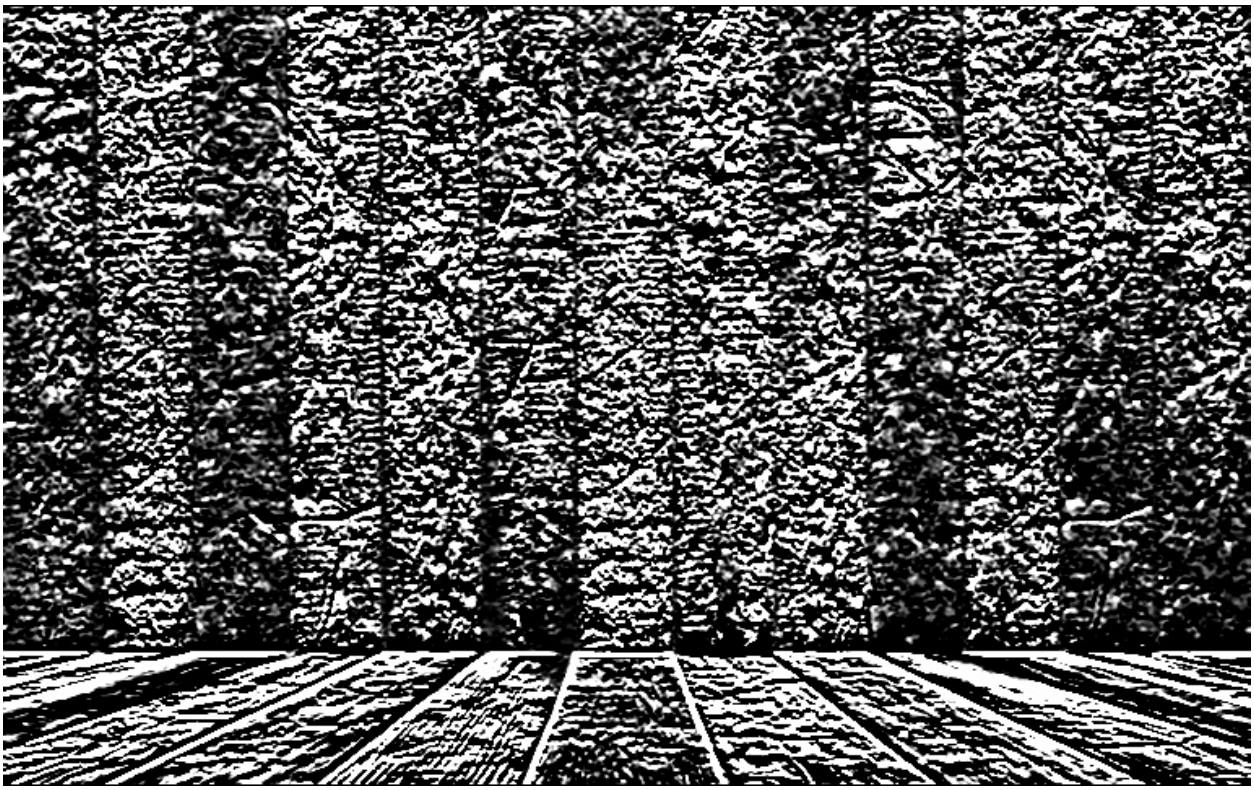
For key ‘d’:

For key ‘D’:

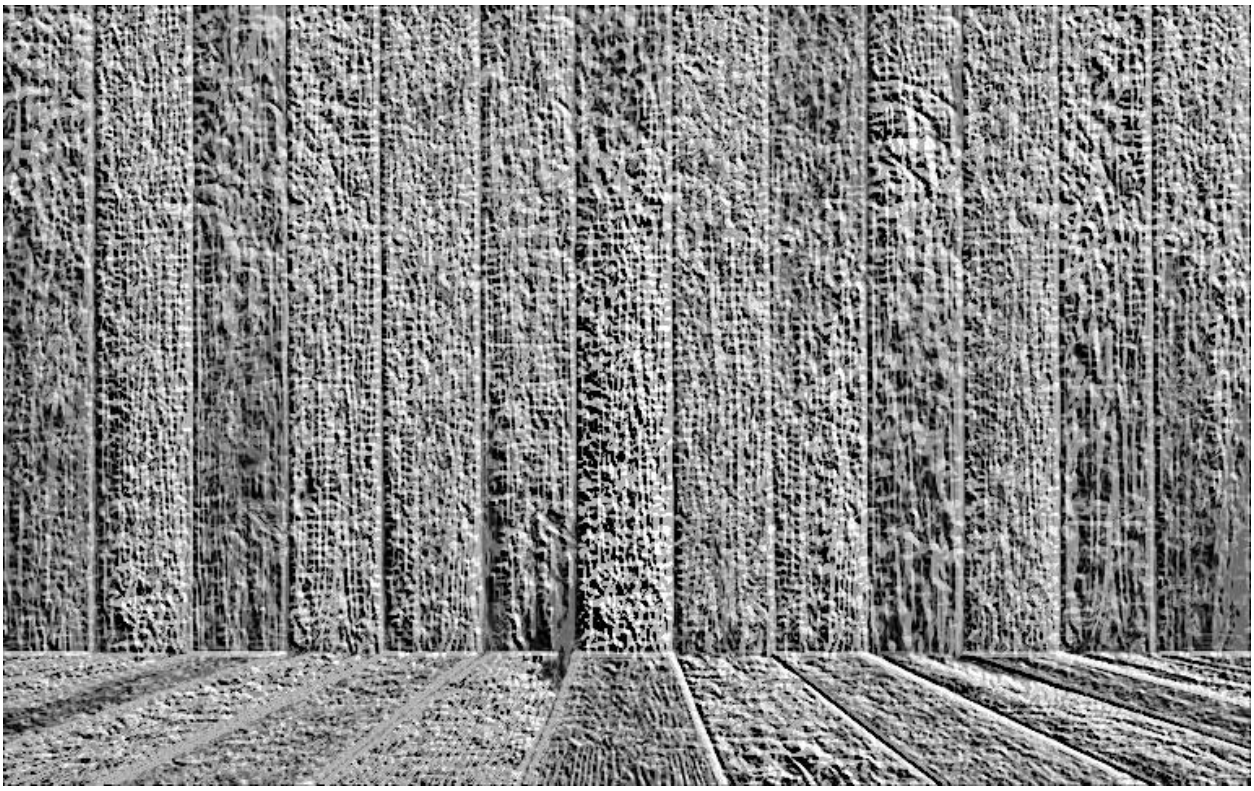
For key ‘x’:



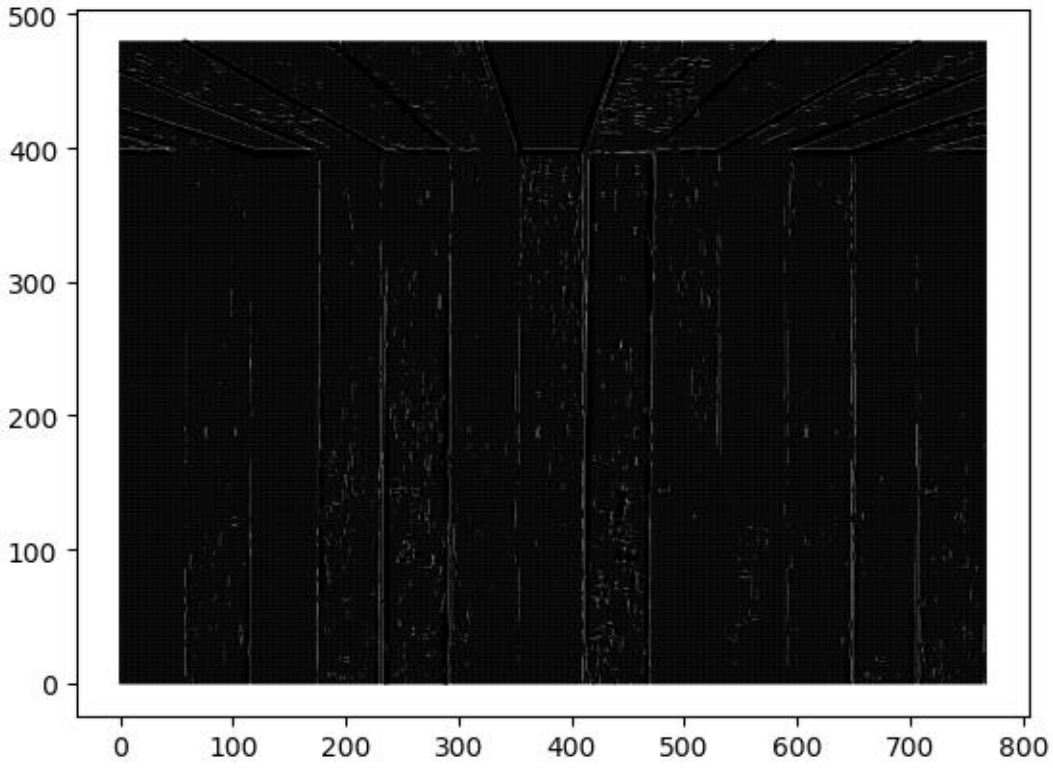
For key ‘y’:



For key ‘m’:



For key ‘p’:



For key ‘r’:

For key ‘h’:

```
Help menu:
i - Reload the original image i.e cancel any processing
w - Save the current image as "out.jpg"
g - Convert image into Grayscale using OpenCV
G - Convert image into Grayscale using Conversion Function
c - Cycle through the different color channels of the image
s - Convert image into Grayscale and smooth using OpenCV
S - Convert image into Grayscale and smooth using Smoothing Function
d - Downsample the image by a factor of 2 without Smoothing
D - Downsample the image by a factor of 2 with Smoothing
x - Convert image into Grayscale and take X-derivatives
y - Convert image into Grayscale and take Y-derivatives
m - Display the magnitude of the image gradient
p - Convert image into Grayscale and plot gradient vectors
r - Convert image into Grayscale and rotate the image
h - Display the Help Menu
```

5. References

http://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_tutorials.html
https://docs.scipy.org/doc/