

CS-512 Assignment 1: Sample report

DIMPLE MEHRA

A20457225

Department of
Computer Science
Illinois Institute of
Technology

October 08, 2020

Abstract

This is a sample report for assignments in cs512. The report should be at least 4-6 pages long. It should contain description of the problem you are trying to solve, the algorithm you employed to solve the problem, and the results obtained. The results should be analyzed for correctness. The performance of the algorithm should be evaluated and discussed. Note that answers to review questions should be submitted separately.

1 Problem statement

- The image to be processed by the program should be read from a file if a file name is specified in the command line or capture it from the camera if a filename is not specified in the command line
- The read image should be read as a 3 channel color image
- The program should work for any size image
- Special keys on the keyboard should be used to modify the display image as follows:
- 'i': reload the original image.
- 'w': save the current image into the file 'output.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 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 amount of smoothing.
- 'd': down sample the image by a factor of 2 without smoothing.
- 'D': down sample the image by a factor of 2 with smoothing.
- 'x': convert the image 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

base don the x and y derivatives of the image.

- 'p': convert the image to grayscale and plot the gradient vectors of the image every N pixel and let the plotted gradient vectors have a length of 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 an angle of tetha degrees. Use a track bar to control the rotation angle.
- 'h': display a short description of the program, its command line arguments, and the keys it supports.

2 Proposed Solution

- The user is asked to write a filename to load an image to perform operations on it or press enter key to capture an image from the camera.
- If the image is 1 channel only (it is gray) it is converted to 3 channel image.
- If the image is too big to fit the screen it is reduced until it fits it.
- Special keys functions:

The image is converted to gray using openCV function `cvtColor()`

The image is converted to gray converting each pixel to gray from the bgr values using the formula:
$$g = 0.299*b + 0.587*g + 0.114*r$$

To cycle through the 3 channels bgr each time the key is pressed two of the channels are made zero.

The image is converted to gray using openCV function `cvtColor()` and track bar that controls a smoothing filter is created in the window to control the amount of smoothing.

I convolved a filter with the image creating a function that performed convolution.

To downsample the image without smoothing its dimensions are divided by 2.

To downsample the image with smoothing the openCV function `pyrDown()` is used.

The image is converted to gray using openCV function as before then a Sobel filter of size 5 is applied to image only in x direction. Finally, it is normalized.

The image is converted to gray using openCV function as before then a Sobel filter of size 5 is applied to image only in y direction. Finally, it is normalized.

Both Sobel filters in x and y directions are applied to calculate the magnitude doing the square root of the sum of them to the power of 2. The magnitude is normalized afterwards.

Image is converted to gray and Sobel filter in both directions is applied. For every N pixel the angle of its gradient is calculated as the inverse of the tangent of Sobely divided by Sobel x. Then knowing the angle, both x and y components can be calculated and the arrow drawn.

Image is converted to gray and rotate through the track bar using openCV functions `getRotationMatrix2D()` and `warpAffine()`.

Information about keys' functionally is print on terminal.

3 Implementation details

First of all, I had some problems installing openCV since I had already anaconda3 installed with python 3.6 for another course. After a couple days trying I was able to install openCV but now spyder is not compiling anything. I used Jupyter notebook at last to build code.

I have had some problems with global and local variables for functions I was getting errors that

variables were not declared. Finally, I figured it out but I think not in the best way. Apart from that, most openCV functions were straight forward to use.

I had some issues as well with the track bar because I wasn't reloading the image correctly but once I discover the problem it was the same solution for all the functions that used the track bar.

I have not found a method to eliminate the track from the window when a new button is pressed if it was previously there. A possible solution could have been to create two windows, one without the track bar and another one if the track bar was used, but I did not like it very much so I didn't program it that way.

4 Results and discussion

Below is the output of all the functions:

- a) This is the original image and it is the one that is reloaded when 'i' key is pressed.



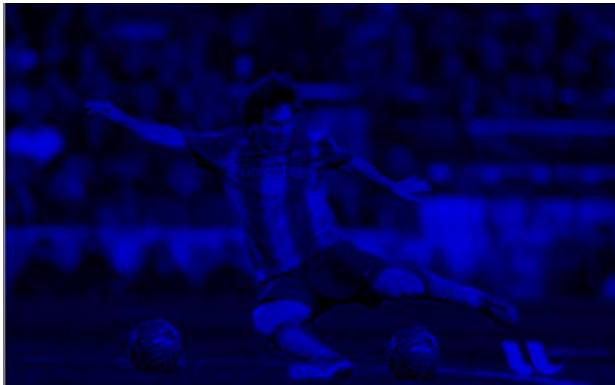
- b) All images used in this document have been exported from the program using the key 'w' that is the one that saves the image to a file.
- c) When 'g' key is pressed the image is converted to gray using the openCV function.



- d) When 'G' is pressed the same conversion to gray is done but using my own code instead of openCV function. The result is indistinguishable from the previous one but it takes some a couple seconds to do it since there is a for loop.



- e) When 'c' is pressed several times the image displays the 3 channel colors bgr.



- f) When 's' is pressed image is converted to gray and smooth is controlled with the track bar.



- g) When 'S' is pressed it was supposed to do the same as the previous one but I could not get it done.



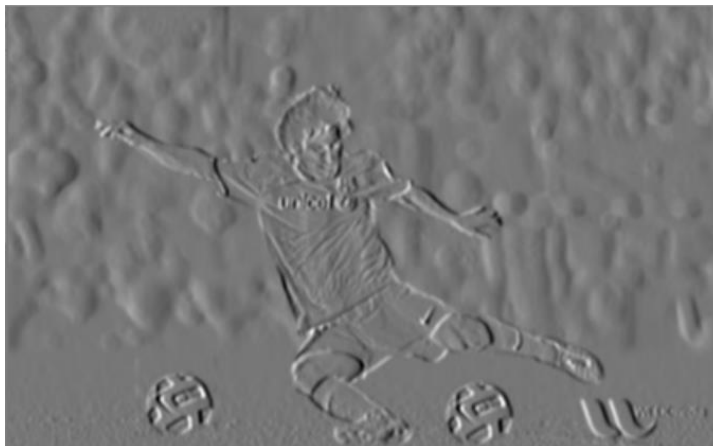
- h) When 'd' is pressed the image is down sampled by 2 without smoothing. Since I am reducing the size of the image I cannot show that it is down sample by 2.



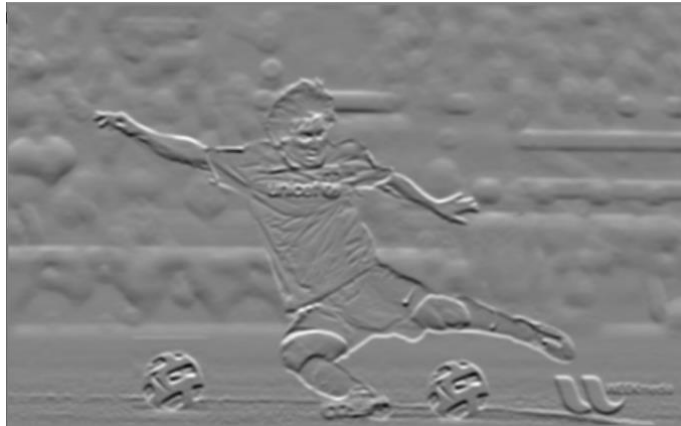
- i) When 'D' is pressed the image is down sampled by 2 with smoothing. Again I cannot show the down sample but the smoothing can be seen in the image compared to the previous one.



- j) When 'x' is pressed the image is converted to gray and convolve with an x derivative filter (Sobel).



- k) When 'y' is pressed the image is converted to gray and convolve with an y derivative filter (Sobel).



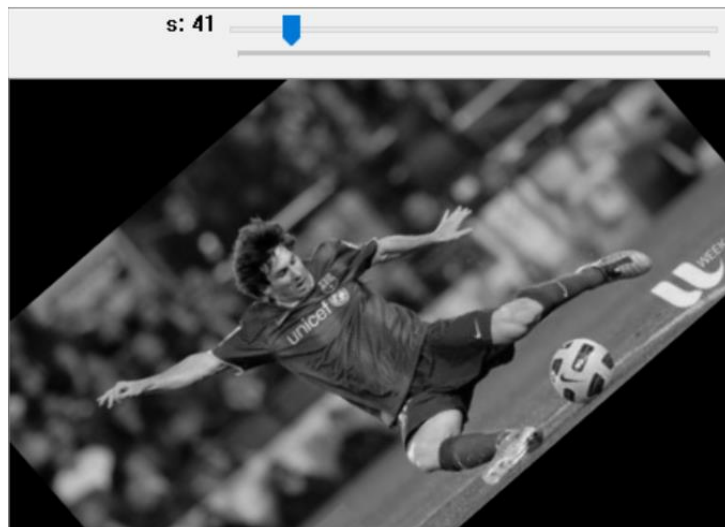
- l) 'm': show the magnitude of the gradient normalized to the range [0,255]. The gradient is computed based on the x and y derivatives of the image.



- m) 'p': convert the image to grayscale and plot the gradient vectors of the image every N pixel and let the plotted gradient vectors have a length of K. Use a track bar to control N. Plot the vectors as short line segments of length K.



- n) 'r': convert the image to grayscale and rotate it using an angle of θ degrees. Use a track bar to control the rotation angle.



5 References

- <https://www.udemy.com/course/python-for-computer-vision-with-opencv-and-deep-learning/>
- <https://www.tutorialspoint.com/opencv/index.htm>
- <https://stackoverflow.com/questions/36428108/python-and-opencv-how-can-i-make-a-slider-window-in-which-i-can-use-to-change>
- Handouts about OpenCV provided by Professor.