CS412 - Homework 1 - Basic operation on image

Khac Le Duy CAO - 1351008

November 23, 2016

1 Note

This should be attached to the homework 1 project of CS412 course. This report will list out some main features and the techniques used in the project

2 Features

When these keys are invoked, the corresponding functionality will work:

- i - reload the original image (i.e. cancel any previous processing)
- w - save the current (possibly processed) 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.
- 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 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 Q degrees. Use a track bar to control the rotation angle. The rotation of the image should be performed using an inverse map so there are no holes in it.
- h - Display a short description of the program, its command line arguments, and the keys it supports.

- **Special feature**: real-time filtering when capturing image

3 Usage

Prerequisite stuff:
- Python installed
There are two independent stage of the program: - First is capturing video:

```
> python main.py
```

- Second is editting image:

```
> python main.py [picture_name]
```

4 Main Techniques

4.0.1 Convolution

I carry out this operation on each pixel with the formulae:

$$H(x,y) = \sum_{i=0}^{M_i-1} \sum_{j=0}^{M_j-1} I(x+i-\alpha_i, y+j-\alpha_j)K(i,j) \tag{1}$$

For special cases that the pixel is at the edge of the image matrix, I replicate 4 edges that the new image matrix can fit the kernel when considering any pixels.

4.0.2 Sobel Derivatives

To compute Sobel Derivatives, first of all we need to create a sobel kernel, which can be Gx for x-derivative or Gy for y-derivative for this operation then convolve that kernel with the image matrix. For example, we have the Gx and Gy kernel matrix size 3 as follow:

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \tag{2}$$

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \tag{3}$$

4.0.3 Gradient Magnitude

Gradient magnitude is the magnitude of the gradient vector at each pixel computed by the formulae:

$$|\nabla f| = \sqrt{(\frac{\delta f}{\delta x})^2 + (\frac{\delta f}{\delta y})^2} \tag{4}$$

Or in the other words, the value of x-derivative and y-derivative at each pixel is x value and y value respectively in gradient vector which can be computed by using sobel derivatives

5 Project Structure

5.0.1 Image Processor

This processor will be initiated with the path to the loaded image in image-load mode. It will hold two image, one is for processing and one is for conservation There is a global function to map keys to functionality that uses the power of my Image Processor

5.0.2 Video Processor

This is just an adapter from the frame read from video capturer to the Image Processor. It will execute the method of the image processor every frame based on the key users choose. It means everytime a key is invoked, that key will be held in the Video Processor to let the Image Processor know which method should be executed on the input frame.

6 Samples

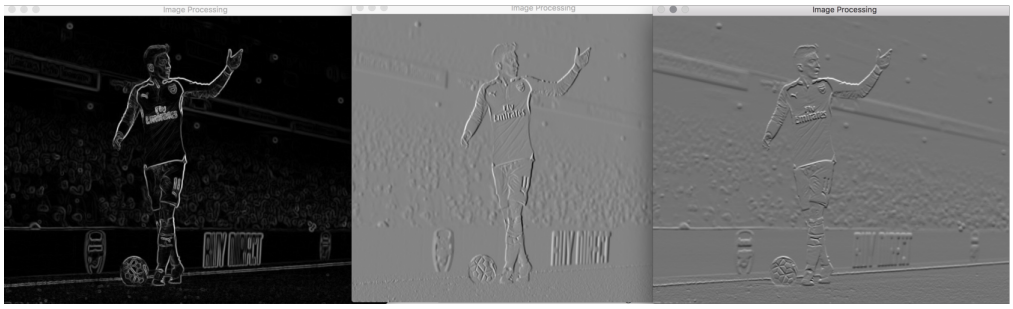

Figure 1: Gradient Magnitude - Y-Derivative - X-Derivative

Figure 2: Grayscale

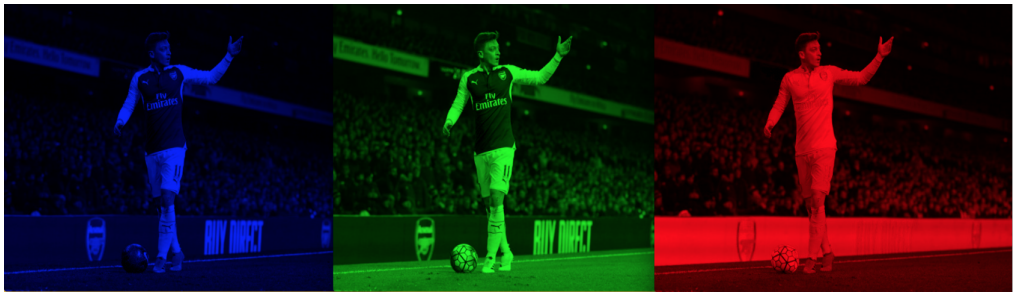

Figure 3: Three channel of an image

Figure 4: Plot Gradient Vector

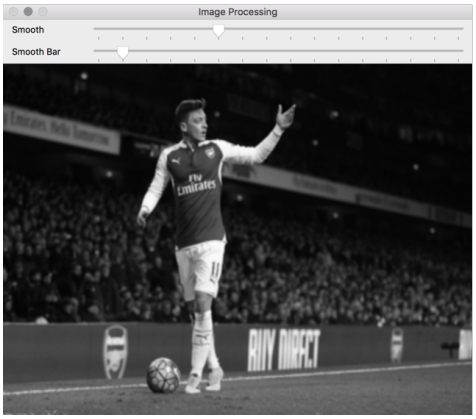

Figure 5: Smootify image

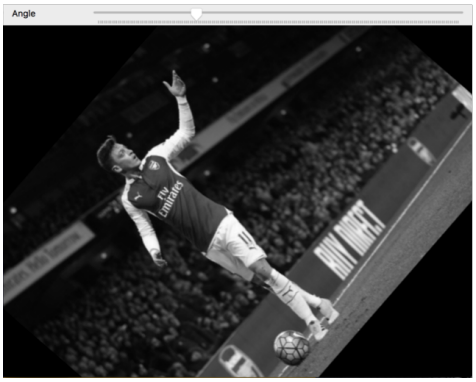

Figure 6: Image rotation