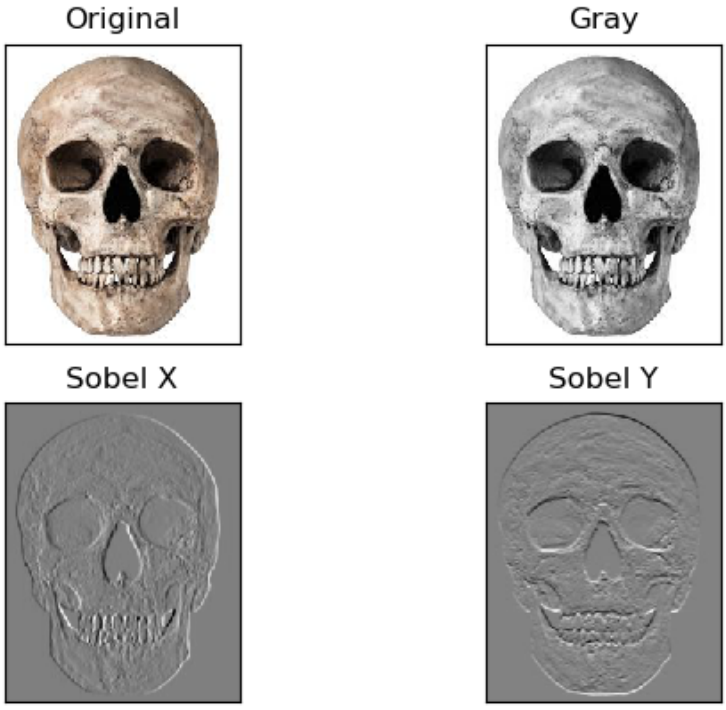
Computer Vision Lab 1

Working with images and gradients

In this lab, you will practice working with images, for example, using Sobel filter over input image. In all the labs of this module, you may either use Python or Matlab for programming. However, Python will be the language for demonstration and you are encouraged to try Python if you are not familiar with the language. We will use OpenCV 3.x and Python 3.x. You also needs numpy and matplot library for the labs. You may need to look at OpenCV online tutorials for help https://docs.opencv.org/3.2.0/d6/d00/tutorial_py_root.html . See moodle also for the cheatsheets of python and some useful libraries.

1. Firstly, read in the input image from moodle and convert the color image into grayscale image. Then apply Sobel filter to the input image. Note that there will be 2 filters, one for x direction and one for y direction. Please plot the gradients for x and y as two separate images. The output plots should look like the following:



To get started, you can try the following python code:

```
import cv2
```

```
import numpy as np
from matplotlib import pyplot as plt

img = cv2.imread('skull.jpg')
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

sobelx = cv2.Sobel(gray,cv2.CV_64F,1,0,ksize=3)
sobely = cv2.Sobel(gray,cv2.CV_64F,0,1,ksize=3)

plt.subplot(2,2,1),plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
plt.title('Original'), plt.xticks([]), plt.yticks([])
plt.subplot(2,2,2),plt.imshow(gray,cmap = 'gray')
plt.title('Gray'), plt.xticks([]), plt.yticks([])
plt.subplot(2,2,3),plt.imshow(sobelx,cmap = 'gray')
plt.title('Sobel X'), plt.xticks([]), plt.yticks([])
plt.subplot(2,2,4),plt.imshow(sobely,cmap = 'gray')
plt.title('Sobel Y'), plt.xticks([]), plt.yticks([])
```

Now, please practice by implement your own Sobel filter and you can try other gradient filters introduced in the lecture.

2. Compute the magnitude and orientation from the above gradients. Please plot these two as two separate images. For the gradient orientation, you can show the image with color coded direction. For example, use red color for 0-90 degree, cyan for 90-180 degree, green for 180-270 degree, and yellow for 270-360 degree. The output plots should look like the following: