Computer Vision (SPRING 2016) Problem Set #1

Fan Zhou
fan.zhou@gatech.edu

1a: Interesting Images

Image 1 - ps1-1-a-1.png

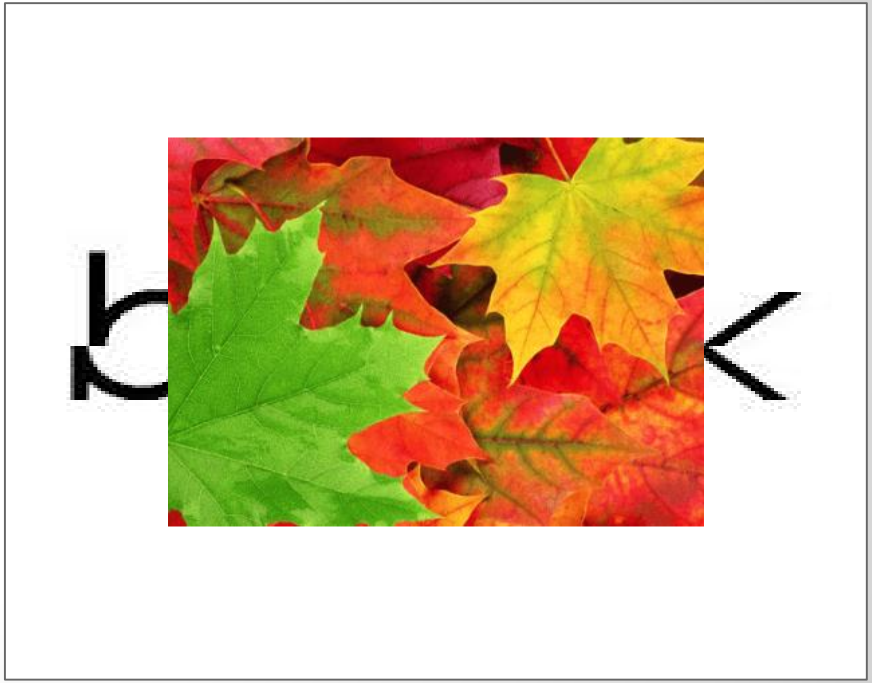


Image 2 - ps1-1-a-2.png

2a: Swapped Red and Blue

ps1-2-a-1.png

2b: Monochrome Green

Img1_green - ps1-2-b-1.png

The left image is the monochrome green image. For comparison, a grayscale image converted from the original BGR image is stored in the “temp” folder.

2c: Monochrome Red

Img1_red - ps1-2-c-1.png

2d: Expectations

The green monochrome image most resambles the grayscale conversion of the original image. This is the same as what I thought. Although human eyes has three kinds of cone cells which are sensitive to red, green, and blue, respectively, from luminosity point of view green colors (555 nm) contribute the most towards human perception of brightness.

When converting an image to a grayscale in openCV, the converted intensity is calculated from 0.114*Blue+0.587*Green+0.299*Red. We see the grayscale image has the most “green” portion.

To simulate human vision system, I expect CV algorithm to work better on the green color. Depending on different applications like military, weather, etc., CV algorithm also expect to work better on other wavelengths like GHz, THz, infrared, visible, and UV.

3a: Replacement of Pixels

ps1-3-a-1.png

4a: Geometric Values

The standard deviation of img1_green is 41.0

The mean of img1_green is 78.0

```
##Code
import numpy as np
standard_deviation = np.round(np.std(img1_green))
mean_value = np.round(np.mean(img1_green))
print ('The standard deviation of img1_green is',
standard_deviation)
print ('The mean of img1_green is', mean_value)
```

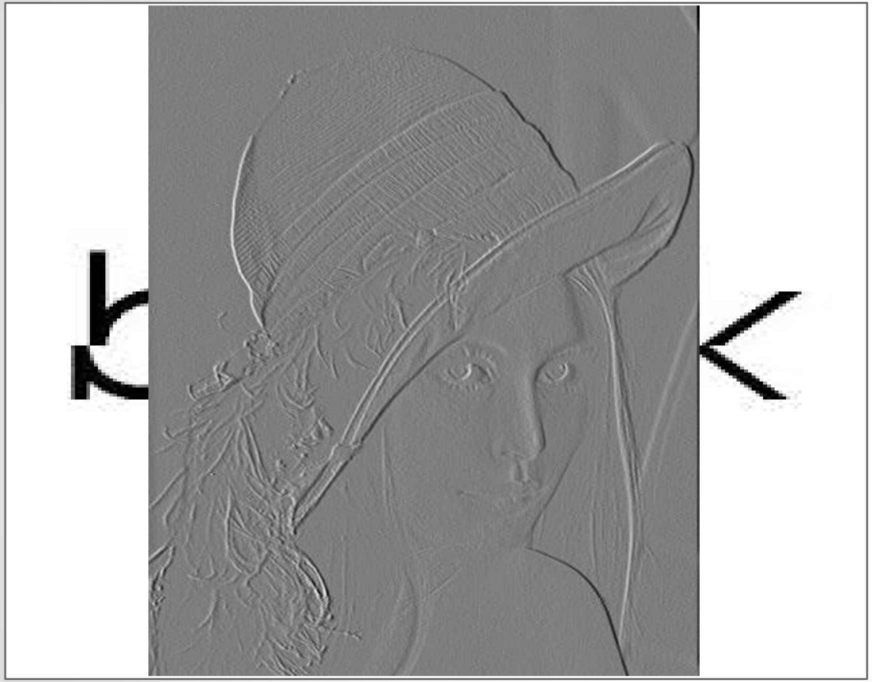
4b: Arithmetic Operation

ps1-4-b-1.png

4c: Shifted Image

ps1-4-c-1.png

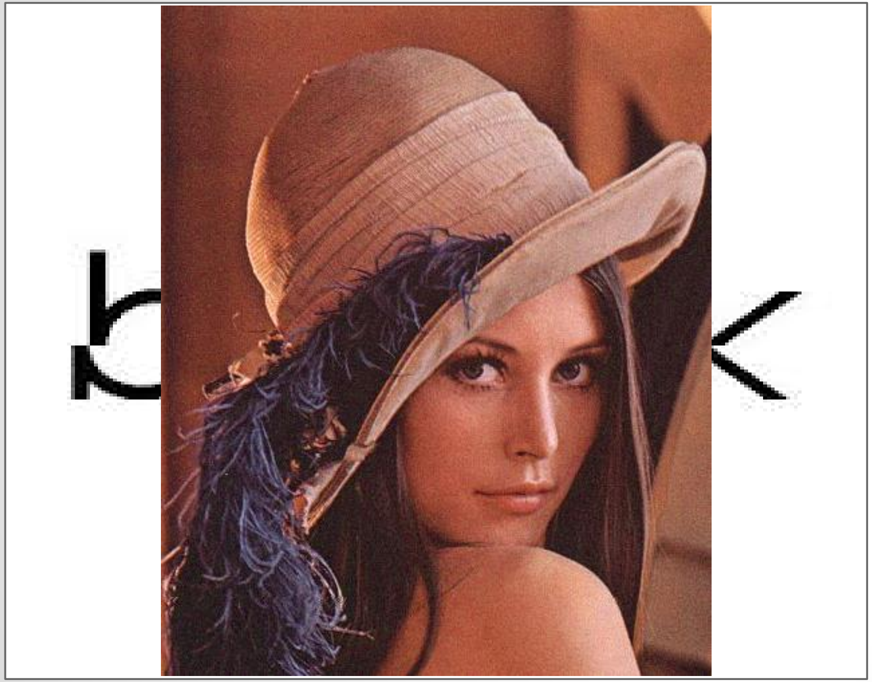
4d: Difference Image



ps1-4-d-1.png

The left image is intensity-scaled in the difference calculation. An image without scaling is stored in the “temp” folder. As we see, the scaled one retains more information of edges than the unscaled one.

5a: Noisy Green Channel



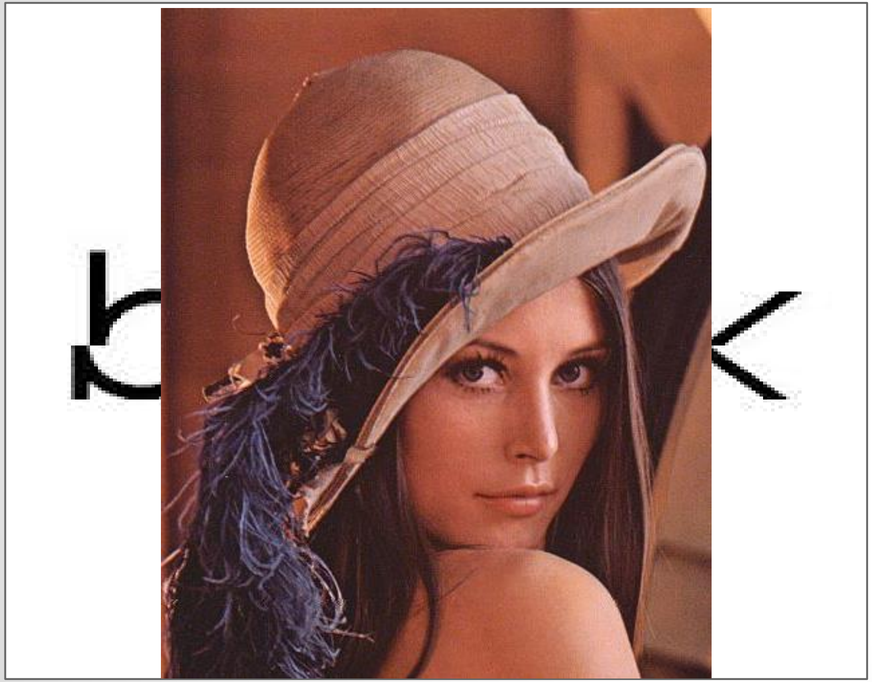
ps1-5-a-1.png

[What value of Sigma is used?]
The value of Sigma I used is 10. With this value, there are obvious noises in the image as you can see the skin is not very smooth.

This number may change depending on different images, people, and monitors.

Note: After adding (+/-) noise to green channel, the pixel values clips at 0 and 255. Although rescaling the intensity is also doable, I think for an image sensor once noises overflow its limits, it is reasonable to clip at the limits since noises intend to do this.

5b: Noisy Blue Channel



ps1-5-b-1.png

5c: Comparing Noise

Noisy blue channel image looks better to me since it is closer to the original image and losses less information. However, noisy green channel is more “rough”. This may be because the visual system of most people including me is more sensitive to the green color (eventhough the standard deviaiton of green channel is smaller than the blue channel in this image.). Thus, for the same noises, people will feel more tiny pixel intensity change in green channel than blue channel.

The results may vary depending on different monitors.