Comp Photography (Sp 2015)
HW 5

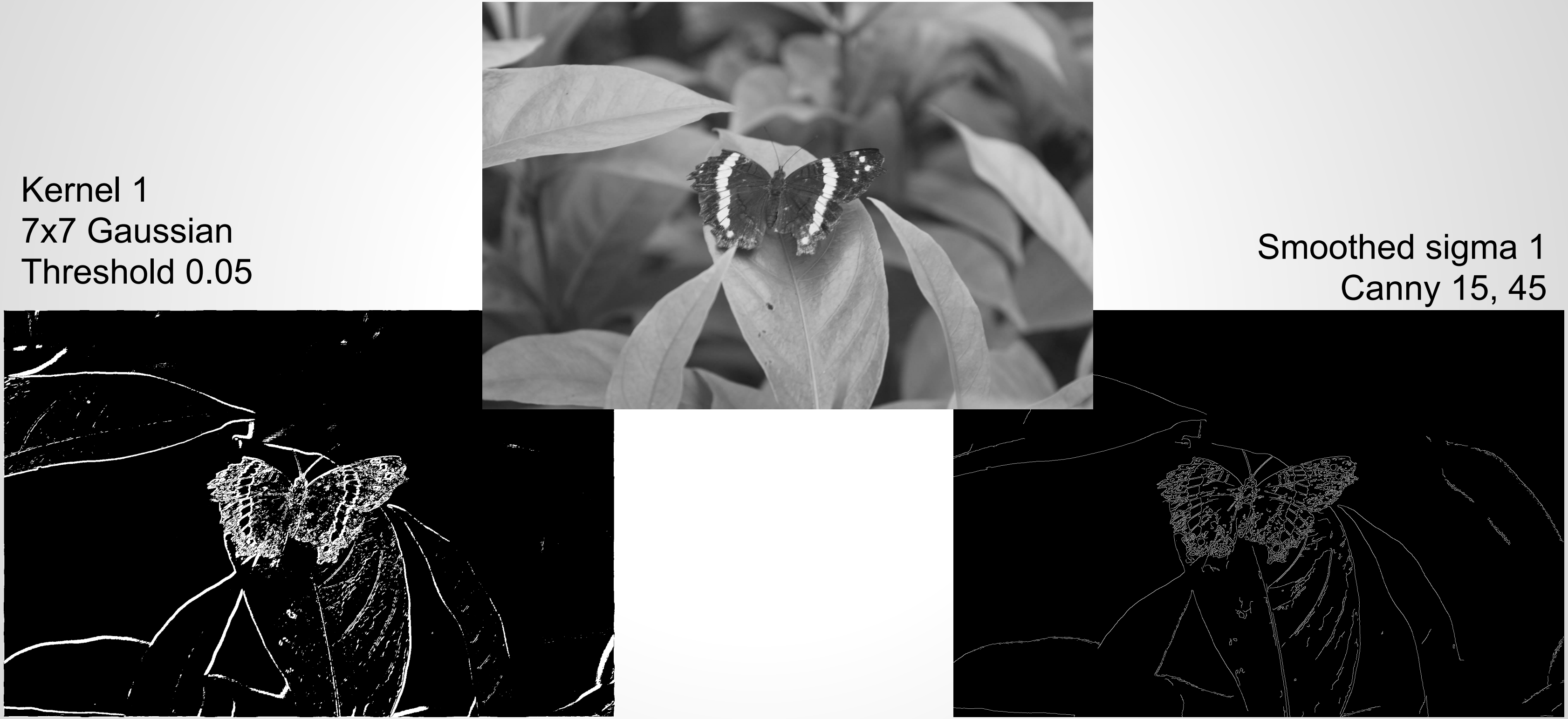
Jonathan Hudgins

Gradient Functions

1. imageGradientX
   - a. The simplest and fastest way of accomplishing this is to use the numpy “roll” function to shift all pixels to the right (with wrapping).
   - b. But because we are supposed to iterate over the pixels, I converted image to int, looped through y, then x, setting the result value to:

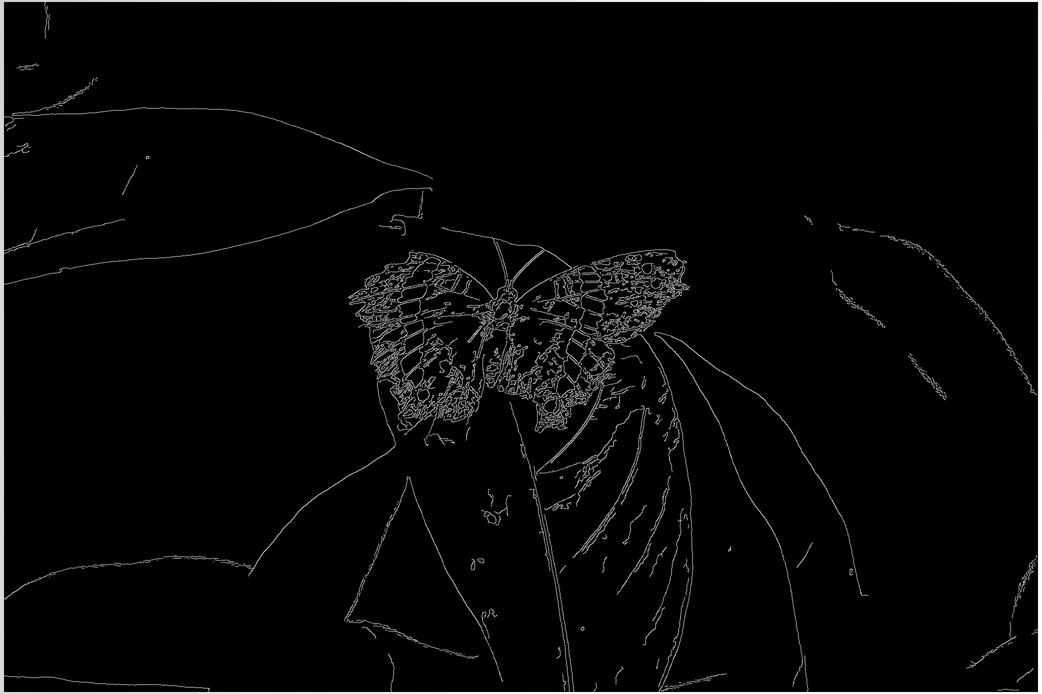
     abs(image[y, x]-image[y,(x+1)%image.shape[1]]
   - c. Convert back to uint8 (no need to worry about overflow because the biggest absolute delta will be 255)
2. imageGradientY
   - a. Same as “imageGradientX” only use:
   - b. abs(image[y, x]-image[(y+1)%image.shape[0],x])
3. computeGradient
   - a. Convert to float
   - b. The simplest and fastest way is to use “scipy.signal.convolve2d”
   - c. But because other libraries are disallowed I loop through y, then x (starting at 1 ending 1 less than image.shape)
   - d. set result equal to sum of numpy.multiply of slice around y,x and the kernel

Overview



Kernel 1
7x7 Gaussian
Threshold 0.05

Smoothed sigma 1
Canny 15, 45

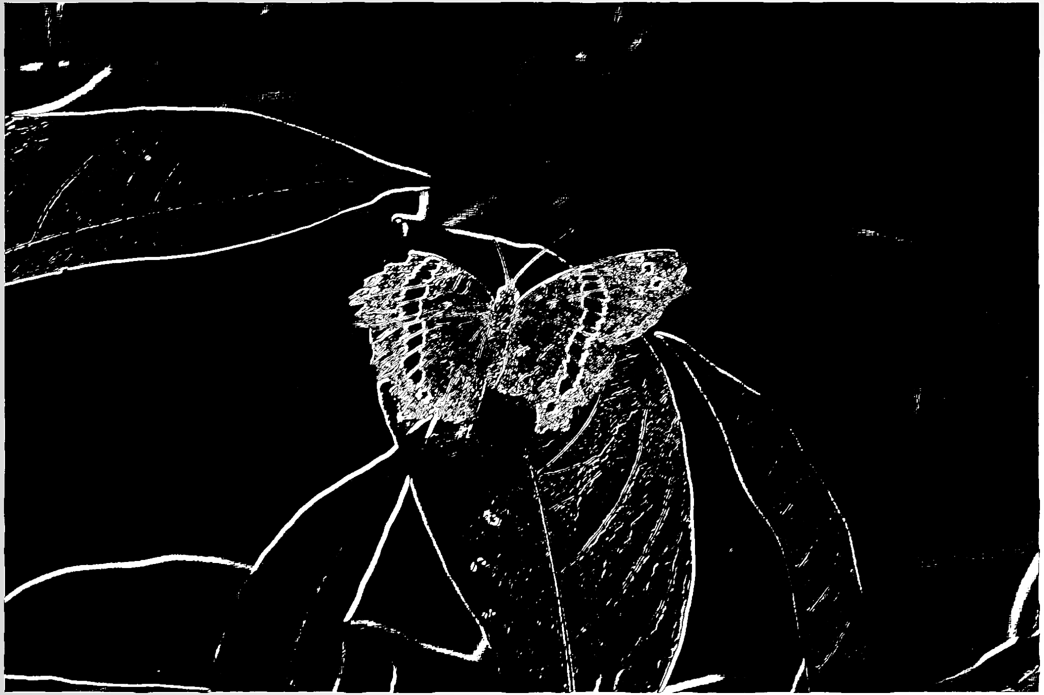


Canny

First I smoothed the image using OpenCV “GaussianBlur” (7x7 with sigma of 1)

Then I used the OpenCV function “Canny” with threshold1 = 15 and threshold2 = 45.

Notice the single pixel width.



Single Kernel

To create my own edges I tried a variety of kernels convolving a differential and gaussian kernel and then applying.

The differentials I tried were:

$$\begin{matrix}0 & -1\\ -1 & 2\end{matrix} \qquad \begin{matrix}-1 & -1\\ -1 & 3\end{matrix} \qquad \begin{matrix}-1 & -2\\ -2 & 5\end{matrix}$$

(with the other rows and columns 0)

The gaussians ranged from 3x3, 5x5, 7x7 with sigma 0.5, 1, 2.

I then convolved the kernel with the image. Then use absolute value of difference between pixel and median (distance from median). Normalize these values based on the maximum. Then threshold to 0 or 255. (Some nifty numpy matrix functions make all of this expedient!)
This image is pretty close (biggest difference is the edge width)