We'll be covering image gradients and edge detection. Image gradients can be used to measure directional intensity, and edge detection does exactly what it sounds like: it finds edges! Bet you didn't see that one coming.

First, let's show some gradient examples:

import cv2

import numpy as np

cap = cv2.VideoCapture(1)

while(1):

# Take each frame

\_, frame = cap.read()

hsv = cv2.cvtColor(frame, cv2.COLOR\_BGR2HSV)

lower\_red = np.array([30,150,50])

upper\_red = np.array([255,255,180])

mask = cv2.inRange(hsv, lower\_red, upper\_red)

res = cv2.bitwise\_and(frame,frame, mask= mask)

laplacian = cv2.Laplacian(frame,cv2.CV\_64F)

sobelx = cv2.Sobel(frame,cv2.CV\_64F,1,0,ksize=5)

sobely = cv2.Sobel(frame,cv2.CV\_64F,0,1,ksize=5)

cv2.imshow('Original',frame)

cv2.imshow('Mask',mask)

cv2.imshow('laplacian',laplacian)

cv2.imshow('sobelx',sobelx)

cv2.imshow('sobely',sobely)

k = cv2.waitKey(5) & 0xFF

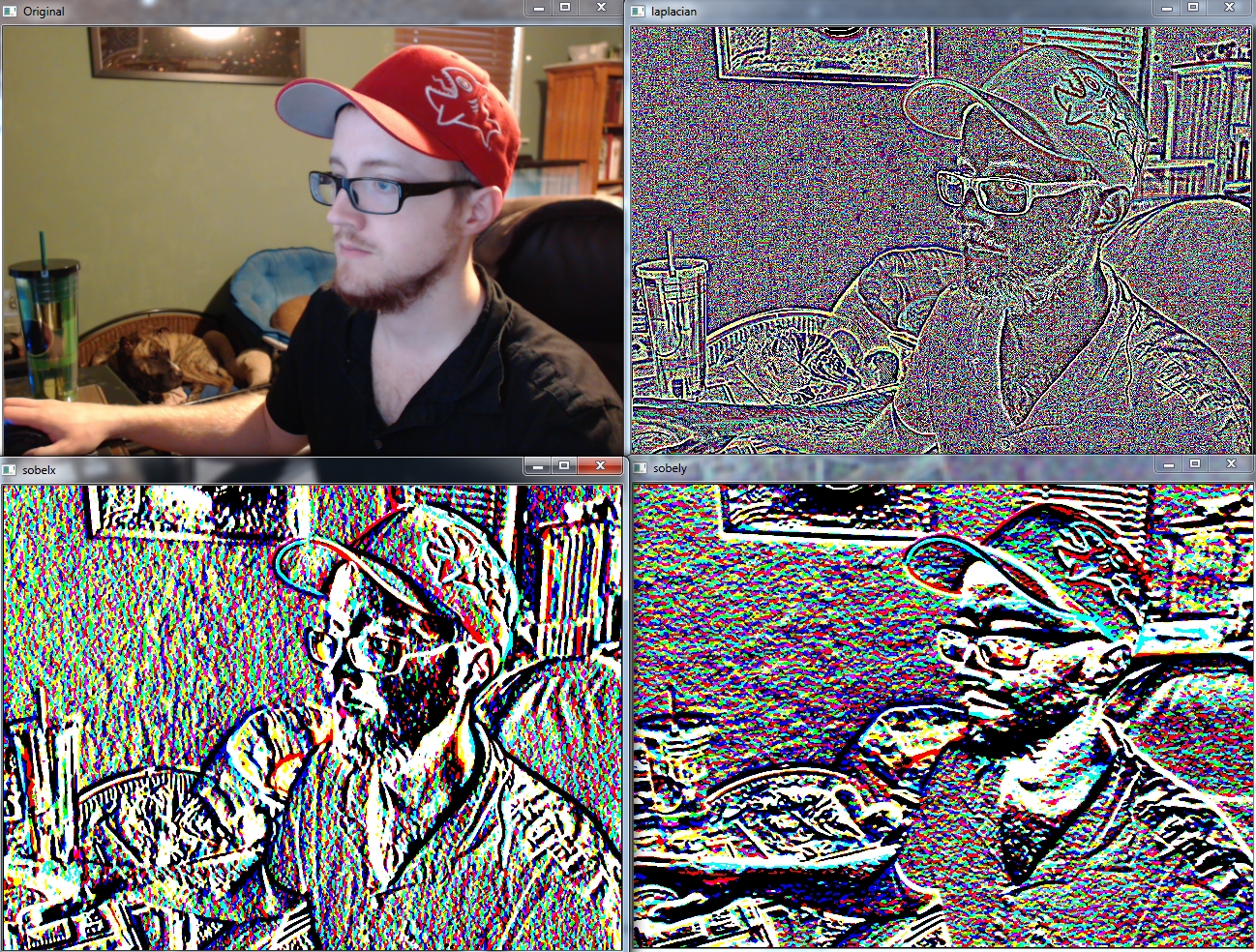
if k == 27:

break

cv2.destroyAllWindows()

cap.release()

Result:



If you're wondering what the cv2.CV\_64F is, that's the data type. ksize is the kernel size. We use 5, so 5x5 regions are consulted.

While we can use these gradients to convert to pure edges, we can also use [Canny Edge detection!](https://docs.opencv.org/master/dd/d1a/group__imgproc__feature.html#ga04723e007ed888ddf11d9ba04e2232de)

import cv2

import numpy as np

cap = cv2.VideoCapture(0)

while(1):

\_, frame = cap.read()

hsv = cv2.cvtColor(frame, cv2.COLOR\_BGR2HSV)

lower\_red = np.array([30,150,50])

upper\_red = np.array([255,255,180])

mask = cv2.inRange(hsv, lower\_red, upper\_red)

res = cv2.bitwise\_and(frame,frame, mask= mask)

cv2.imshow('Original',frame)

edges = cv2.Canny(frame,100,200)

cv2.imshow('Edges',edges)

k = cv2.waitKey(5) & 0xFF

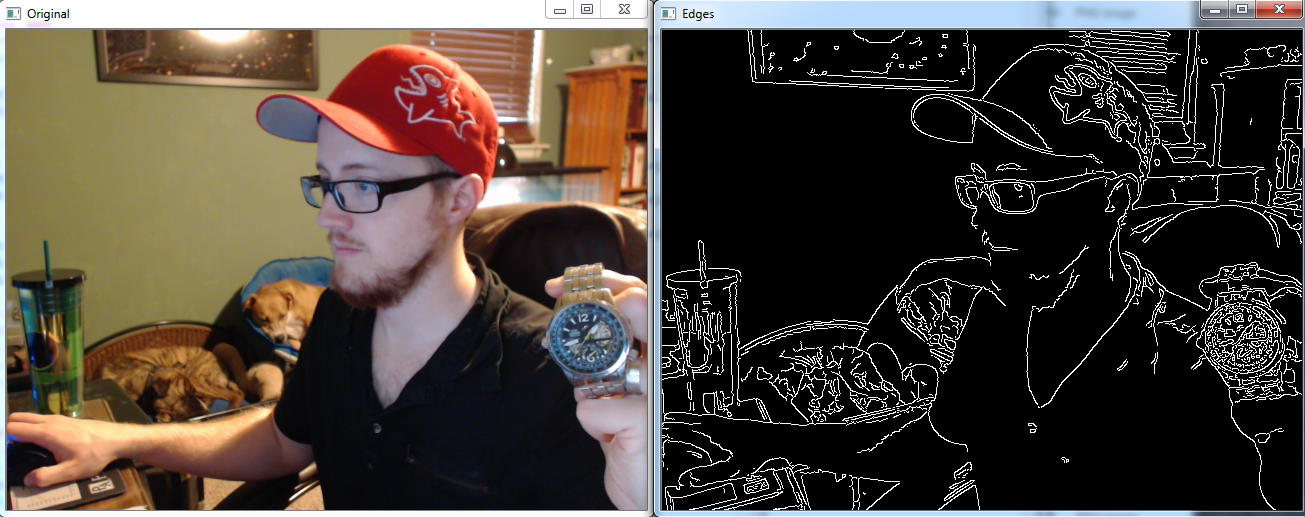
if k == 27:

break

cv2.destroyAllWindows()

cap.release()

Result:



That's pretty darn good! It's not all perfect, however. Notice shadows are causing edges to be detected. One of the most obvious would be from the shadow that the blue dog bed is giving off.