Demo Programs

find_blue_in_image.py
find_blue_in_video.py
The goal is to detect objects of the color that is specified in the program. The following three images illustrate the operation of the program. Figure 1 is the original input image in which we wish to highlight blue objects. Figure 2 is the mask applied to Figure 1. Notice it has high values in areas where the color blue is present. Figure 3 is the result of masking the original image

video_rgb.py
video_rgb_keys.py

cv2 functions

cv2.cvtColor(image, cv2.COLOR_BGR2HSV)
This function converts an image from a BGR representation to an HSV representation. The first argument is the image. The second argument must be cv2.COLOR_BGR2HSV. This is a standard.

cv2.inRange(image, lower, upper)
This function illuminates the regions that have HSV values between the lower and upper bound specified

cv2.bitwise_and(img1, img2, mask)
This function computes the bitwise 'and' between img1 and img2, if the corresponding element in mask is not 0. In the demos, img1 and img2 are the same image.

Steps to segment out objects of certain color

Find the HSV value associated with a certain color. This can be obtained via:

```
blue = np.uint8([[[255, 0, 0]]])   # 3D array
hsv_blue = cv2.cvtColor(blue, cv2.COLOR_BGR2HSV)
```
Let H be the Hue value of blue (H = 120).
Take lower bound as [H-10, 50, 50] and upper bound as [H+10, 255, 255] (for example).

Note: It is important to make sure that the BGR values are fed in as a numpy array of type uint8. It is also important that the array is fed in as a 3D array as shown above