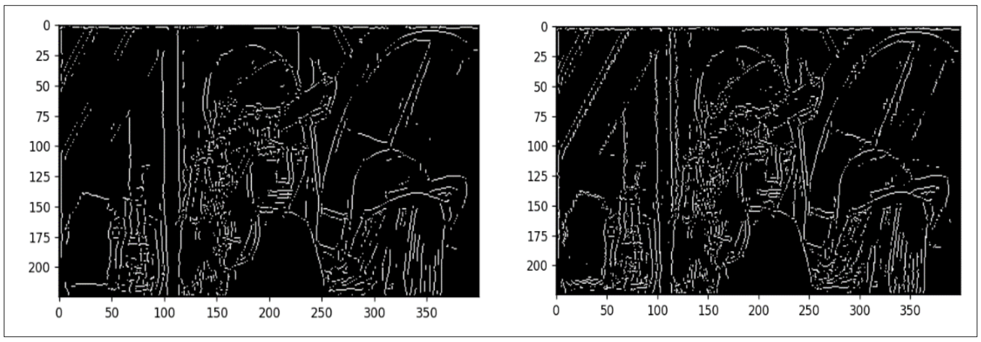
Computer Vison Project 2

120210211 Lee Jihwan

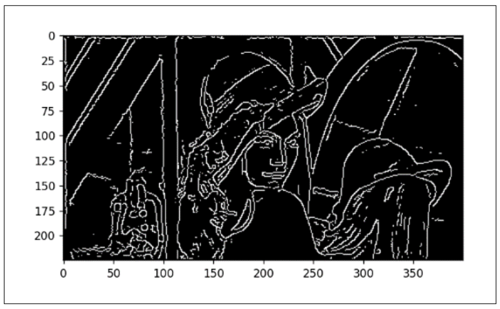
[1] How can we improve the edge detection performance?

In the code, we can get the image's vector via numpy array method. Then, we apply the Gaussian Filters for the images. In the python codes, there are defined "gaussian filters" function. The gaussian filters functions parameter is image vector value. The array b in gaussian filters function adjusts mask size. Smoothing depends on mask size. For example, Larger mask size reduce noise, but worsen localization. So we have to reduce mask size to get the more detailed edge detection. So first, I tried to change the array b variable. That's meaning about the to apply the small mask size than before. I changed b = array([[2, 4, 5, 2, 2],[4, 9, 12, 9, 4],[5, 12, 15, 12, 5],[4, 9,   12, 9, 4], [2, 4, 5, 4, 2]])/156 to b = array([[2, 4, 5, 2, 2],[4, 9, 12, 9, 4],[5, 12, 15, 12, 5],[4, 9, 12, 9, 4], [2, 4, 5, 4, 2]])/1000. The result shows that improved certainty to the location of the edges.



[fig 01] After applying the more small mask, the picture's edges become more detailed (left: original image, right: after change b)

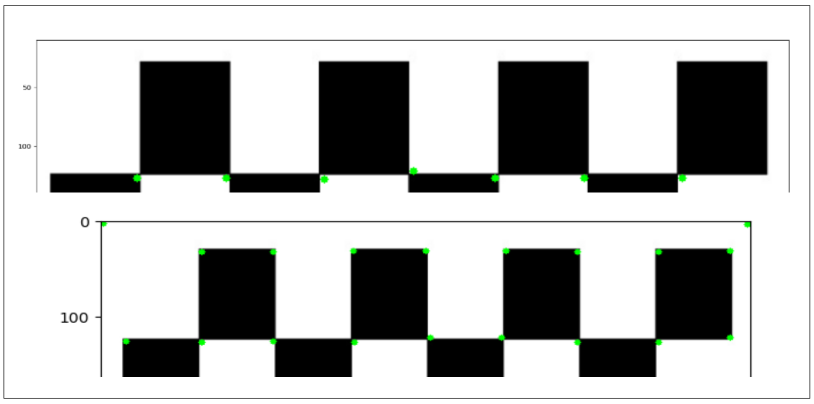
Edges become more detailed, but the problem is that edges looks more complicated. So it need to smoothing steps. I apply the Finte difference filters in "finding gradient function". The input parameter 'im' is the applying the gaussian filter images. I changed the op2 = array([[-1, -2, -1],[0, 0, 0],[1, 2, 1]]) filters to op2 = array([[1, 2, 1],[0, 0, 0],[-1, -2, -1]]) to apply the sobel filters. After changing the filters, the edges become more clean and approciate.



[fig 02] After applying the sober filters, the picture's edges become more clean and exquisite.

[2] How can we improve the keypoint matching performance?

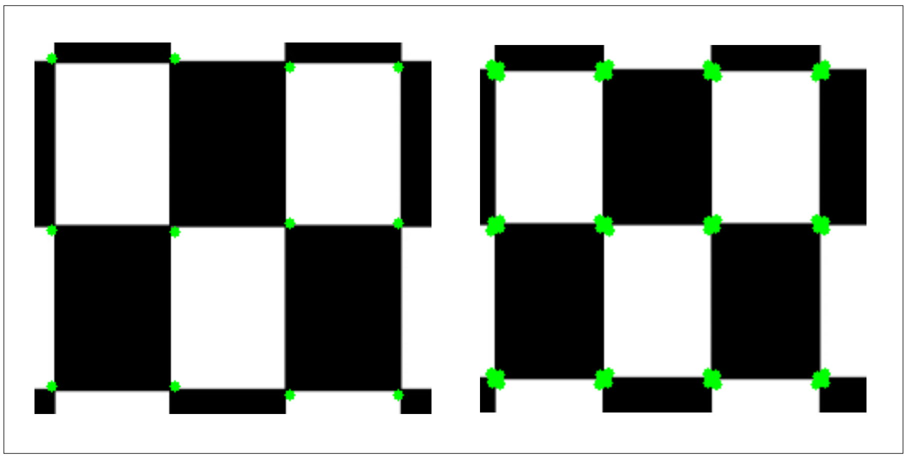
When the 'lst_sigma' parameter value's less than 0.5, the keypoint matching performance becomes worse. And the lst_sigma value between 0.707 and 0.999, the key point matching performance becomes better. The lst_sigma effects in the creating haar filters. For example, when the lst_sigma value is 0.5, Haar wavelet becomes 2*2. And when the lst_sigma value is 0.75, haar wavelet becomes 3*3. And if the lst_sigma value is 1, haar wavelet becomes 4*4. So It means that when the wavelet size is 3*3, it is best to match the keypoint matching performance. You can see the difference between 2*2 and 3*3 through the below [fig 03].



[fig 03] Top: 2*2 haar wavelet, Down: 3*3 wavelet

The 2*2 haar wavelet can't detect top corner, while 3*3 wavelet detect it. And the terminal result shows that 2*2 haar wavelet's number corners is 43, but 3*3 haar wavelet is 83.

Second, When I adjust 'nms_kernel_size' parameters, it shows better performance. 'nms_kernel_size ' is to be used during non maximal suppression. When the 'nms_kernel_size' is 25, the number corners is 83, and I changed it to 1, the number corners become 508. The difference in performance is shown in [fig 04].



[fig 04] Left: 25 nms_kernel_size, Right: 1 nms_kernel_size