CS6550 Computer Vision Homework1 – Image Features

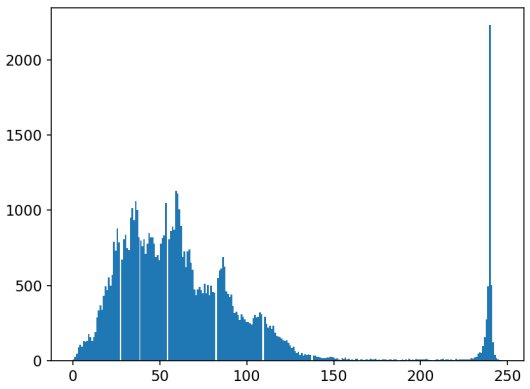
Due Date: 10/20 23:30

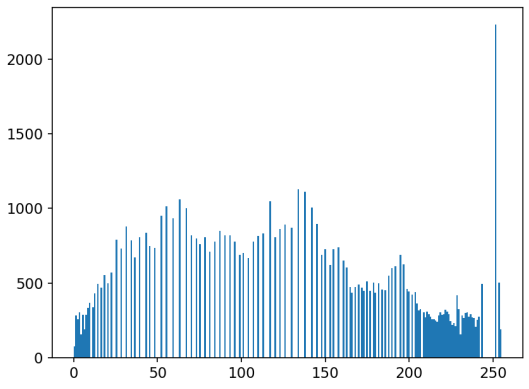
1.(30%)Histogram equalization

- (a.)(10%)grayscale the image and implement histogram equalization and explain how do you implement step by step (you can not use cv2.equalizeHist() or other similar functions.)
- (b.)(10%)(1 image)show the image before and after histogram equalization
- (c.)(10%)(1 image)plot the distribution of pixel value before and after histogram equalization

(result of (b.))

(result of (c.))





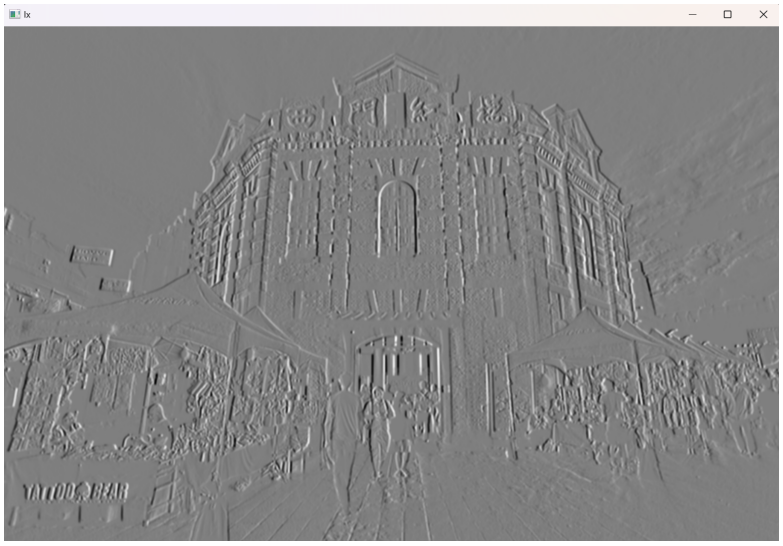
2.(40%)Harris corner detector

- (a.)(10%)implement Harris corner detector. You should implement each of the following steps as separate functions(you can not use cv2.cornerHarris(), cv2.Sobel(), cv2.GaussianBlur() or other similar functions.)
  - (i.)Grayscale and Gaussian Smooth(1 image): turn image into grayscale and apply Gaussian blur with σ=3 and kernel_size=3
  - (ii.)Intensity Gradient (Sobel operator)(2 image): Apply the Sobel operators(size=3) to the blurred images and compute the magnitude of the gradient, show gradient intensity map of x and y direction.
  - (iii.)Structure Tensor(1 image): Use the Sobel gradient magnitude above to compute the structure tensor H of each pixel. Next, compute the corner response of H with window size 3x3 using Harris operator. Show the corner response R after thresholding the response.
  - (iv.)Non-maximal Suppression(1 image): Perform non-maximal suppression on the results above with window_size=5.
- (b.)(10%)
  - (i.)Show images processed after each step in (a.)
  - (ii.)(1 image)Shows the original image(grayscale) with corner points overlaid.
- (c.)(10%)(2 images similar to (a.)(iii.))
  - (i.)Try a different window size in computing the structure tensor H of each pixel.
  - (ii.)Try a different threshold in thresholding corner response.
- (d.)(10%)Discuss the result of (b.) and (c.)
  - (i.)How does window size affect the result?
  - (ii.)How does threshold affect the result?

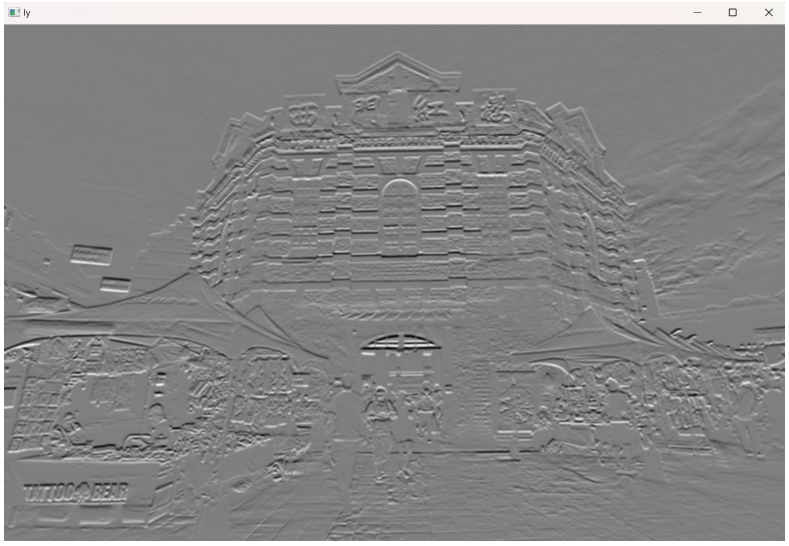
(original image)

(result after Gaussian blur)

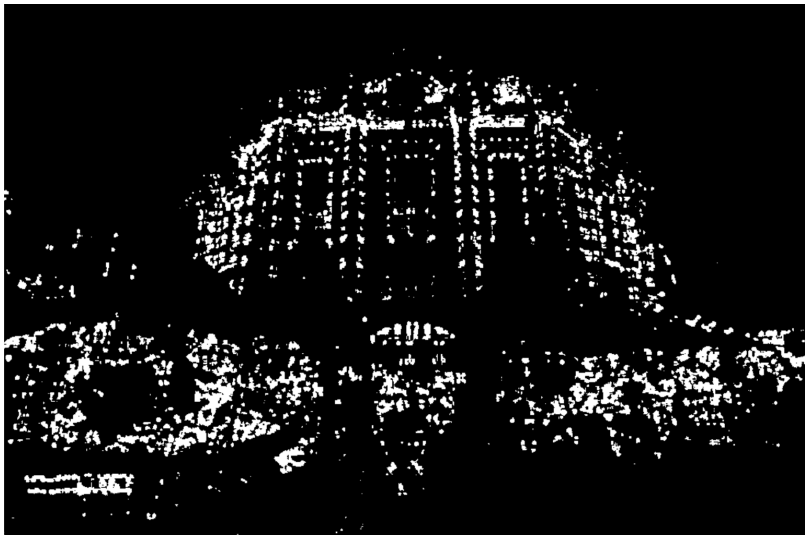
(result after applying Sobel operator(x-direction))



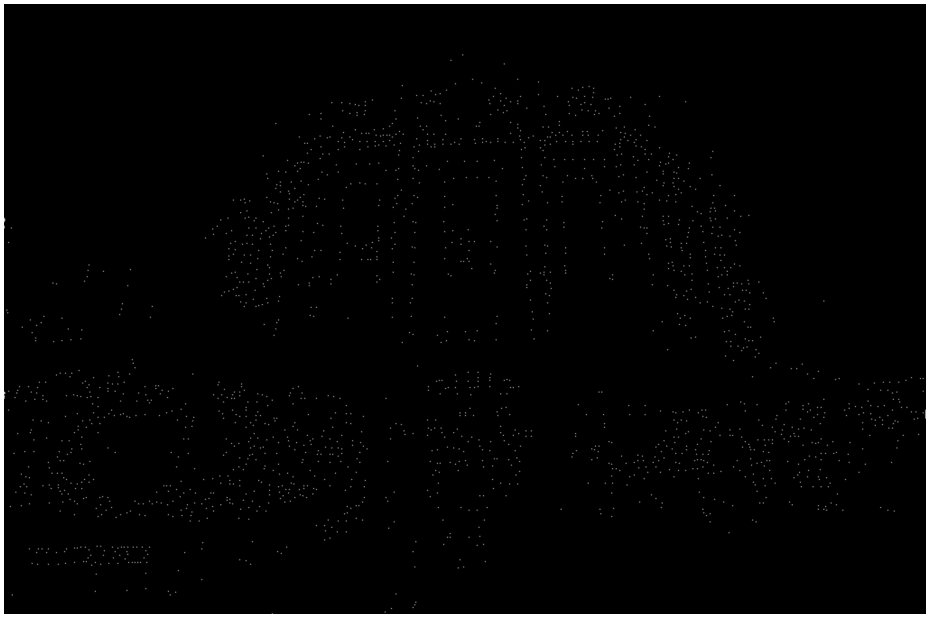
(result after applying Sobel operator(y-direction))



(Harris response)



(result after non-maximal suppression)



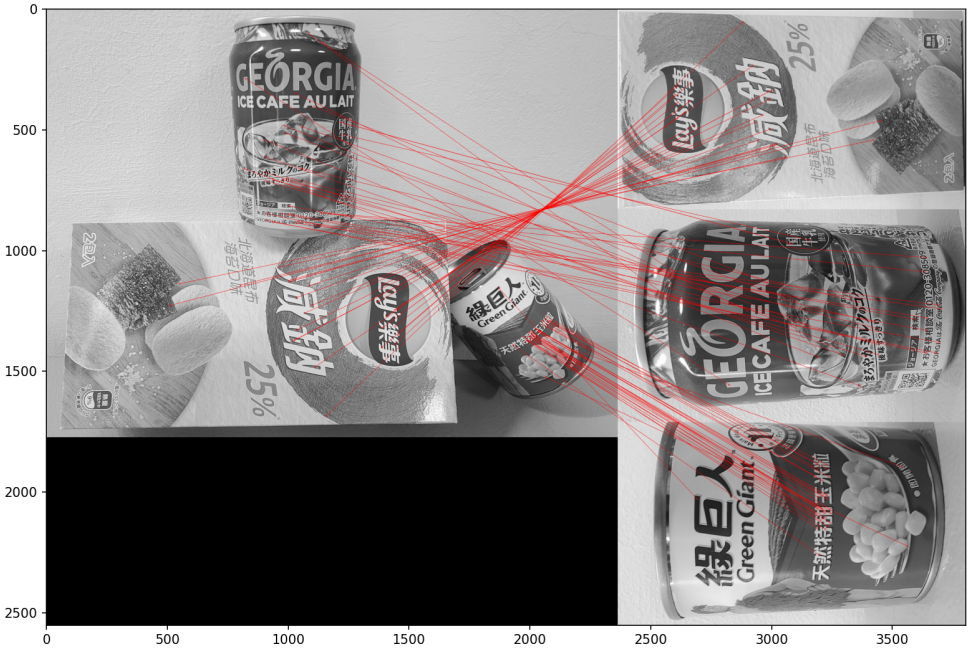
(final output)

3.(40%)SIFT object recognition (2 image)

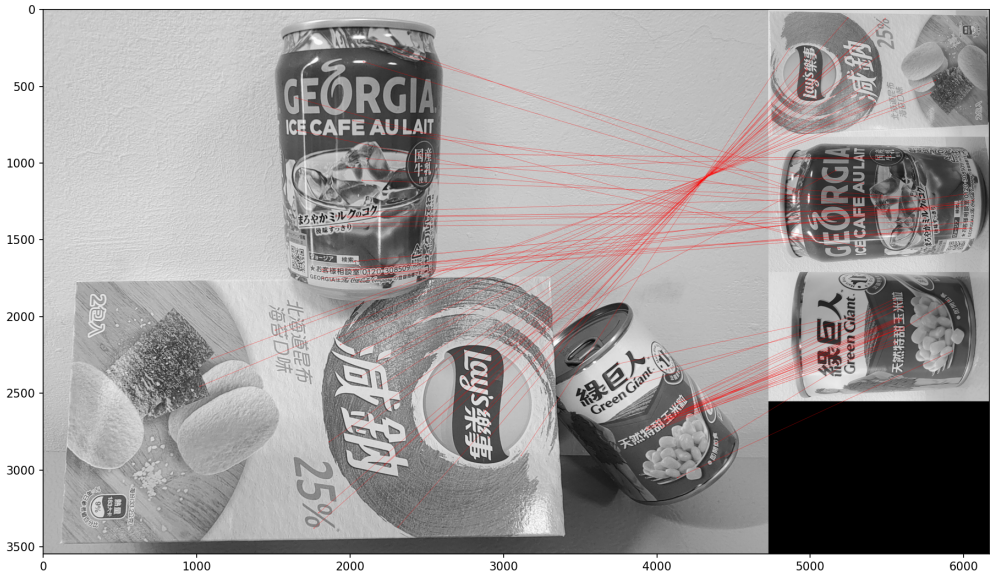
- (a.)(5%)Grayscale the image, Detect keypoint and extract SIFT feature
- (b.)(5%)Brute-force matching and ratio test, you need to implement your own matching function and ratio test function.(you can not use cv2.BFMatcher() or other similar functions)(you need to find as more as possible but not excess 20 matches for each object, for example, there are 3 objects in the right image, so there should be 20 matches for the corn can, 20 matches for the coffee can, and 20 matches for the chips box, if there are more than 20 matches for an object, extract top 20 matches with the smallest distance.)
- (c.)(10%)(1 image)Plot matching result on the images
- (d.)(10%)(1 image)Scale the scene image to 2.0x and redo (a.)(b.)(c.)
- (e.)(10%)
  - Discuss the cases of mis-matching in the point correspondences in (c.) or (d.).
  - Discuss the difference between the results before and after the scale.

(original image)

(result of (c.))



(result of (d.))



Tips:

1. Some helpful functions to Harris corner detection:
    a. cv2.cvtColor()
    b. cv2.getGaussianKernel()
    c. cv2.filter2D()
2. Some helpful functions to SIFT object recognition:
    a. cv2.SIFT_create()
    b. cv2.drawMatchesKnn()

Note:

1.The report should contain:

(i.) your implementation code
(ii.) explanation of code
(iii.) result images

and organized them as the following format:

Title(Question 1)

1.implementation(step by step)
step1:
    code
    explanation of code
    result image
step2:
    code
    explanation of code
    result image

2.discuss section:
    a.discuss about …
    b.discuss about …

2.Your code should show the image while running, it's recommended to use OpenCV to show the images.
3.You should provide a ReadMe.txt file about how to run your code
4.Try adding comments as much as you can to better understand your code
5.You should submit a HW1_{Student-ID}.zip containing only the following files(.jpg files are the images we provided, not the result images):
    hw1-1.jpg
    hw1-1.py
    hw1-2.jpg
    hw1-2.py
    hw1-3-1.jpg
    hw1-3-2.jpg
    hw1-3.py
    ReadMe.txt
    report.pdf
6. If you have any questions, please pose your questions in the eeclass.