Homework 2 Image Processing

Due: Apr 9th, 2023 11:59pm

TAs responsible for Homework2:
高聿翀: gaoych@shanghaitech.edu.cn
许多: xuduo2022@shanghaitech.edu.cn

Notes:

Please follow the following steps to submit HW2. The submission will be open till Apr 9th, 2023. 23:59:00.

- Submit the report as the pdf file (SI100B/SI100F-ID-Name-hw2.pdf) to Blackboard
- Package all the files (Python/Matlab code) into <SI100B/SI100F-ID-Name-hw2>zip, also submit the zip file to Blackboard

Make sure the name of your code looks like 'q1.py' so we can easily match your answer to the question. You can also use q1a.py, q1b.py to separate different question. Please write the report **in English** carefully and complete the coding tasks with Python or Matlab. Failing to submit the homework as the required format (including not writing in English) will deduct 20 points from your score. Failing to submit the homework before the due date will receive 60% of the score you should get. No submission will be no score for this homework. The maximum latency for your homework is one week after the DDL, which is Apr 16th, 2023. 23:59:00.

Problem 1: Image computation (10')

Angiography is a medical examination method that uses a contrast agent injected into the blood to make the vascular system visible under X-rays. Two lower-limb angiography images are provided in the attachment, where "start.png" and "end.png" are images before and after blood with contrast agents flowing through the lower-limb, respectively. You need to extract the image of blood vascular system and show the result image. (Hint: Implement the image subtraction and then complement image.)

Problem 2: Noise deduction (30')

We give two images "board_a.png" and "board_b.png". You need to:

(a) Describe the noise types of given images, briefly explain why.
(b) For each image, choose a filter to reduce its noise and claim the reason you choose it.
(c) Change the kernel size of two filters, explain what is the influence. (i.e. kernel size of 5*5, 7*7 and 9*9).

Problem 2 Checkpoints:

You need to answer the following questions in your report:

(a) The noise types of the two images. (2.5'*2) Why? (2.5'*2)
(b) The filters you choose to reduce noise on images. (1'*2) Why? (1'*2)
(c) The influence of kernel size to each filter. (6')

You need to show the following figures in your report:

(a) Two images after filtering. (5'*2)

(Hint: You can try cv2.GaussianBlur, cv2.blur(), cv2.medianBlur() and cv2.bilateralFilter())

Problem 3: Image Enhancement (30’)

Given an image “lena.png”, you need to: (30’)

(a) Implement Power-law (Gamma) Transformation to the image with parameter $c = 1,\ \gamma = 0.4$ and show the result image. (The formula of Gamma Transformation is $s(r) = c \cdot r^{\gamma}$) (10’)

(b) Implement Histogram Equalization to the image and show the result (You can use the function cv2.equalizeHist() in the package *opencv*.) (6’), then plot the normalized histogram of the original image and the result image. (8’)

(c) Compare the effects of different mentioned enhancement methods, and briefly describe the differences between these enhancement methods. (6’)

Problem 4: Thresholding and Edge detection (30’)

“Bone.png” shows a cross-section of human cortical bone imaged by synchrotron radiation microcomputed tomography. The black areas are canals that host bone vasculature. In this question, you need to:

(a) Apply binary thresholding with any value you choose and otsu thresholding algorithm to the image, show your threshold value, images and compare them.

(b) Apply canny edge detection algorithm to “Bone.png” to segment all black areas in figure. Show your result.

(c) Apply mean filter to the image to reduce the noise regions. Then, apply edge detection algorithm to the figure after filtering, show your result and explain why it is better.

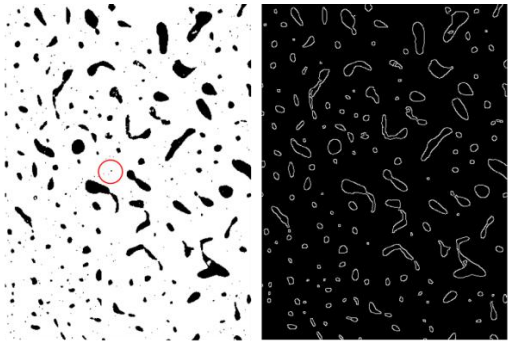

Fig 2. (a) Global thresholding result. Pixels in red circle should be reduced. (b) Edge detection result.

Problem 4 Checkpoints:

You need to answer the following questions in your report:

(a) Comparisons of binary thresholding and otsu algorithm, show their advantages and disadvantages. (3*2)

(b) Threshold value of binary thresholding and otsu algorithm. (3’)

(c) Explain why mean filtering algorithm can reduce noise regions and why it improves the edge detection quality. (5’)

You need to show the following figures in your report:

(a) Images of binary thresholding and otsu algorithm. (3’*2)

(b) Edge figure of direct application of canny detector. (3’)

(c) Image of “Bone.png” after mean filtering. (3.5’) Edge figure of the filtered image. (3.5’)

(Hint: You can try cv2.Canny(), cv2.threshold(), cv2.blur())

Bonus: Image Dehazing (10')

In bonus part, you will implement an algorithm that removes the fog in the images given by us. Your method is not limited. You can get most of the scores from the bonus part as long as your result is reasonable. The example of the defogging result is shown below:

(a) original image (b) processed image

Figure 2. Examples of image dehazing

You need to dehaze "city_fog.png" to achieve effect like Fig 2. (b). File "Dark_channel.py" contains full dehazing procedure. The steps of dark channel prior dehazing algorithm are:

1. Calculate the dark channel image I_{dark} of the fogged image I, apply guided filter to I_{dark} to achieve better image.

2. Choose top 0.1% points from I_{dark}, calculate the mean value of these points as the estimation of global atmospheric light A.

3. For each channel, restore pixel k with $J(k) = \frac{I(k) - I_{dark}}{1 - \min(I(k)/A)}$

The code of dark channel min filtering and guided filter have been provided in code, you just need to use them in the coding part and you will get the result.

Bonus Checkpoints:

You need to answer the following questions:

(a) What is your method? Briefly explain how it works. (5')

You need to submit the following images:

(a) The dehazed image of "city_fog.png". (5')

(Hint: You may try method in this paper:
Kaiming He, Jian Sun and Xiaoou Tang, "Single image haze removal using dark channel prior" 20 09 IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 1956-1963)