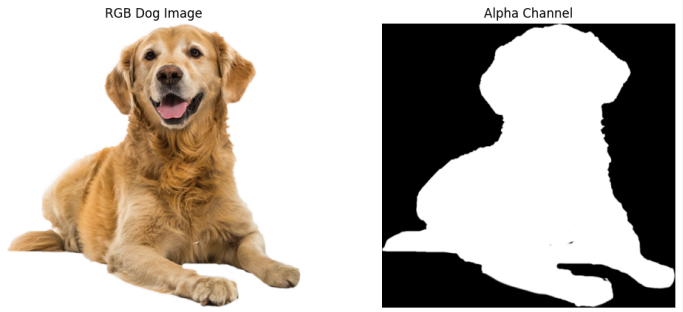
OPTI536_Homework_1_Answer_pdf

Date	@February 18, 2025
Tags	Homework 1
Person (Reporting Author)	Qi Wen Gan

This is the answer sheet for the homework 1 of OPTI365 Introduction to Imaging Science. To differentiate the required explanations from the problem, answers are provided in purple color background in all three questions below.

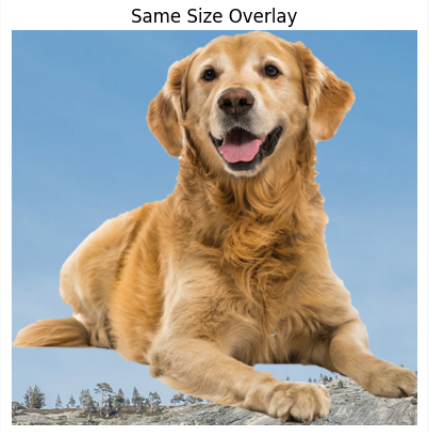
Question 1

a. RGB Dog Image and its Alpha Channel



b. Original Background

c. Overlay of dog image on top of cropped background (the upper left corner) image.



d. Overlay images with arbitrary sizes that is centered of the background image.

e. Bokeh Effect with standard deviation (SD) value of 10

f. Calculation for b and its result

```
# All measurements in mm
f = 50  # focal length in mm
D = 10  # aperture diameter in mm
d1 = 1500  # dog distance (1.5m = 1500mm)
d2 = 1000000  # background distance (1000m = 1000000mm)
sensor_width = 35  # sensor width in mm
crop_factor = 1.0 # Factor representing how much of the sensor width the image represents 1.0 means full sensor width, 0.5 means half width (cropped), etc.

# Calculate image distances using thin lens equation
i1 = (f * d1) / (d1 - f)  # dog image distance in mm
i2 = (f * d2) / (d2 - f)  # background image distance in mm
```

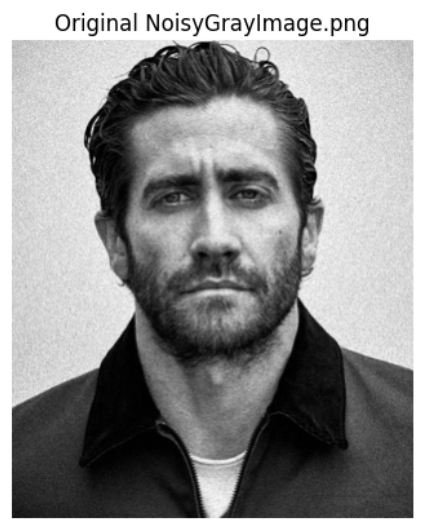
```
# Calculate blur circle in mm
b = D * abs(i2 - i1) / i2

# Consider actual sensor coverage
effective_sensor_width = sensor_width * crop_factor

# Convert blur circle to pixels
image_width = 6000
pixels_per_mm = image_width / effective_sensor_width
b_pixels = b * pixels_per_mm

# Set sigma according to b = 2σ
# σ  = b/2
sigma = b_pixels / 2

Blur circle (mm): 0.344
   Effective sensor width (mm): 35.000
   Pixels per mm: 171.4
   Blur circle (pixels): 59.0
   Sigma (pixels): 29.5
   Calculated sigma for bokeh effect: 29.51 pixels
```

g. Motion Blur


Regular Overlay
Motion Blur Overlay

Question 2

a. NoisyGrayImage.png Plot



b. Explain σb, σs in the bilateral filter and the value range

Sigma_s represent the spatial sigma (standard deviation for spatial gaussian). It controls the spatial extent of the filter, i.e., how far neighboring pixels influence the center pixel. A small σs means only nearby pixels contribute to the smoothing. Since large σs means pixels farther away will have a stronger effect and act like a gaussian blur, while the Sigma_s is independent of the intensity such as bit-depth of a image, for an 8-bit image, a reasonable range is σs ≈ 1 to 20 (measured in pixels).

Sigma_b represent brightness sigma (standard deviation for intensity gaussian). It controls how much the filter respects intensity differences (preserves edges). A small σb means only pixels with similar intensities contribute. A large σb allows more diverse intensity values to mix, which will produce a "overexposed blur image".

Assuming the 8 bit image intensity range value can be range within 2^8 = 256, 0-255, and when needs to be normalized by 255, to 0 to 1, the intensity value also ranged from 0 to 1. Therefore, the normalized value typically ranged from σb ≈ 0.05 to 0.2, to avoid the "overexposed effect".

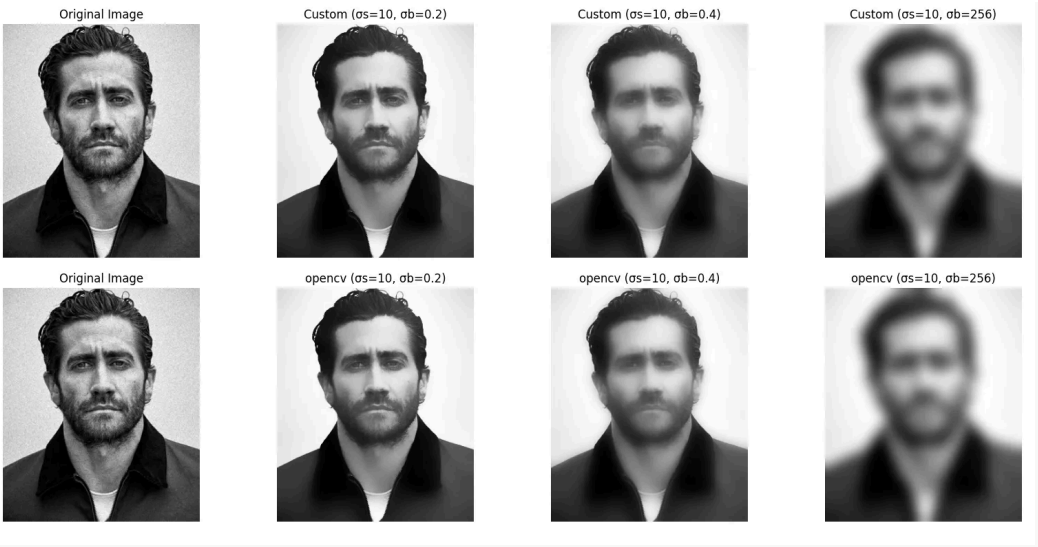
.

C. Plot NoisyGrayImage.png by using all four possible combinations of σs = 3,10 and σb = 0.2,0.4. Plot all four results and explain why the differences you see.

Overall, all four combinations successfully eliminated the salt and pepper noise at the background. With lower Sigma_b such as 0.2, the edge of the human contour and mustache are well preserved as compared to the higher Sigma_b of 0.4. On the other hand, with higher Sigma_s value such as 10, the image looks "flatter" compared to Sigma_s value of 3, which preserves less contrast information of the portrait. Given the same Sigma_b value of 0.2, the lower Sigma_s value, i.e. σb = 0.2 and σs = 10. This combination of parameters makes it the most suitable to eliminate the salt and pepper noise in the background while preserving the high-frequency details such as edge and contour.

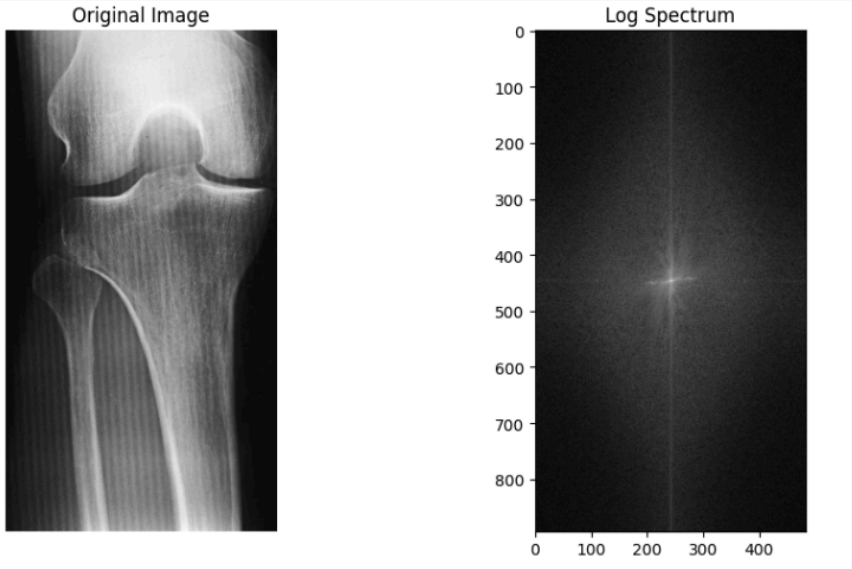
D. Comparison With bilateral filter from OpenCV

E. Fun and extra case with `sigma_s_values = [10]` and `sigma_b_values = [0.2,0.4,256]`

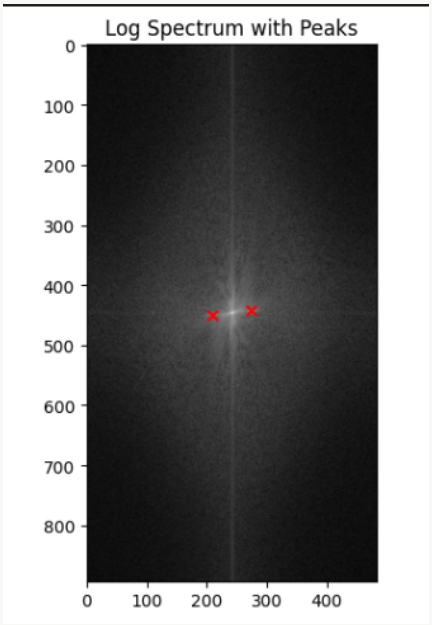

Original Image
Custom (σs=10, σb=0.2)
Custom (σs=10, σb=0.4)
Custom (σs=10, σb=256)
Original Image
opencv (σs=10, σb=0.2)
opencv (σs=10, σb=0.4)
opencv (σs=10, σb=256)

Question 3

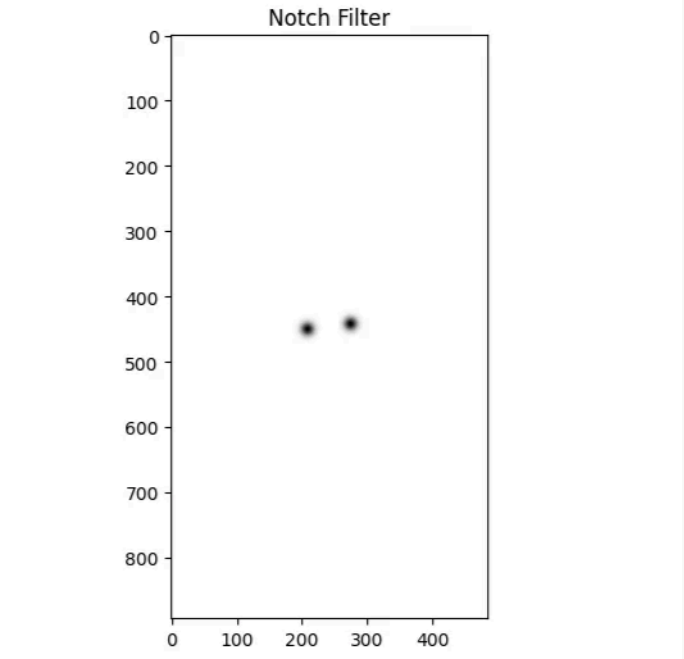
A. Logarithmic plot of Moire_example in Fourier space.



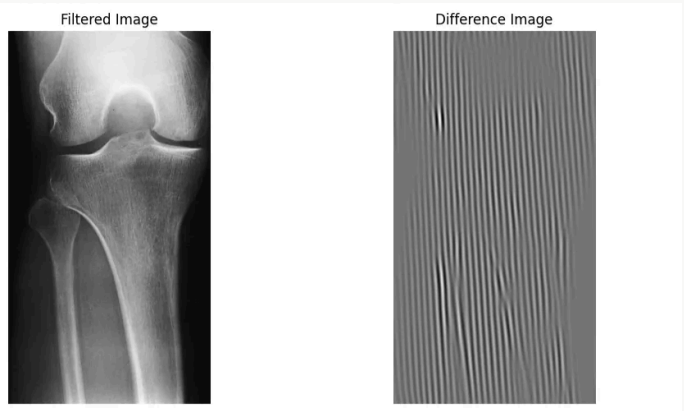
B. Plot the found position of the peaks (k=2) in the Fourier Transform image.



C. Notch filter with D0 = 7 (D0 with 25 is too large for the case of Moire_example.jpg)



D. Plot the difference image and Explain

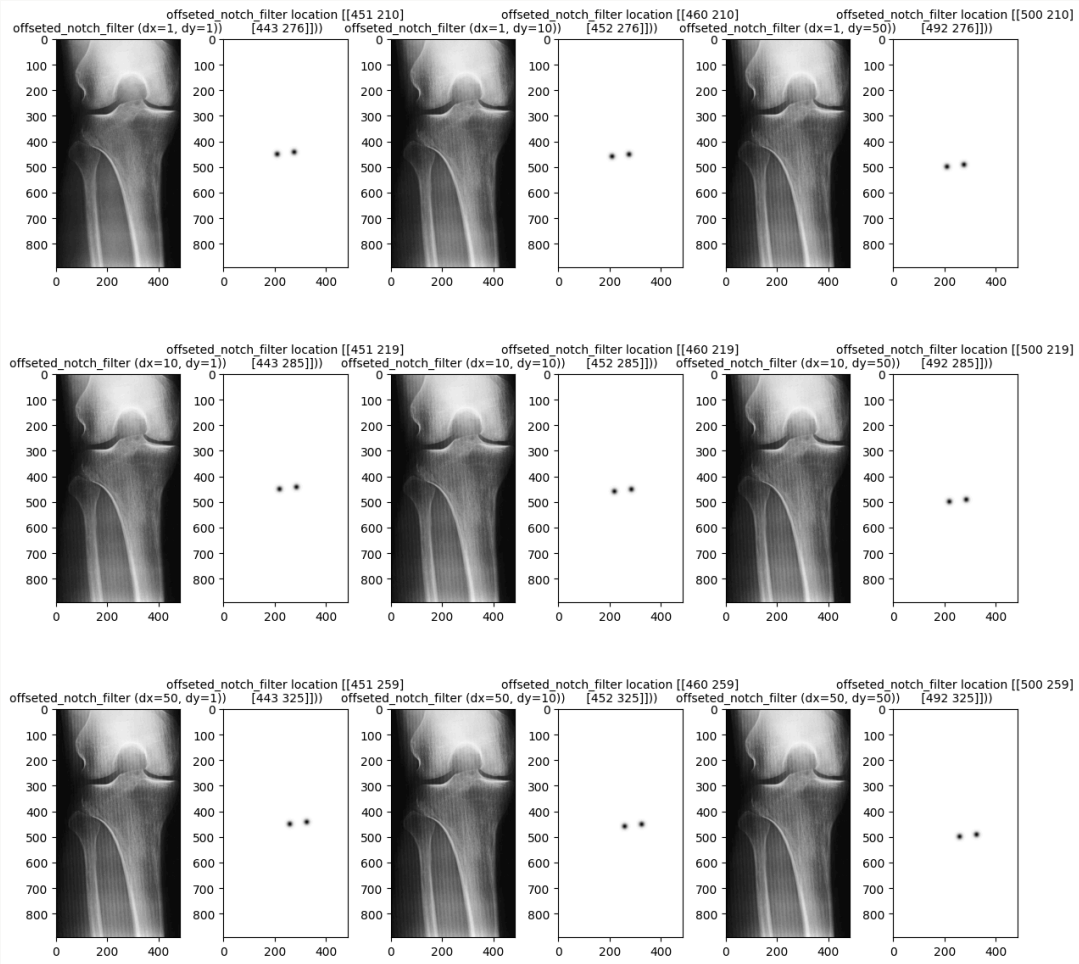


ANS: The Moire pattern now is absent from the image, and from the difference image, we can see the high frequency detail of the Moire was remove from the original picture.

E1. very large D0 of `d0_values = [1,5,10,50,100]`

ANS: When the notch filter size D0 is about 10, it gave us the most optimal Moire removal effect. While the D0=1 did remove some of the Moire pattern, it did not remove the sufficient amount of high frequency produced by the Moire. On the other hand, when the D0 is in large number, e.g. >50, it remove almost all the frequency details of the image, including the medium frequency details provided by the bones.

E2. Shifting of notch filter in fourier space with `y_shift = [1,10,50] and x_shift = [1,10,50]` and D0 with 7



ANS: When the notch filter is slightly offseted (e.g. dx=dy=1), it gave us a very similar optimal Moire removal effect. In the case of the dx or dy or both =10, the Moire pattern started to appear as part of the high frequency produced by the Moire did not masked from the original image. Thus, some other frequency bands were removed from the original image. On the other hand, when the offset is in large number, e.g. dx or dy or both =100, the Moire pattern became more obvious as almost all the high frequency band of the Moire pattern did not masked from the original image. Therefore, others medium frequency details provided by the bones have been eliminated, but the frequency details of moire pattern remained.