CSC 476/676 Computer Vision Mid-term Take-home Exam

Due March 28th (Sunday) end of the day

Total points: 100 pts

Your name: **Yunting Chiu**
AU ID: **5046705**

1. **The exam is open book; open notes. But you must submit your own independent work!**
2. **No discussion is allowed in the exam**. Copying from each other is strictly forbidden. **Copying lines of code from the Internet is strictly forbidden and will be considered cheating.**
3. But be careful of how you time yourself. You do not have to finish the questions in order. Pick up the easiest questions first to work on.
4. **Please write in clear font and you can use extra papers.**
5. **For the short answer questions, please directly answer them in this word document.**
6. **For programming questions, please submit a zipped folder of images and .ipynb as well as as link of Google CoLab.**
7. **Please write equations clearly with equation editor.**
8. **You can insert photos and drawings directly to the word document. But please make sure your writing/pictures are clear and easy to read.**

Short Answers

1. (5pts) Under what conditions will a line viewed with a pinhole camera have its vanishing point at infinity?

   Ans: It is required that the line be parallel to the image plane.

2. (15pts)

   (A) What is the Fourier transform of the 8×1 vector [1,-1,1,-1,1,-1,1,-1]?
   Please show how you drive the answer.

   This is an eight-dimensional signal. Given the vector [1,-1,1,-1,1,-1,1,-1], we know that N = 8, k = N-1 = 7

$N = 8$ and $w_8 = e^{-j2\pi/8} = e^{-j\pi/4}$. Therefore, applying the geometric sum we obtain

$$X[k] = \sum_{n=0}^{7} (w_8)^{nk} = \begin{cases} \frac{1-(w_8)^{8k}}{1-(w_8)^{k}} = 0 & \text{when } k \neq 0 \\ 8 & \text{when } k = 0 \end{cases}$$

Applying the geometric sum, we calculate the answer is [0, 0, 0, 0, 8, 0, 0, 0] with the following note.

$$X[k] = \sum_{n=0}^{7} x[n](W_8)^{nk}$$

$$X[0] = x(0)(W_8)^{0k} + x(1)(W_8)^{\overset{1\times 0}{1k}} + x(2)(W_8)^{\overset{2\times 0}{2k}} + x(8)(W_8)^{8k}$$

$$= 1-1+1-1+1-1+1-1 = 0$$

$$X[1] = x(0)(W_8)^{0k} + x(0) + (W_8)^{1k} + \cdots + x(8)(W_8)^{\overset{8\cdot 1}{8k}} = 0$$

Similarly we can get $X[2], X[3], X[4], X[6], X[7], X[8] = 0$

$$X[5] = x(0)(W_8)^{0.5} + x(1)(W_8)^{1.5} + \cdots + x(8)(W_8)^{8.5}$$

$$= 1 - W_8^{5} + W_8^{10} - W_8^{15} + \cdots$$

$$= 1+1+1+1+1+1+1+1 = 8$$

Ans: $[0, 0, 0, 0, 8, 0, 0, 0]$ #

B) If I take every second sample (show the result), what is the Fourier transform of that signal ?
[-1, -1, -1, -1]

By 4 point Discrete Fourier Transform (DFT) matrix, we can get the new signal by multiplying the transform matrix. So the answer is [-4, 0, 0, 0]

$$\underset{4\times 4}{\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix}} \times \underset{4\times 1}{\begin{bmatrix} -1 \\ -1 \\ -1 \\ -1 \end{bmatrix}} = \underset{4\times 1}{\begin{bmatrix} -4 \\ 0 \\ 0 \\ 0 \end{bmatrix}}$$ #

$$X(0) = -1-1-1-1 = -4$$

$$X(1) = -1 + j + 1 - j = 0$$

$$X(2) = -1 + 1 - 1 + 1 = 0$$

$$X(3) = -1 - j + 1 + j = 0$$

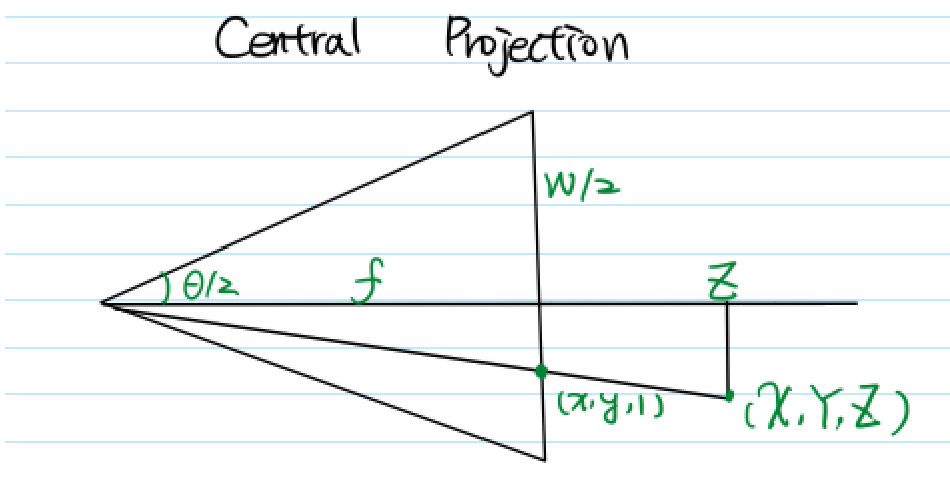
C) Is there any “aliasing” going on above ? Explain in words.

Yes, aliasing is basically a form of under sampling and what is displayed on the scopes is an indistinguishable waveform. Aliasing occurs when sampling a continuous geometric model, such as converting a continuous signal to a discrete signal.

3. (5pts) What is the pinhole camera geometry and the camera matrix P for a pin-hole camera? In other words, how do I relate 3D world points (X,Y,Z) to a 2D image point (x, y) through pin-hole camera projection? Please draw an illustration as well as derive the equations.

The pinhole camera geometry refers to the idea of projecting a 3D point through an ideal pinhole using a projection matrix known as the camera matrix where K is the intrinsic matrix, which is used to map 3D camera-centered points p_c to 2D pixel coordinates. (R, t) is the extrinsic matrix.

the following image showing the association between 3D and 2D coordinates. W = the width of image, f = the focal length f. theta = the horizontal field of view theta *H



4. (15pts) The binary image below is an image of a 3 pixel thick vertical line.

0	0	0	1	1	1	0	0	0
0	0	0	1	1	1	0	0	0
0	0	0	1	1	1	0	0	0
0	0	0	1	1	1	0	0	0
0	0	0	1	1	1	0	0	0
0	0	0	1	1	1	0	0	0
0	0	0	1	1	1	0	0	0
0	0	0	1	1	1	0	0	0
0	0	0	1	1	1	0	0	0

A) **Show the resulting image obtained after convolution of the original with the following approximation of the derivative filter [−1, 0, 1] in the horizontal direction.**

Convolving the binary image in the horizontal direction with [-1, 0, 1] yields [0, 0, 1, 1, 0, -1, -1, 0, 0].

B) How many local maxima of the filter response do you obtain?

Two local maxima: -1 and 1.

C) Suggest a filter which when convolved with the same image would yield a single maximum in the middle of the line. Demonstrate the result of the convolution on the original image.

The binary image can be convolved with the filter [1, 2, 1]^T to produce a single maximum in the middle of the line

5. (5pts) Using lens maker's formula to solve this.
When an object is held in front of a **convex lens** of focal length 5.0cm a real image forms at a distance of 15cm from it.

What is the object distance u? Ans: object distance is 7.5 cm

Lens formula: $\frac{1}{u}+\frac{1}{v}=\frac{1}{f}$

Concave lens ⇒ negative focal length

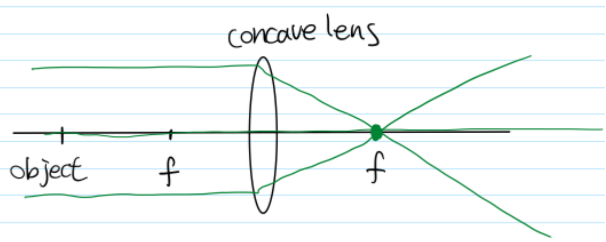
f is the focal length $=5$

Because it's a convex lens, so it has a positive focal length.

v is the img distance $=15$

$f=+5$

u is the object distance = ?



$\frac{1}{5}=\frac{1}{u}+\frac{1}{15}$

$\frac{1}{5}-\frac{1}{15}=\frac{1}{u} \Rightarrow \frac{3}{15}-\frac{1}{15}=\frac{1}{u}$

$\frac{2}{15}=\frac{1}{u} \Rightarrow 2u=15 \Rightarrow u=7.5$ #.

6. (5pts) What is the effect of the sigma parameter of a Gaussian, on the appearance of the image that has been convolved with a Gaussian?

   What about the derivative of the Gaussian filter and the effect of sigma on filtered images?

   Gaussian is a low pass filter, meaning that the feature is keeping low frequencies in image. Since the function of sigma in the Gaussian filter is to monitor the variance around its mean value, the image will be more blurred if the value of sigma is high.

$$G(x, y; \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

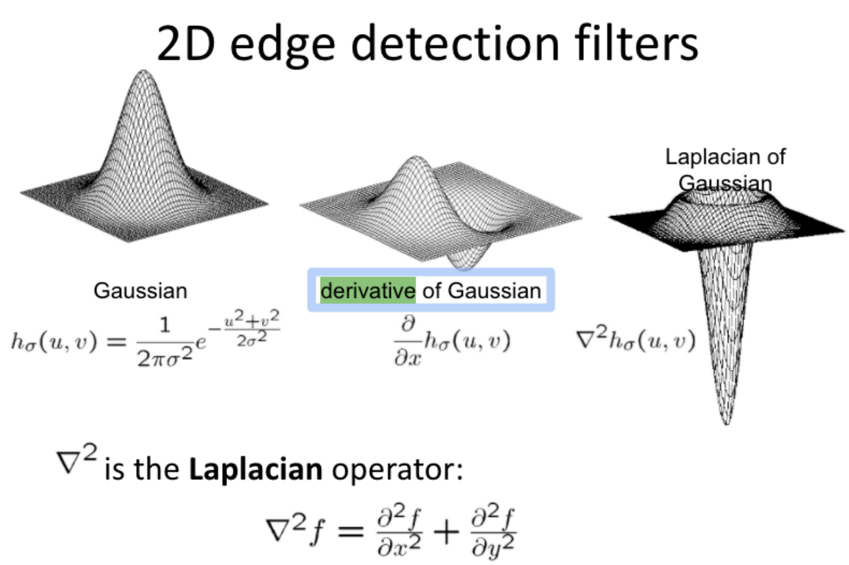
(Gaussian filter)

The role of sigma in the Gaussian filter is to control the variation around its mean value. So as the sigma becomes larger the more variance allowed around mean and as the sigma becomes smaller the less variance allowed around mean. In simple terms, the sigma leads the degrees of blur in images (smoothing).

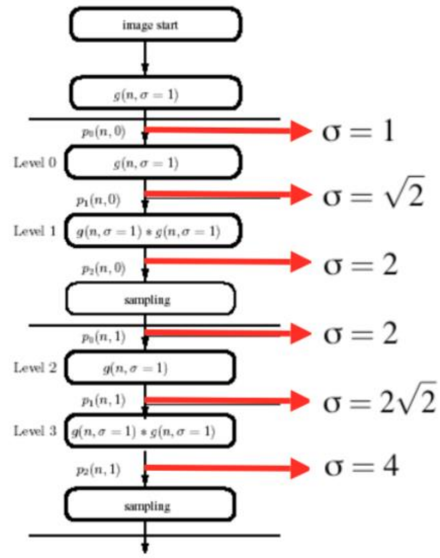
There are many ways to derive a Gaussian filter.

- Taking the first or second derivatives are known collectively as band-pass filters, since they filter out both low and high frequencies.
- The undirected second derivative of a two-dimensional image is known as the Lapacian operator.
- Taking a directional derivative of a Gaussian filter is known as the Sobel operator. -

Taking the Laplacian filter as an example. The Laplacian is often applied to an image that has first been smoothed with something approximating a Gaussian smoothing filter in order to reduce its sensitivity to noise, and hence the two variants will be described together here. The operator normally takes a single gray level image as input and produces another gray level image as output.



Effective Sigma at Each Level



Crowley etal.

7. (5pts) Say I am at a pixel (r, c), where r is the **row index** and c is the **column index**. How can you find the difference between pixels to the right of me and pixels to the left of me, i.e. between pixels (r, c-1) and (r, c+1), using a filter?

   What about the difference between pixels below/above me, i.e. (r+1, c) and (r-1, c)?

   To demonstrate, let us use a very simple 2D array. For instance, a sharpen 3*3 filter looks like:

   np.array([[0, -1, 0] ,[-1 ,5 ,-1], [0, -1, 0]])

   a. Use the sharpen filter like the below one to find the difference between columns.

      [0, -1, 0]

   b. Use the sharpen filter like the below to find the difference between columns.

      [0,

      5,

      0]

8. (5pts) Say you want to illustrate both the finer and coarser details in an image, separately. Describe a process that allows you to produce 3 "detail" images (i.e. images that only show the sharp detail of the image), the first one being very fine, the second a little coarser, and the third even more coarse.

   Feel free to explain this with both English and Equations.

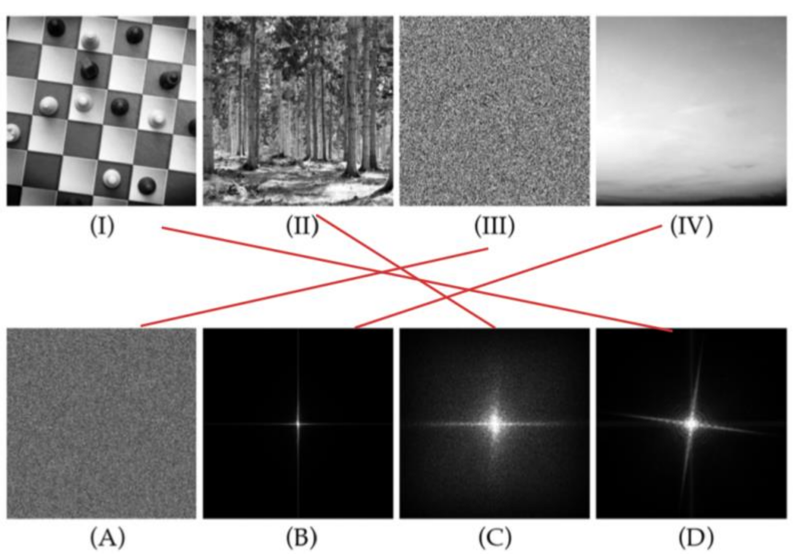
   For example, I need to explain what a low-pass filter is and what is a feature about this. First, I'll show the audience an original image. Assume this is an image of an apple, the audience can see the apple image in my first slide. An ideal low-pass filter must be transformed in the fourier domain, so I will demonstrate the second image to explain how the appearance of low-pass filter in the fourier domain. Third, we begin to use a low-pass filter to blur the original image, and the result is shown as the third image. In addition, I will re-compare image 1 to highlight what the main feature of the low-pass filter is.

   And the ideal low pass filter is:

$$H(u,v) = \begin{cases} 1 & D(u,v) \le D_0 \\ 0 & D(u,v) > D_0 \end{cases}$$

$$D(u,v) = \sqrt{u^2 + v^2}$$

9. (5pts) Match the corresponding Fourier transform images with real images.



10 (5pts) How can you find patterns in an image (e.g. let's say you're looking for a plus sign in images) in an image using a filter? You can show examples and write down specific methods.

Imagining you have two images, one is source and the other is destination. To blend these two images smoothly, we must first find a "mask" image. When you mask an area of an image, it protects the area from changes made to the rest of the image. For example, if you want to blend two images: moon and mountain, you should look for a mask that looks like a circle in order to remove the background from the original moon smoothly.

Would a 3x3 filter work for any image, i.e. where the plus signs appear at different sizes and orientations? Explain why it would/wouldn't.

Yes, 3*3 convolution filter is an optimal choice as it can be used for blurring, sharpening, embossing, edge detection, and more. Compared to 3*3 filter, 1x1 kernel size is only used for dimensionality reduction, which reduces the number of channels. Plus, 2*2 and 4*4 are even-sized filters. There will be distortions around the layers if this symmetry is not present. 5*5 is a larger one, and the training time is too long.

11 (5pts) Describe an algorithm that removes ONLY objects moving with a particular velocity in a video. Suppose the moving object has the velocity (vx,vy) and the sequence of the moving images can be described as:

In global translation, f0is the new position, where v is velocity, t is times.

$$f(x, y, t) = f_0(x - v_x t - v_y t)$$

As a result, the temporal derivative of f can be written as a function of the spatial derivatives of f0.

$$\frac{\partial f}{\partial t} = \frac{\partial f_0}{\partial t} = -v_x \frac{\partial f_0}{\partial x} - v_y \frac{\partial f_0}{\partial y}$$

And from here (using derivatives of f):

$$\frac{\partial f}{\partial t} + v_x \frac{\partial f}{\partial x} + v_y \frac{\partial f}{\partial y} = 0$$

It is preferred you describe it with both texts and math equations. When you write down equations, please explain in words what your equations mean and how it helps solving the problem.

When velocity equals zero in the Spatio-temporal Gaussian domain, the time axis becomes a vertical axis. If the velocity is negative one, the direction will be skewed from top left to bottom right. If the velocity is positive one, the direction will be skewed from top right to bottom left. The temporal Gaussian filter will only keep the same direction with it in the video frame during the space time session (x-axis is n, and y-axis is t) ; the other direction will be blurred. In this case, we can create a temporal Gaussian filter that removes objects that move at a certain velocity while keeping the rest throughout the process, such as isolating the boundary of the object.

Long answers (Programming Questions) (please submit a Google Colab link

https://colab.research.google.com/drive/1h3WSrVjY4ItwkPuXJFWsmSiOa9EgCMkk?usp=sharing

Question 1 (10pts) Basic image processing with Python. Write Python code to do the following.

1. Read in an image into Python as a matrix, and write down its dimensions. For instance, you can use Georgetown.jpg in the folder.
2. Convert the image into grayscale. You can use the image attached in the exam folder.
3. Find the darkest pixel in the image, and write its value and [row, column] in your answer sheet. Hint: Convert to a vector first, and use `numpy.flatten`.
4. Use the function `numpy.sum` and a logical operator measuring equality to a scalar, to determine and write down how many pixels in the grayscale image equal the value 6.
5. Consider a 31x31 square (a square with side equal to 31 pixels) that is centered on the darkest pixel. Replace all pixels in that square with white pixels (pixels with value 255). Do this with loops.
6. Now use the code you wrote above to find one of several pixels with value 6. Find which of those pixels are at least 15 pixels away from the border of the image in any direction (not including the 6-valued pixel itself). You can use loops. Let's call these 15-away 6-valued pixels `inds` (you don't have to call them this in your code).
7. Write code to *randomly* choose one of the `inds` pixels.
8. Now consider another 31x31 square, but this time gray (e.g. with pixel values 150). Take the image with the white square in it. Replace the randomly chosen pixel from above, and the 31x31 square in the image that's centered on this pixel, with the gray square. This time you are NOT allowed to use loops. Note that you shouldn't run into border issues because of the 15-away code you wrote above.
9. Make a new figure, display the modified image (which includes both a white square and gray square), and save it to a file using pyplot. `savefig(gcf, 'new_image.png')`.

Question 2 (15pts)

a) Implement an algorithm to sharpen a blurred image. You can get a blurred image by filtering an original image with a low-pass filter or shoot an image in low light or without focus. You are not allowed to use OpenCV or existing sharpening operators. Hint: You can do this via filtering in space domain or manipulating the image's Fourier Spectrum. You can find the blurred image in the attached folder. Display your results and compare the methods. Please return a sharpened color image. Compare your results with Photoshop.

b) Take a blurry or noisy image (shooting in low light is a good way to get both) and try to improve their appearance and legibility using Python code. Hint: To deblur, you can do a de-convolution such as inverting the convolution process. To remove noises, consider a median filter. Comment on the effects of your methods on the images.