CS 5390/7390 Midterm Fall 2020 Name: Eric Miao

This is a take home, open-book exam. Answers **must be handwritten** and submitted to Canvas as a single pdf. Your answers should fit in the space provided. Your answers should primarily be text, but you may include text, sketches, diagrams, graphs, mathematical formulas or code to best express your answers,

You may use any informational resources available to you including the textbook, research articles, OpenCV website, forum boards. You **may NOT use other students** in the class or post questions about these questions anywhere. If you wish clarity on questions, or directions, you may ONLY ask the instructor.

If you used websites or other sites other than the text and tutorial sites listed references in our course syllabus or linked through Canvas assignments and breakout room activities, please **include those links as entries in the textbo**x on Canvas in addition to uploading the pdf with your answers.

Your answers to this, in addition to work submitted for HW and Projects, will form the basis for questions during the conference-style portion of the exam that comprises 40% of your grade.

I. Theoretical Foundations Concepts

1. (a) What is the convolution theorem? (b) How can you use that to decide on an approach to implement efficient smoothing efficiently?

Convolution is based on the Fourier theory. The convolution of f and g is written f * g. It is defined as the integral of the product of the two functions after one is reversed and shifted.

Formula for Convolution: $(f*g)(t)=\int f(\tau)g(t-\tau)d\tau$

Smoothing could be achieved by using convolution in the spatial domain or doing point-wise multiplication in the frequency domain. Doing point-wise multiplication process in the frequency domain is more efficient than doing coevolution in the spatial domain.

2. (a) What is an image derivative and how might that be useful. (b) Provide a specific example of how to implement it, noting any particular characteristics of the output that we need to understand in order to interpret properly.

(a) A derivative of an image is a discrete derivative, and it's more of an approximation of the derivative. There are two different derivatives we talked about in class. The first derivative captures the "ramp (or slope)" and we can take a look at the rate of pixel value changes. One simple example is that you can take the derivative in the x-direction by taking the difference between the pixel values to the left and right of your pixel. Besides the first derivative, when applying the second derivative we can look at the zero crossing on the images and that's where the edges are located if we are using it for edge detection.

(b) To be more specific, we use image derivative in most of the edge detections algorithms, because change could be measured by the first and second derivative: A pixel is an edge if the magnitude of the first derivative of the image at that pixel is large. The zero crossings in the second derivative also tells us where the edges are. A great example is Canny edge detection algorithm: it uses a filter based on the derivative of a Gaussian in order to compute the intensity of the gradients.

Reference: (https://mccormickml.com/2013/02/26/image-derivative/)

3. (a) What type of information does the second derivative of an image provide? (b) How would you compute the second derivative in the spatial domain? (c) How would you compute second derivative of an image in the frequency domain?

In compare to 1st derivative, 2nd derivative is a more sophisticated method towards automatized edge detection because its derivatives enhance fine detail much better. After applying 2nd derivative on an image, the zero crossings in the image imply where the edges are. To compute the second derivative in the spatial domain, we can use the formula $f(x+1,y)-2f(x,y)+f(x-1,y)$. To compute the second derivative in the frequency domain, we can multiply the real and the imaginary components of the image with filter point for point.

4. (a) Explain why color images are customarily represented in a BGR color model system. Include a description that relates to characteristics of the human visual system.

BGR (Blue, Green, and Red) are considered the primary colors of all colors, because all colors could be combined through different proportions of these three colors. For human visual system, there are receptors at the back of the that are sensitive to three main wavelengths which are converted to BGR.

I also did some research and found that the reason OpenCV uses BGR as its main color model is more of a historical reason. This is interesting. (Reference: https://www.learnopencv.com/why-does-opencv-use-bgr-color-format/)

5. (a) What is the HSV color model? What correspondence does that have to the human visual system? And when would that be useful?

The HSV means Hue, Saturation, Value. The HSV is very similar to how human visual system functions. In real world, a normal full color image contains all types of data: brightness, hue, and saturation. In human visual system, the retina is made up of the layers of neurons where the visual processing of images is begun: Rods are highly sensitive to light and can respond to light from a single photon. However, they do not provide any color information. Cones give us the ability to distinguish color (photopic vision), saturation, and to a lesser extent intensity of light. Combining the function of Retina, Rods, and Cones, we see the world in colors. In image processing, since HSV abstracts color (hue) by separating it from saturation and pseudo-illumination, it is better for object detection than using BGR color model.

6. (a) What is bilinear interpolation? (b) Describe the scenario in which it is relevant. (c) Compare that to at least one other type of interpolation.

Bilinear interpolation entails interpolating in one direction followed by interpolating in the second direction. (Meaning that uses a weighted average of the four nearest cell centers.) It is a great tool to use when we want to scale or resample an image. In compare to nearest neighbor, the output of the nearest neighbor is determined by the nearest cell. However, the output values of the bilinear interpolation could be different than the nearest cells, but still stays within the range of the input values. And I guess that's why bilinear interpolation produces a smoother image than does the nearest neighbor approach.

7. (a) What is quantization of an image? (b) Where is that relevant in a discussion comparing analog and digital images? (c) How is that different from sampling?

(a) The definition of quantization in image processing is the process of converting a continuous range of values into a finite range of discreet values.

(b) Analog images are generally continuous signals. From analog to digital, it requires both sampling and quantization. Digital images are stored with quantified parameters. It only requires quantization when needed.

(c) Quantization is doing the opposite job in compare to image sampling. It is done on the y axis (Imagine we have coordinate values on the X-axis, and we have amplitudes on the Y-axis). When we are quantizing an image, we are actually dividing a signal into partitions.

Reference: https://vivadifferences.com/difference-between-sampling-and-quantization-in-digital-image-processing/

8. Why is it important to smooth an image before applying an edge detector?

Simple answer: To reduce noise. By smoothing an image, it removes the high frequency content from the signal, whereas edge detection usually looks into high frequency to detect edges.

9. Geometric transformations are often classified according to the preservations of different characteristics and the type of transformation. (a) What types of transformations on the 2D plane are possible in the type known as "rigid transformations" (b) What is the difference between an affine transformation and a projective transformation in these terms? (i.e. how are geometric elements transformed/preserved on a plane?)

(a): Normally, Translations, and rotations. Reflections could be a sub-category of ridge transformation too.

(b): There are two main differences between an affine transformation and a projective transformation:

- The projective transformation does not preserve parallelism, length, and angle. But it still preserves collinearity and incidence.
- Since the affine transformation is a special case of the projective transformation, it has the same properties. However, unlike projective transformation, it preserves parallelism.

Reference: https://www.graphicsmill.com/docs/gm5/Transformations.html

II. Mathematical and Computational Tools

10. (a) What is the gradient of an image and why is it useful? (b) Provide specific description of how to compute it, including specific kernels required and a mathematical or code expression of computation.

An image gradient is a directional change in the intensity or color in an image. It provides two pieces of information – magnitude and direction: The direction of the gradient tells us the direction of greatest increase while the magnitude represents the rate of increase in that direction.

When working with edge detection, we often need to compute an image gradient. We can compute gradient at a pixel in x-direction and y-direction separately and then combine them to get the gradient of the image at that point.

Magnitude : $$||\nabla f|| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

Direction : $$\theta = \tan^{-1}\left(\frac{\partial f}{\partial y} / \frac{\partial f}{\partial x}\right)$$

A simple example:
(Reference: https://en.wikipedia.org/wiki/Image_gradient)

11. (a) What steps comprise the Canny Edge Detector method? (b) What is the purpose of non-maximal suppression in the Canny Edge Detection?

The Process of Canny edge detection algorithm can be broken down to 5 different steps:

(a) Apply Gaussian filter to smooth the image in order to remove the noise
(b) Find the intensity gradients of the image
(c) Apply non-maximum suppression to get rid of spurious response to edge detection
(d) Apply double threshold to determine potential edges
(e) Track edge by hysteresis: Finalize the detection of edges by suppressing all the other edges that are weak and not connected to strong edges.

Non-maximum suppression is applied to find the locations with the sharpest change of intensity value.
(Reference: https://en.wikipedia.org/wiki/Canny_edge_detector)

12. (a) Write a mathematical expression that defines a binary image, *g*, by thresholding an image, *f*, against a constant value *t*. (b) Is this a linear or non-linear operation?

$$\boldsymbol{g}(\boldsymbol{x},\boldsymbol{y}) = \binom{0}{1}$$

This is a linear operation.

13. What is the difference between a Laplacian Pyramid and a Gaussian Pyramid? Why might you use one over the other?

The elements of a Gaussian Pyramids are smoothed copies of the image at different scales. There are two steps to generate a Gaussian pyramid: Smoothing: Remove high-frequency components that could cause aliasing. Down-sampling: Reduce the image size by half at each level.
For Laplacian Pyramid, it starts with smoothing and we get a lowpass version of the image. Then, this image is unsampled by inserting zeros in between each row and column and interpolating the missing values by convolving it with the same filter w to create the expanded lowpass image. In short terms, Laplacian Pyramid computes the difference between unsampled Gaussian pyramid level and Gaussian pyramid level.

14. Suppose that you were interested in detecting diagonal lines in an image, rather than vertical and horizontal. (That is you want to apply a feature detector that will respond more to edges at 45 and 135 degrees than edges at 0 and 90 degrees.) Define the values of a 7x7 spatial domain kernel to achieve this:

-1	-1	-1	-1	-1	-1	**2**
-1	-1	-1	-1	-1	**2**	-1
-1	-1	-1	-1	**2**	-1	-1
-1	-1	-1	**2**	-1	-1	-1
-1	-1	**2**	-1	-1	-1	-1
-1	**2**	-1	-1	-1	-1	-1
2	-1	-1	-1	-1	-1	-1

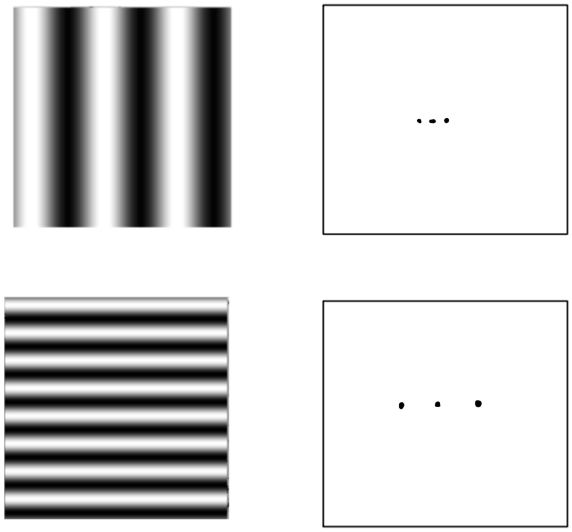
III. Visual Understanding and Interpretation

1. In each row below, the left column shows a 128x128 spatial domain image with intensity determined from function(s) of spatial location.

(a) Sketch the magnitude spectrum in the frequency domain associated with each image in column B.

(b) In column C, write the equation, or line of code, that defines the intensity as a function of spatial location at a single point (x, y). (Determine the function too!)

A (intensity image) B (magnitude spectrum) C (function)



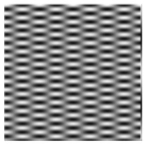
Code for Image A:

```
void showMat1(){
    Mat m1(128, 128, CV_32FC1, Scalar(0,0,0));
    for(int i = 0; i < m1.rows; i++){
        for(int j = 0; j < m1.cols; j++){
            m1.at<float>(i,j) = 0.5*sin(j/255. * 4 * 2 *M_PI)+0.5;
        }
    }
    imshow("output", m1);
    waitKey(0);
}
```

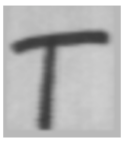
Code for Image B:

```
void showMat2(){
    Mat m1(128, 128, CV_32FC1, Scalar(0,0,0));
    for(int i = 0; i < m1.rows; i++){
        for(int j = 0; j < m1.cols; j++){
            m1.at<float>(i,j) = sin(i/255. * 16 * 2 *M_PI)+0.5;
        }
    }
    imshow("output", m1);
    waitKey(0);
}
```

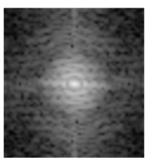
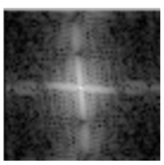
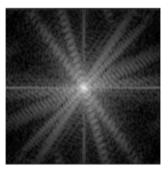
Which image in the first set goes with the magnitude spectrum in the second set?

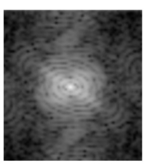
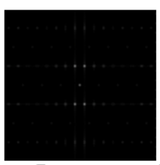
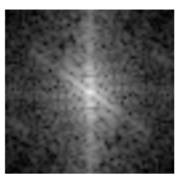
1: F 2: E 3: A

4: B 5: C 6: D

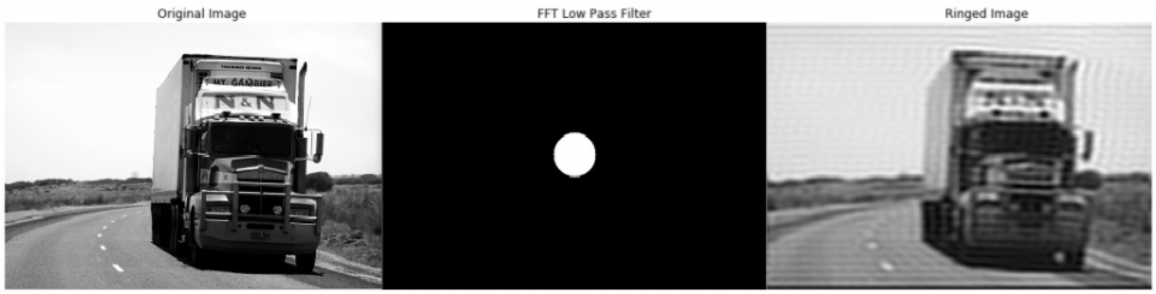
  

A B C

D E F

Below you will see an image on the left, a frequency domain filter shown as an intensity image in the middle and the effect of applying that filter on the right. (a) Explain what this filter is intended to do, visually. (b) Can you explain how a frequency domain filter, designed in this way, relates to the human visual system and its sensitivity to spatial frequency? (c) Please describe the visual artifacts that appear in the resulting image and (d) the likely cause and a reasonable approach to a solution.



Low Pass Filter

(a) This low pass filleter is blurring the original image.

(b) In the spatial domain, the further away from the center, the higher the frequency is. By using a filter that has a bright center and dark surrounding, we can eliminate the high frequency components in the image. This process acts like a low pass filter and give us a low frequency image like the result image on the right.

(c) There is some ‘dots’ effects on the ringed image.

(d) The filter in the frequency domain is not the same when converting into the spatial domain. To make it easier to understand, I have added a screenshot below. In the frequency domain, the filter would have a continuous curve (Like the image on the right). However, in the spatial domain, the filter would look like the function on the left side. The upper bound causes the artifacts on the result image.