EECE-5626 (IP&PR) : Midterm-1 Exam : 2021-FALL

Friday October 22 for 130 Minutes between 3:30 pm -- 10 pm (EDT)

[50 Points]

Instructions:

1. You are required to read the NU Academic Integrity policy (given below) and sign below that the submitted work is your own work. This is a COE requirement.
2. You are required to complete this exam using Live Editor.
3. All your plots must be properly labeled and should have appropriate titles to get full credit.
4. Use of the equation editor to typeset mathematical material such as variables, equations, etc., is highly recommended. However, in the interest of time, a properly scanned image of a neatly hand-written fragment can be inserted as your work at the required space.
5. After completing this assignment, export this Live script to PDF. In the exam you will be submitting the PDF file. **DO NOT SUBMIT LIVE EDITOR (.MLX) FILE**. **IT WILL NOT BE GRADED**.
6. You should try to complete this exam in two hours. Use additional ten minutes for submission activities. You will have only chance for submission.

Academic Integrity Policy

A commitment to the principles of academic integrity is essential to the mission of Northeastern University. The promotion of independent and original scholarship ensures that students derive the most from their educational experience and their pursuit of knowledge. Academic dishonesty violates the most fundamental values of an intellectual community and undermines the achievements of the entire University.

As members of the academic community, students must become familiar with their rights and responsibilities. In each course, they are responsible for knowing the requirements and restrictions regarding research and writing, examinations of whatever kind, collaborative work, the use of study aids, the appropriateness of assistance, and other issues. Students are responsible for learning the conventions of documentation and acknowledgment of sources in their fields. Northeastern University expects students to complete all examinations, tests, papers, creative projects, and assignments of any kind according to the highest ethical standards, as set forth either explicitly or implicitly in this Code or by the direction of instructors. The full academic integrity policy is available at

http://www.northeastern.edu/osccr/academic-integrity-policy/

Declaration

By signing (i.e., entering your name in the given format below) and submitting this exam through the submission portal, I declare that I have read the Academic Integrity Policy and that the submitted work is my own work.

Enter your name (Tyler B McKean):

IF YOU DO NOT SIGN ABOVE, 10% POINTS WILL BE CUT FROM THE OVERALL SCORE.

Default Plot Parameters: Please execute the following code before any other code.

```
set(0,'defaultfigurepaperunits','points','defaultfigureunits','points');
set(0,'defaultaxesfontsize',10);
set(0,'defaultaxestitlefontsize',1.4,'defaultaxeslabelfontsize',1.2);
```

Problem-1 [15-Points] Descriptive Mini-Questions

Answer the following questions concisely. These questions are not related.

(a) [5-Points] Human Vision System

If you shield your eyes using a sheet of white paper when looking directly at the sun, the side of the sheet facing you appears black. Which of the visual process(es) discussed in class is responsible for this effect? Explain this process (or processes) in your own words.

Answer:

This visual process is discusses the effect of Simultaneous Contrast, where the perceived brightness of sunlight eluminating through the paper does not depend only on its intensity. As the sunlight shinning through the piece of white paper intensifies, the edges of the white piece of people would appear to become darker.

(b) [5-Points] [5-Points] Linear Filtering

You are given a compiled MATLAB function, **`g = linFilter2D(f,h)`**, in the form of a pcode (or **`.p`** file) that performs linear space-invariant filtering, but you are not told whether it performs a convolution or correlation operation. Note that you cannot edit or print this **`.p`** file to see what it does because it is a binary file. Provide details of a test that you would perform to determine which operation this function performs.

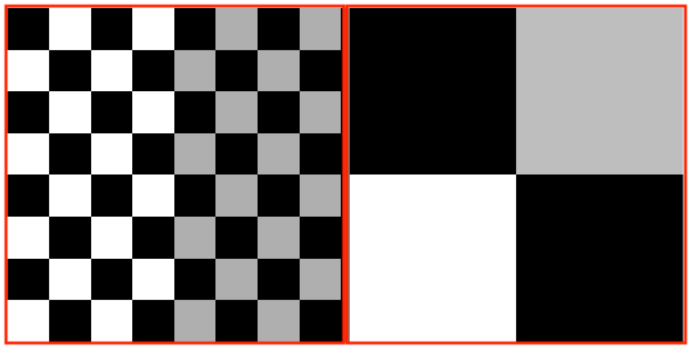
Answer:

Since, Correlation differs from the process of Convolution, in that Correlation in not commutative, we could change the order of the input arguments, f and h. If the resulting matrix, g, is the same matrix for either order of input arguments, then the function performs Convolution. If the answers differ when changing the input argument order, then the function performs Correlation.

(c) [5-Points] Image Histogram

Two images shown below are quite different, but their histograms are the same. Each image is blurred by using the same 3×3 averaging impulse response. Explain clearly if the histograms of the blurred images would still be equal?

Note: The red borders below show the extent of images, they are not part of images.



Answer:

Because the image on the right has larger boundary points between the black and white regions than the image on the left, when the images are blurred, the image on the right will contain a larger number of different intensity values resulting in the histograms of the two images differing from one another.

(d) [5-Points] Visual Effect in the Reconstruction of Sampled Images

What is the ***single most important*** factor that leads to the visual effect of ***Moirè pattern***? Justify your answer. How can this effect be eliminated while displaying an image on the monitor?

Answer:

The factor that leads to the visual effect known as the Moire pattern is when a periodic image or pattern clashes with another periodic pattern around the same frequency which cause a third pattern to arise. A resolution to this for the application of a image displaying on a monitor would be to slightly defocus the image so that the high frequencies are attentuated. However, this would need to be done beofre the image was sampled and displayed to the monitor.

Problem-2 [10-Points] Two-Dimensional Linear Convolution

Let the rectangular support (or extent) of image $f(m,n)$ be $[M_{f1}, M_{f2}] \times [N_{f1}, N_{f2}]$ and that of the impulse response $h(m,n)$ be $[M_{h1}, M_{h2}] \times [N_{h1}, N_{h2}]$. These supports are shown below on the left of the '=' sign.

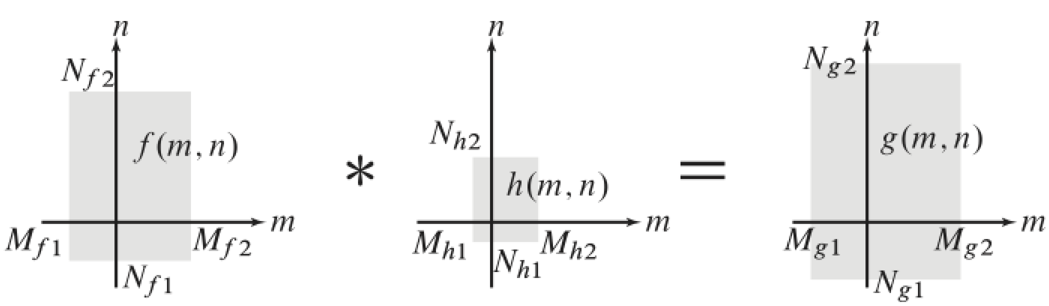


Figure 2.1

Let $g(m,n) = f(m,n) * h(m,n)$, which is a linear convolution between $f(m,n)$ and $h(m,n)$. Let the rectangular support of the resulting $g(m,n)$ be $[M_{g1}, M_{g2}] \times [N_{g1}, N_{g2}]$, which is shown above on the right of the '=' sign.

(a) [4-Points] Size Determination

Show that the support parameters of $g(m,n)$ are given by

$$\begin{aligned} M_{g1} &= M_{f1} + M_{h1}, \quad M_{g2} &= M_{f2} + M_{h2}, \\ N_{g1} &= N_{f1} + N_{h1}, \quad N_{g2} &= N_{f2} + N_{h2}. \end{aligned}$$

You must provide mathematics-based approach in your proof and not just a verbal description of a proof.

Proof:

Convolution of a of $(M_1 \times N_1)$ size sequence with a $(M_2 \times N_2)$ size sequence is a $(M_1 + M_2 - 1) \times (N_1 + N_2 - 1)$ size sequence.

so in the case of $(M_{g1} \times N_{g1})$ = $(M_{f1} + M_{h1} - 1) \times (N_{f1} + N_{h1} - 1)$

and $(M_{g2} \times N_{g2})$ = $(M_{f2} + M_{h2} - 1) \times (N_{f2} + N_{h2} - 1)$

(b) [4-points] 2-D Convolution

The finite extent images $f(m,n)$ and $h(m,n)$ are shown below. The values of these images are 1 on the red heavy-dots and 0 elsewhere on the light dots.

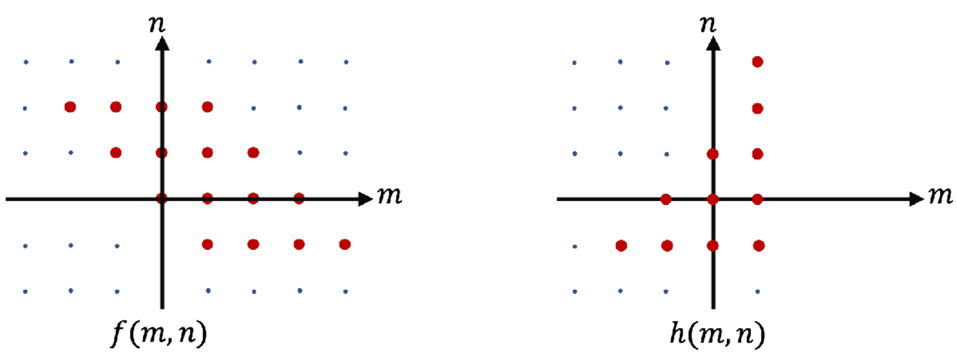


Figure 2.2

Using MATLAB determine the linear convolution image $g(m,n)$. Use variables given in Figure 2.2 above as your MATLAB variable names. Print the result of your convolution array **g** (i,e., do not put semicolon '**;**' after the statement that computes it so that the result shows up in the editor).

MATLAB script:

```
clc; clear; close all;
% Insert your code below
f = [1,1,1,1,0,0,0;0,1,1,1,1,0,0;0,0,1,1,1,1,0;0,0,0,1,1,1,1];
h = [0,0,0,1;0,0,0,1;0,0,1,1;0,1,1,1;1,1,1,1];
g = conv2(f,h)
```

```
g = 8×10
     0     0     0     1     1     1     1     0     0     0
     0     0     0     1     2     2     2     1     0     0
     0     0     1     2     3     4     3     2     1     0
     0     1     2     4     5     5     5     3     2     1
     1     2     4     6     7     7     6     4     2     1
     0     1     2     4     6     7     7     5     3     1
     0     0     1     2     4     6     6     5     3     1
     0     0     0     1     2     3     4     3     2     1
```

(c) [2-Points] Location of `g(0,0)`

From the results of part (a), indicate the ***value and location*** of **`g(0,0)`** sample by providing row and column indices of your **`g`** array.

Answer:

The resulting convolved matrix is of size 8 x 10 with the location of $g(0,0)$ being of the value 6 and located at row 6 column 4 of the array above.

Problem-3 [15-Points] Spatial Domain Image Enhancement

This problem deals with an adaptive enhancement technique called ***statistical differencing*** using the image file **`stairs.png`** which is available in the Midterm Exams module in Canvas. Download and store it in your working directory.

```
clc; close all; clear;
f = imread('stairs.png');
figure('Position',[0,0,4,4.3]*72); subplot('Position',[0,0,1,4/4.3]);
imshow(f); title('Stairs Image');
```

Stairs Image

Let $f(m,n)$ denote the **stairs.png** image. Let $\mu_f(m,n)$ be the *space-varying* average of $f(m,n)$ that can be computed by blurring $f(m,n)$ using a small mask, that is,

$$\mu_f(m,n) = \frac{1}{K} \sum_{(k,\ell) \text{ over } K \text{ pixels}} f(m-k, n-\ell) \qquad (3.1)$$

Similarly, $\sigma_f(m,n)$ be the *space-varying* standard deviation (SD) of $f(m,n)$ that can be computed by blurring $\{f(m,n)-\mu_f(m,n)\}^2$ using a small mask and then taking the square-root, that is,

$$\sigma_f(m,n) = \sqrt{\frac{1}{K} \sum_{(k,\ell) \text{ over } K \text{ pixels}} \{f(m-k,n-\ell) - \mu_f(m-k,n-\ell)\}^2} \qquad (3.2)$$

(a) [2-Points] Average Image

Using MATLAB compute and display the space-varying average image $\mu_f(m,n)$. Use the 9×9 averaging (or uniform) mask for blurring.

Solution: Use the given variables in equations (3.1) as MATLAB variables. For μ_f use **mu_f**.

```
% Use the following fragment for image display
figure('Position',[0,0,4,4.3]*72); subplot('Position',[0,0,1,4/4.3]);
imshow(mu_f); title('Space-varying Average Image');
```

```
% Insert your code below and use the above fragment for image display
```

```
h = fspecial('average',9);
mu_f = imfilter(f,h);
figure('Position',[0,0,4,4.3]*72); subplot('Position',[0,0,1,4/4.3]);
imshow(mu_f); title('Space-varying Average Image');
```

(b) [2-Points] Standard Deviation Image

Using MATLAB compute and display the space-varying SD-image $\sigma_f(m,n)$. Use the 9×9 averaging mask for blurring.

Solution: Use the given variables in equations (3.1) and (3.2) as MATLAB variables. For μ_f use **mu_f** and for σ_f use **sigma_f**.

```
% Use the following fragment for image display
figure('Position',[0,0,4,4.3]*72); subplot('Position',[0,0,1,4/4.3]);
imshow(mu_f); title('Space-varying SD Image');
```

```
% Insert your code below and use the above fragment for image display
M = 362; N = 362; sigma_f = zeros(M,N);
for i = 1:M
    for j = 1:N
        sigma_f(i,j) = sqrt((1/(M*N))*sum(f(i,j) - mu_f(i,j)).^2);
    end
end
sigma_f = sqrt(sigma_f);
figure('Position',[0,0,4,4.3]*72); subplot('Position',[0,0,1,4/4.3]);
```

```
imshow(sigma_f); title('Space-varying SD Image');
```

Let σ_{new} be the *constant* standard deviation. Then the following technique adaptively enhances small details in the contrast:

$$g(m,n) = \mu_f(m,n) + [f(m,n) - \mu_f(m,n)]\,\frac{\sigma_{\text{new}}}{\sigma_f(m,n)}. \qquad (3.3)$$

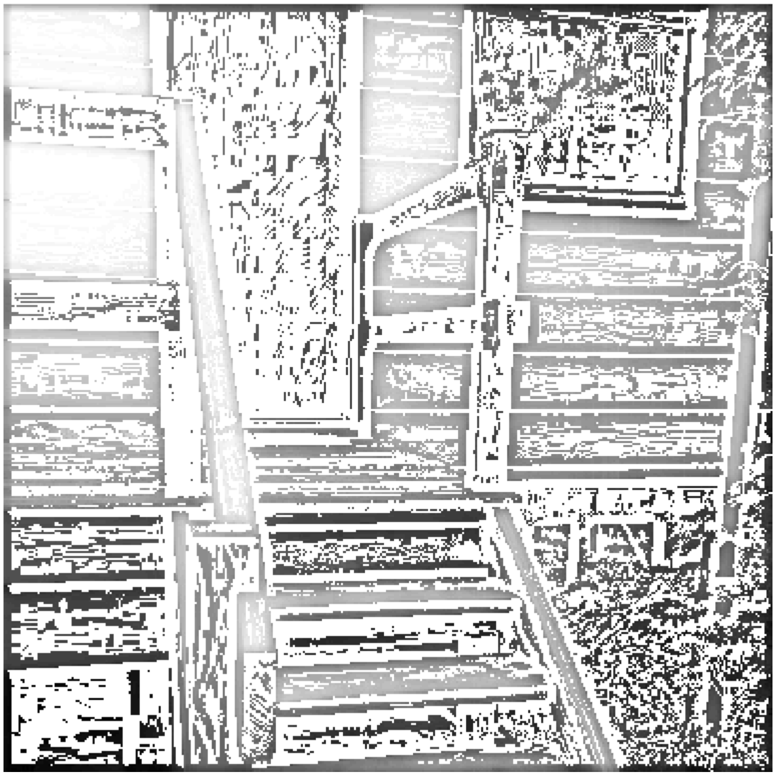
(c) [3-Points] Adaptive Enhancement-1

Using MATLAB compute $g(m,n)$ via (3.3). Use σ_{new} as the average of the minimum and maximum of $\sigma_f(m,n)$. Scale $g(m,n)$ properly to display it.

Use MATLAB variable **sigma_new** for σ_{new} and **g1** for $g(m,n)$ in equation (3.3).

```
% Use the following fragment for image display
figure('Position',[0,0,4,4.3]*72); subplot('Position',[0,0,1,4/4.3]);
imshow(mu_f); title('Adaptive Enhancement-1');
```

```
% Insert your code below and use the above fragment for image display
sigma_new = (min(sigma_f)+max(sigma_f))/2;
g1 = intensityScaling(double(mu_f) + double(f - mu_f).*(sigma_new./sigma_f));
figure('Position',[0,0,4,4.3]*72); subplot('Position',[0,0,1,4/4.3]);
imshow(g1); title('Adaptive Enhancement-1');
```



(d) [2-Points] Discussion-1

Discuss the visual quality of the image $g(m,n)$. Did you observe any unusual pixels? Explain.

Answer:

The image contains a high contrast of pixel intensities with a majority of the values being in the upper range. The photo look almost like a photo negative of the original image, f.

You may have encountered display problems when $\sigma_f(m,n) \approx 0$. To avoid it, let β be a parameter that controls the local standard deviation. Then one possible approach is

$$g(m,n) = \mu_f(m,n) + [f(m,n) - \mu_f(m,n)]\frac{\beta\sigma_{\text{new}}}{\sigma_{\text{new}} + \beta\sigma_f(m,n)}. \tag{3.4}$$

(e) [4-Points] Adaptive Enhancement-2

Using MATLAB compute $g(m,n)$ via (3.4). Convert $g(m,n)$ into an intensity image prior to its display. Use σ_{new} as in (c) above. Experiment with values of β over `[0:1:5]` range and select the visually best (in your opinion) image for display. Display only this image and not others.

Use MATLAB variable **`beta`** for β and **`g2`** for $g(m,n)$ in equation (3.4).

```
% Use the following fragment for display.
figure('Position',[0,0,8.25,4.3]*72);
subplot('Position',[0,0,4/8.25,4/4.3]);
imshow(f); title('Stairs Image');
subplot('Position',[4.25/8.25,0,4/8.25,4/4.3]);
imshow(g2); title('Adaptive Enhancement-2');
```

```
% Insert your code below and use the above fragment for image display
beta = 5;
g2 = intensityScaling(double(mu_f) + double(f - mu_f).*((beta.*sigma_new)./(sigma_new + beta.*(
figure('Position',[0,0,8.25,4.3]*72);
subplot('Position',[0,0,4/8.25,4/4.3]);
imshow(f); title('Stairs Image');
subplot('Position',[4.25/8.25,0,4/8.25,4/4.3]);
imshow(g2); title('Adaptive Enhancement-2');
```

(f) [2-Points] Discussion-2

Discuss the visual quality of the image **g2**.

Answer:

The result of the Adaptive Enhancement blurred the image compared to the original image but not as intense as the μ_f image. The image has a foggy look to it and some of the gray level intensities have been darkened such as the shadows on the porch door.

Problem-4 [15-Points] Two-Dimensional DSFT

Let the 2-D signal be $f(m,n) = \left(\frac{1}{2}\right)^{|m|}\left(\frac{1}{3}\right)^{|n|}$, $-\infty < m,n < \infty$ and let $F(u,v)$ denote its 2-D discrete space Fourier transform (DSFT).

(a) [5-Points] Computation of 2-D DSFT

Determine $F(u,v)$ where u and v are digital frequencies in cycles/sample. Since the given 2-D signal is real and symmetric, its DSFT must also be real and symmetric.

Note that the frequency symbols are lowercase "U" and "V" letters, respectively. Notations used in the lecture notes, μ ('mu') and ν ('nu'), are not used here so that it will be easier for you while writing or typing.

The most compact expression for the DSFT will get full credit.

Solution:

$$F(u,v) = \sum_{m=-\infty}^{\infty}\sum_{n=-\infty}^{\infty}\left(\frac{1}{2}\right)^{|m|}\left(\frac{1}{3}\right)^{|n|}e^{-j2\pi(um+vn)}$$

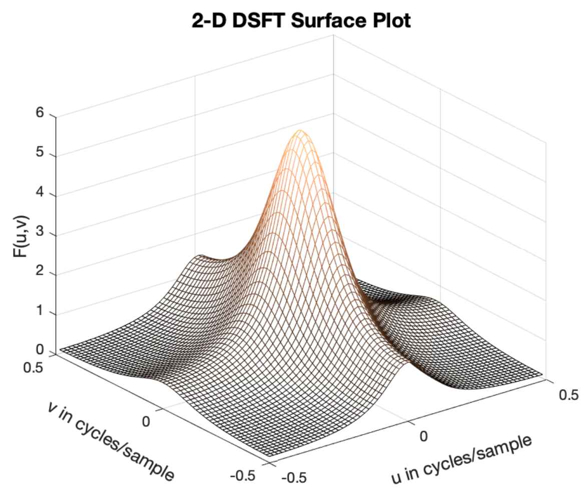
(b) [5-Points] Plotting of 2-D DSFT

Using MATLAB, provide a mesh plot of the real-valued $F(u,v)$ in the region $-\frac{1}{2} \le u,v, \le \frac{1}{2}$ over 64×64 grid.

There are two approaches to the plotting of this function.

1. Use the derived expression from part (a), compute it over the 64×64 grid, and plot the resulting 2-D array.
2. Take a 64×64 size 2-D DFT of suitably truncated $f(m,n)$, compute its real part, and plot the results.

Choose any one approch to solve this part. The plot should look like the one given below.



Use the **`mesh`** function for your mesh plot and use the following fragment for mesh plot display.

```
figure('Position',[0,0,6,5]*72);
```

```
% insert your plotting code here

set(gca,'view',[-37.5,30]); shading interp; colormap("copper");
xlabel('u in cycles/sample',"Rotation",16,...
    "Position",[-0.15,-0.6,-0.6]);
ylabel('v in cycles/sample',"Rotation",-30,...
    "Position",[-0.63,-0.25,-0.53]);
zlabel('F(u,v)'); title('Magnitude Surface Plot');
```

MATLAB script:

```
clc; close all; clear;
% Insert your code below and use the above fragment for image display
```