Assignment 3

Eternal Elegance: Exploring Fourier Transform Applications in Image Processing

Homeworks Guidelines and Policies

- ***What you must hand in.*** It is expected that the students submit an assignment report (HW3_[student_id].pdf) as well as required source codes (.m or .py) into an archive file (HW3_[student_id].zip). Please combine all your reports just into a single .pdf file.
- ***Pay attention to problem types.*** Some problems are required to be solved *by hand* (shown by the icon), and some need to be implemented (shown by the icon). Please do not use implementation tools when it is asked to solve the problem by hand, otherwise you will be penalized and lose some points.
- ***Don't bother typing!*** You are free to solve by-hand problems on a paper and include their pictures in your report. Here, cleanness and readability are of high importance. Images should also have appropriate quality.
- ***Reports are critical.*** Your work will be evaluated mostly by the quality of your report. Do not forget to explain your answers clearly, and provide enough discussions when needed.
- ***Appearance matters!*** In each homework, 5 points (out of a possible 100) belong to compactness, expressiveness, and neatness of your report and codes.
- ***MATLAB is also allowable.*** By default, we assume you implement your codes in Python. If you are using MATLAB, you have to use the equivalent functions when it is asked to use specific Python functions.
- ***Be neat and tidy!*** Your codes must be separated for each question, and for each part. For example, you have to create a separate .m file for part b. of question 3, which must be named 'p3b.m'.
- ***Use bonus points to improve your score.*** Problems with bonus points are marked by the icon. These problems usually include uncovered related topics, or those that are only mentioned briefly in the class.
- ***Moodle access is essential.*** Make sure you have access to Moodle, because that is where all assignments as well as course announcements are posted. Homework submissions are also made through Moodle.

- ***Assignment Deadline.*** Please submit your work **before the end of June 3rd**.
- ***Delay policy.*** During the semester, students are given only 7 free late days which they can use them in their own ways. Afterwards, there will be a 20% penalty for every late day, and no more than four late days will be accepted.
- ***Collaboration policy.*** We encourage students to work together, share their findings, and utilize all the resources available. However you are not allowed to share codes/answers or use works from the past semesters. Violators will receive a zero for that particular problem.
- ***Any questions?*** If there is any question, please do not hesitate to contact us through the following email addresses: **ebp.mohsen@gmail.com** and **ali.the.special@gmail.com**.

1. Practicing the Basics of 1-D Fourier Transform (15 Pts.)

Keywords: *Fourier Transform, Inverse Fourier Transform, Duality/Linearity/Time Shift/Convolution Property of Fourier Transform, Dirac Delta Function*

The **Fourier Transform** stands as an extraordinary milestone in the annals of scientific discovery. It offers us a profound understanding that virtually all phenomena in our world can be depicted as waveforms, encompassing functions of time, space, or other variables. By employing the Fourier transform, we gain an exceptional and formidable perspective on these waveforms, disassembling them into an alternative representation comprised of sine and cosine components.

The objective of this problem is to provide you with an opportunity to reinforce your understanding of the theoretical foundation of the Fourier transform, before encountering some image-related applications within this domain.

To begin, let's employ the properties of the Fourier transform and the relevant definitions to compute and determine the Fourier transform of the following functions:

a. $g(t)=e^{-2(t-1)}u(t-1)$

b. $g(t)=e^{-|t+3|/2}$

c. $g(t)=\mathrm{rect}(3t+2)$

d. $g(t)=\left(\dfrac{\sin(\pi t/4)}{\pi t}\right)\cos\left(\dfrac{5\pi}{2}t\right)$

e. $g(t)=\sin t e^{-|t|}$

f. $g(t)=\delta(t-2)-\delta(t+1)$

Next, let's determine which, if any, of the real signals depicted in the figure satisfy the following conditions for their Fourier transforms:

g. $\mathfrak{Re}\left\{X(j\omega)\right\}=0$

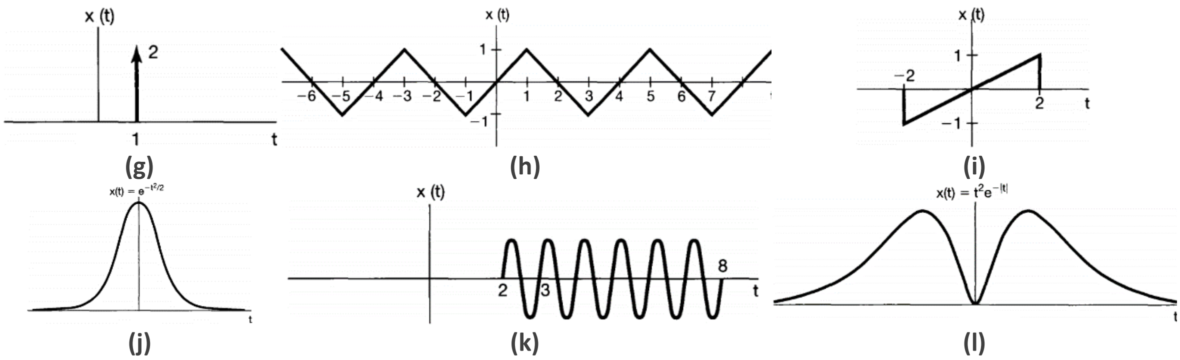
h. $\mathfrak{Im}\left\{X(j\omega)\right\}=0$

i. $\int_{-\infty}^{\infty} X(j\omega)d\omega=0$

j. $\int_{-\infty}^{\infty} \omega X(j\omega)d\omega=0$

k. $X(j\omega)$ is periodic.

l. There is a real α such that $e^{j\alpha\omega}X(j\omega)$ is real.



Finally, let f = [2, 1, -1, 3, 0, -2] and h = [-2, 2, -1].

m. Pad both f and h with zeros to a length of 10 and find the convolution $f * g$.
n. Find the convolution by calculating a product of discrete Fourier transforms.
o. Plot $F_p(u)$, $G_p(u)$ and $H_p(u)$ as points in the complex plane for $0 \le u \le 9$.
p. Calculate and interpret $\sum g_p^2(n)$ and $\sum \left|G_p(u)\right|^2$.

2. The Illusory Nature of Frequency Domain Image Analysis (15 Pts.)

Keywords: *Frequency Domain, Fourier Analysis, Magnitude, Phase*

This problem primarily revolves around the concept of representing images in the **Frequency Domain**. It explores the implications of the **Fourier Representation** of an image and examines the relationship between image representation in the spatial and frequency domains.

To begin, let's examine the images depicted in Figure 1.

a. Identify the image(s) in which the Fourier transforms, $F(u, v)$, have the following properties:

i. The real part of $F(u, v)$ is zero at all u, v.

ii. The imaginary part of $F(u, v)$ is zero at all u, v.

iii. $F(u, v)$ is purely real and positive for all u, v.

iv. $F(0, 0) = 0$

v. $F(u, v)$ has circular symmetry

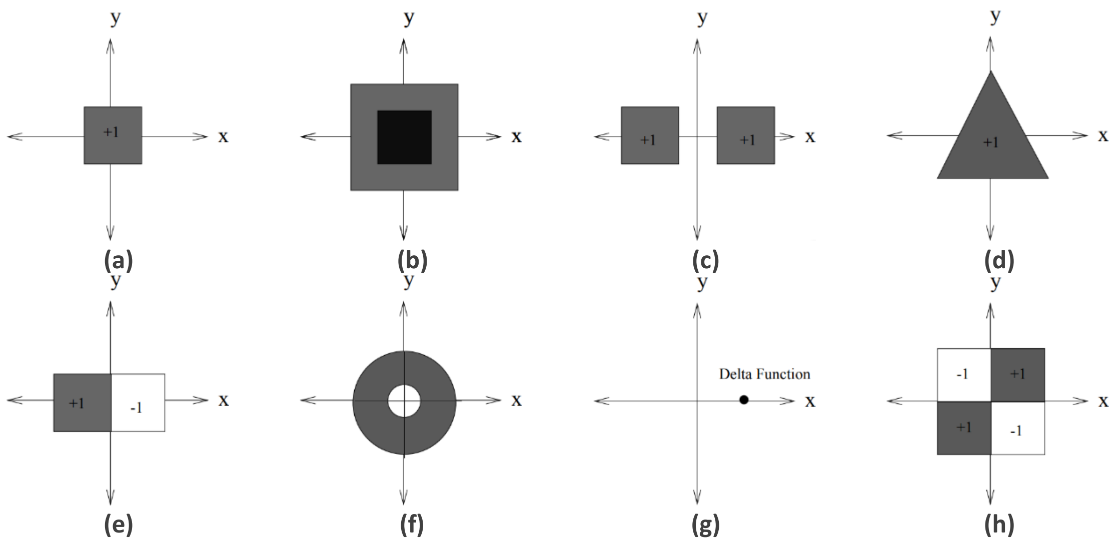


Figure 1 For the initial section, we are presented with images exhibiting distinct structures

Next, we will delve into a practical exercise to analyze the representation of images in the frequency domain. Let's explore the optical illusions depicted in Figure 2.

b. Calculate and visualize the magnitude and phase of the images. Explain the specific structures within each image that contribute to the observed results. Additionally, specify the relationships between the images in the spatial domain and the frequency domain.

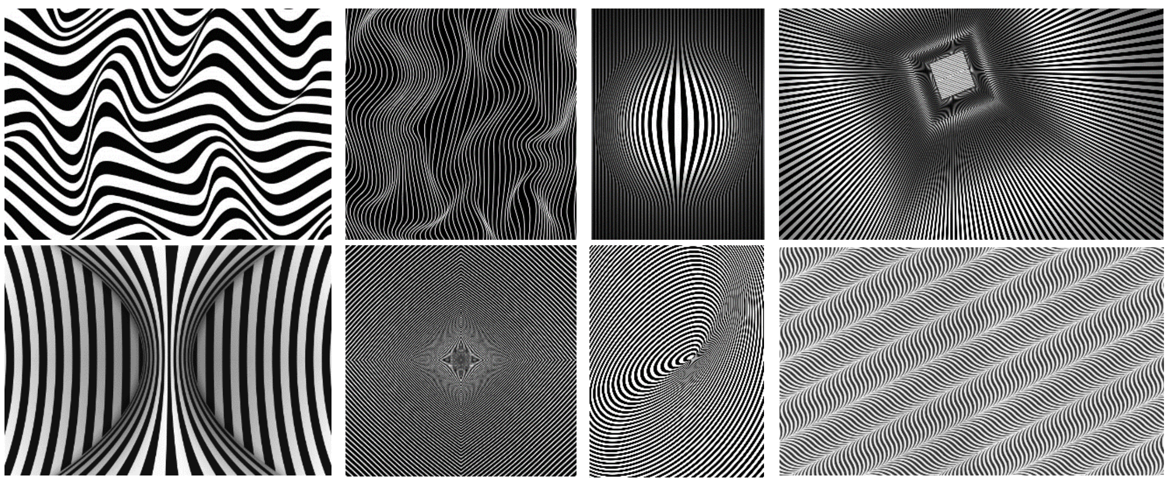


Figure 2 Several optical illusions featuring various patterns and structures

3. Decoding Identities: Exploring Face Recognition in the Frequency Domain (25 Pts.)

Keywords: *Frequency Domain, Fourier Transform, Inverse Fourier Transform, Image Filtering, Face Recognition, Template Matching, Minimum Average Correlation Energy (MACE) filter, Cross Correlation*

Owners of iPhones newer than iPhone X often perceive **Face Recognition** systems as highly sophisticated, complex, and cutting-edge technology that demands extensive expertise and understanding to be implemented. Although this holds true for robust face recognition systems like Apple Face ID, we aim to illustrate that possessing only fundamental knowledge of images in the **Frequency Domain** is sufficient to develop a capable face recognition system.

More precisely, the idea is to use **Template Matching** to design a specific filter, called *minimum average correlation energy* (MACE) filter, and perform a **Cross Correlation** between the test image and this filter. The intuition behind designing a MACE filter is to create a filter using a set of training images that would produce correlation output. Through the MACE filter, the average correlation energy over the given training faces will be minimized, hence it produces sharp peaks at the origin of the correlation plane, while producing lower values in the rest of the plane, Figure 4.

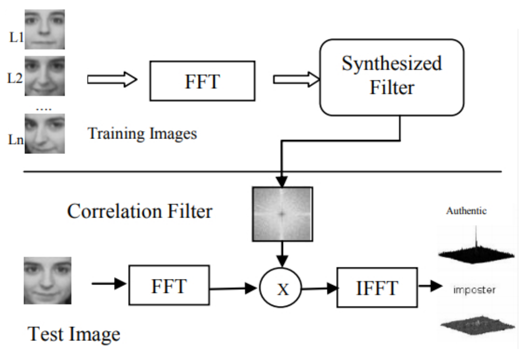


Figure 3 Block diagram of the algorithm. Training images are used to obtain a synthesized filter in the frequency domain. In the testing phase, the filtering output will give high peak value at the origin of the correlation plane if the test face belongs to true class.

Assuming N images of size $n \times m$ with d $(= n \times m)$ pixels in the training set, the MACE filter H is given by:

$$H = D^{-1}X\left(X^{*}D^{-1}X\right)^{-1}u$$

where D is a $d \times d$ diagonal matrix which contains the average correlation energies of the training images in its diagonal elements, X is a $N \times d$ matrix and X^* denotes the complex conjugate of X. The i^{th} column of X represents the Discrete Fourier coefficients of the i^{th} training image. The column vector u is also an $N \times 1$ vector which contains the correlation peak values for a series of training images. It has N elements, each corresponding to desired values at the origin of the correlation plane of the training images. These constraint values are often set to one for all training images from the authentic class.

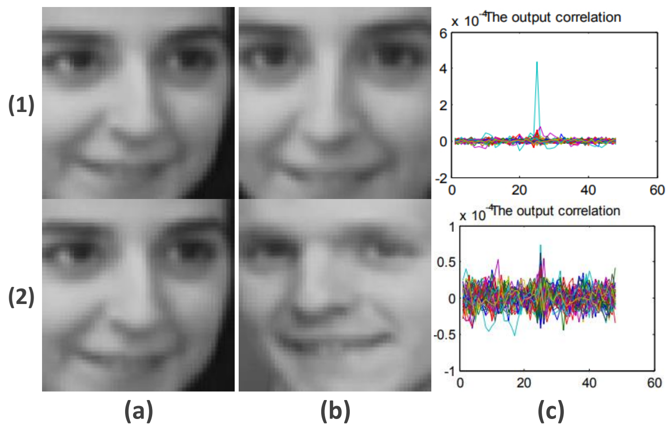


Figure 4 Two test cases, one a 'matched' and the other a 'non-matched' state. As can be seen, correlation coefficients shows a peak when the classes are the same (a) Test image (b) training images (c) The correlation coefficients

The MACE filter optimizes a criterion to produce a desired correlation output plane. Here, we use maximum peak value for measuring correlation plane. The correlation plane with sharp peak value indicates that the test image is matched, while the imposter output denotes that the test image is not matched, Figure 4. Please refer to this paper for more details about how the algorithm works.

Download the cropped version of the [AR face database](). This subset of the database contains 2600 images of 100 individuals, each with 26 images with different facial expressions, illumination conditions and occlusions, Figure 5. The images must be converted to grayscale before being used in the following parts.

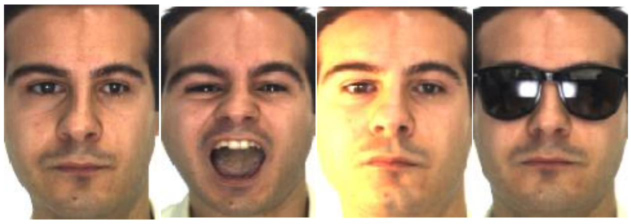


Figure 5 Samples in AR database are varied in their facial expressions, illumination conditions and occlusions

a. Ignore the images with occlusions, and keep the images with postfix 01 to 07 as the training images, and those with postfix 14 to 20 as the test images. Pick a random test image, and perform face recognition using the above algorithm. Display the MACE filter, output correlation and the recognition result.
b. Calculate the recognition rate by dividing the number of true identifications to total number of test images.
c. Now consider the whole dataset, i.e. images with postfix 01 to 13 as the training images, and those with postfix 14 to 26 as the test images. Pick a random test image, and perform face recognition using the above algorithm. Display the MACE filter, output correlation and the recognition result.
d. Calculate the recognition rate, and compare the result with part (b).

4. Forget Photoshop: Embracing the Power of Gaussian Filter! (30 Pts.)

Keywords: *Image Filtering, Frequency Domain, Fourier Transform, Bandpass Filters, Lowpass Filter, Highpass Filter, Gaussian Filter, Laplacian Filter, Image Blending*

Mixing two images and creating a blended image has always been a popular operation in the area of images. This operation – which is usually done using image editing softwares like Photoshop – generally leads to produce hilarious images which amuse everyone. In this problem, we wish to get familiar with a Fourier transform-based method which blend images even better than Photoshop!

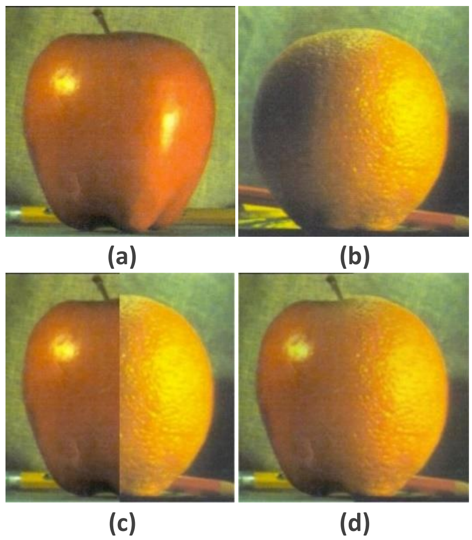


Figure 6 Images blended by the proposed method look very realistic and natural (a) Left pair (b) Right pair (c) Result of the copy-and-paste method (d) Result of the proposed method

Here's the idea. Imagine we want to mix specific regions of an image – defined by a mask – with another image, so that the result looks as natural as possible. One way to do so, is to simply crop those regions using the provided mask and place them in the destination image. But as expected, it leads to a stark result (Figure 6). To obtain a seamless and gradual result in the boundaries, one can compute a gentle seam between the two images separately at each band of image frequencies, and combine them eventually to get the final result.

More specifically, first both images are convolved with a highpass filter at different levels, each with increasing amount of sharpness. At the same time, mask of desired regions is also convolved with a lowpass filter at different levels, each with increasing amount of smoothness. Finally, at each level, highpass filtered images are combined using lowpass filtered mask as weights. The final result is obtained by adding the blended results at each level.

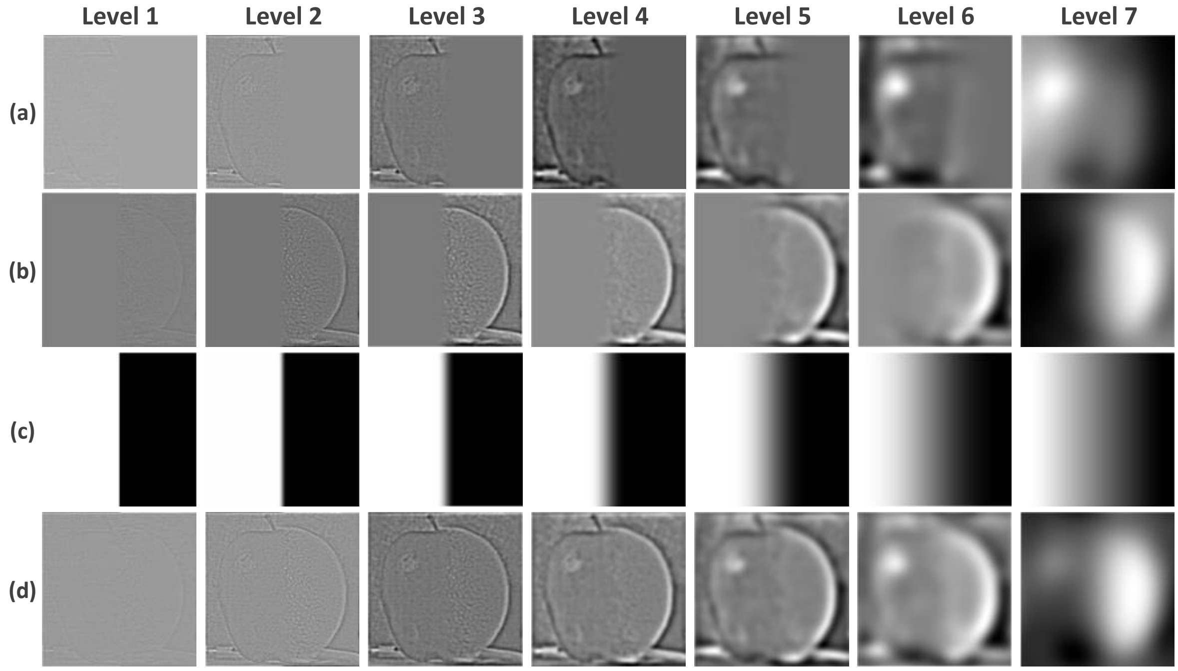


Figure 7 The procedure of the proposed method for mixing two images in different levels (a) The result of applying Laplacian filter on the left part (b) The result of applying Laplacian filter on the right part (c) The result of applying Gaussian filter on the mask (d) The result of combining two filtered images

Now let's see this in action. You are provided with three separate tasks, each with different considerations. The first one is asked to be blended in grayscale, while the other two must be done in color space. Also, the mask of desired regions is given in the first two tasks, while you have to design a mask of your own in the third.

You must apply the following procedure on each one of the given input pairs.

i. Build a set of Laplacian filtered versions of the images A and B using the equation $L_A^l = G_A^l - G_A^{l+1}$ (or $L_B^l = G_B^l - G_B^{l+1}$), where L_A^l (or L_B^l) and G_A^l (or G_B^l) are the Laplacian and Gaussian filtered versions of the image A (or B) at the level l, respectively. Set the number of levels $n = 6$, and the sigma value of the Gaussian filter at each level $\sigma = 2^{l-1}$.

ii. Build a set of Gaussian filtered versions of the mask image M. Consider $n = 6$, and the sigma value of the Gaussian filter at each level $\sigma = 2^{l-1}$.

iii. Compute the combined images C^l of each level l using

$$C^l = G_M^l L_A^l + (1 - G_M^l) L_B^l$$

iv. Obtain the final result C by summing C^l's through the levels $1, 2, \ldots, n$.

Note 1: Display all the intermediate results and also include them in your report.

Note 2: Your results must look as natural as possible. Try to find the most convenient parameters (mainly the number of levels, n) to accomplish this task.

(a) (b) (c) (d) (e)

Figure 8 Two examples of mixing images with the proposed method, one without alignment (top) and the other with alignment (bottom) (a) First image (base) (b) Second image (c) Mask of desired region (d) The result of simple copy-and-paste method (e) The result obtained by the proposed method

I. Joint product of Audi and Saipa!

The first task is a simple one; it only contains mixing images in grayscale.

e. Convert the images to grayscale and mix them using the above mentioned method. Your results must be in grayscale. Set different values for the number of levels *n*, and report the best result you obtained.

(a) (b) (c)

Figure 9 Creating a new model using a mask and a Gaussian filter! (a) 2020 Audi Q7 (b) SAIPA 151 (c) Mask which keeps half of each cars

II. Messi who? We have Donald in our team!

Now a more complicated case. The goal here is to merge images in color space. The above-mentioned method must first be applied to each channels separately, and then the blended results in each channel must be combined together to construct a final image.

f. Apply the method to different channels of the first image and obtain a final colored result. Set different values for the number of levels *n*, and report the best result you obtained.

(a) **(b)** **(c)**

Figure 10 Bringing Donald Trump into the football field! (a) Image of Lionel Messi (b) Aligned and edited image of Donald Trump (c) Mask separating Trump's head

III. Just a reminder: Wash your hands!

In the third and final task, the goal is to mix images in color space, while the mask is not given.

(a) **(b)**

Figure 11 Avoid bacterial infections with regularly washing your hands! (a) T-shirt image (b) Note image

g. Define a zero matrix of the size of the t-shirt image, and then place the note image in the centre of it. Use this matrix to create a mask (set the nonzero values to 1) and a modified image of the note (set the zero values to the background color) to use in the above-mentioned algorithm.

h. Apply the method to different channels of the first image, and then obtain a final colored result. Set different values for the number of levels *n*, and report the best result you obtained.

5. Some Explanatory Questions (10 Pts.)

Please answer the following questions as clear as possible:

a. Provide a mathematical explanation of how image shearing influences the phase and magnitude plots of an image in the Fourier domain.

b. While it is technically possible to perform convolution in three-dimensional space, such as in video processing where spatial and temporal dimensions are considered simultaneously, this approach is generally considered suboptimal and is rarely employed in practical video processing scenarios. Explain the reason.

c. Image Padding is essential for image filtering in the frequency domain. Let's consider two types of padding methods that involve adding an equal total number of zeros to the image: appending zeros to the ends of rows and columns, and surrounding the image with a border of zeros. Do you believe these two padding techniques would result in any differences when applied to filtering? Justify your answer.

d. Why does the shifted 2D DFT typically exhibit a plus sign shape, with higher values along the horizontal and vertical axes? Provide an explanation.

e. Even when the Nyquist criterion is satisfied, aliasing generally cannot be entirely eliminated in practical scenarios. Provide an explanation for this phenomenon.

Good Luck!
Mohsen Ebadpour, Ali Abbasi